Designing Deep Neural Networks for Continual Learning in an Open World

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Informatik und Mathematik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Martin Mundt
aus Görlitz

Frankfurt 2021
(D30)

vom Fachbereich Informatik und Mathematik der

Johann Wolfgang Goethe - Universität als Dissertation angenommen.

Dekan:
Prof. Dr.-Ing. Lars Hedrich

Gutachter:
Prof. Dr. Visvanathan Ramesh (Erstgutachter)
Prof. Dr. Gemma Roig (Zweitgutachter)
Prof. Dr. Stefan Kramer (Drittgutachter)

Datum der Disputation: 23.08.2021

Publications

This cumulative dissertation is based on the following manuscripts:

(Mundt et al., 2017): Martin Mundt, Tobias Weis, Kishore Konda and Visvanathan Ramesh, *Building effective deep neural network architectures one feature at a time*, preprint arXiv:1705.06778, 2017.

(Mundt et al., 2018b): Martin Mundt, Sagnik Majumder, Tobias Weis and Visvanathan Ramesh, *Rethinking Layer-wise Feature Amounts in Convolutional Neural Network Architectures*, Neural Information Processing Systems (NeurIPS), Critiquing and Correcting Trends in Machine Learning Workshop, 2018.

(Mundt et al., 2019a): Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos and Visvanathan Ramesh, *Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11188-11197, DOI: 10.1109/CVPR.2019.01145, final version: https://ieeexplore.ieee.org/document/8953853, 2019

(Mundt et al., 2019b): Martin Mundt, Iuliia Pliushch, Sagnik Majumder and Visvanathan Ramesh, *Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers?*, Proceedings of the IEEE Computer Society International Conference on Computer Vision (ICCV), First Workshop on Statistical Deep Learning for Computer Vision (SDL-CV), DOI: 10.1109/ICCVW.2019.00098, final version: https://ieeexplore.ieee.org/document/9022417, 2019.

(Mundt et al., 2020b): Martin Mundt, Sagnik Majumder, Iuliia Pliushch, Yong Won Hong and Visvanathan Ramesh, *Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition*, preprint arXiv:1905.12019, under review, 2020.

(Mundt et al., 2020a): Martin Mundt, Yong Won Hong, Iuliia Pliushch and Visvanathan Ramesh, *A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning*, preprint arXiv:2009.01797, under review, 2020.

The overall perspective of how these works are connected and specific scientific contributions are described in detail in the upcoming thesis synopsis. Short general article abstracts are also provided separately ahead of the manuscripts.

In addition, the following publications have been co-authored while writing this thesis, but are not incorporated:

(Hess* et al., 2017): Timm Hess*, Martin Mundt*, Tobias Weis and Visvanathan Ramesh (* equal contribution), *Large-scale stochastic scene generation and semantic annotation for deep convolutional neural network training in the robocup SPL*, Lecture Notes in Computer Science (LNAI), RoboCup 2017: Robot World Cup XXI., vol. 11175, pp. 33-44, 2017.

(Weis et al., 2018): Tobias Weis, Martin Mundt, Patrick Harding and Visvanathan Ramesh, *Anomaly detection for automotive visual signal transition estimation*, IEEE Conference on Intelligent Transportation Systems Proceedings (ITSC), 2018.

Co-advised Theses

While pursuing my doctorate degree I have further had the pleasure to co-advise the following theses:

(Hess, 2016): Timm Hess, *Training Convolutional Neural Networks on Virtual Examples for Object Classification in the Robocup-Environment*, Bachelor Thesis, Department of Computer Science and Mathematics, Goethe University, Frankfurt am Main, 2016.

(Majumder, 2018): Sagnik Majumder, *Neural Architecture Meta-learning via Reinforcement*, Bachelor Thesis, Department Of Electrical And Electronics Engineering And Electronics And Instrumentation Engineering, Birla Institute of Technology and Science, Pilani, 2018.

(Wendland, 2020): Hannah Wendland, *Feature-Wise Expansion of Neural Network Architectures*, Bachelor Thesis, Department of Computer Science and Mathematics, Goethe University, Frankfurt am Main, 2020.

(Jaziri, 2020): Achref Jaziri, *A deep learning approach for semantic segmentation of concrete material cracks from virtually generated images*, Bachelor Thesis, Department of Computer Science and Mathematics, Goethe University, Frankfurt am Main, 2020.

Contents

List of Figures (excluding manuscripts)

Deutsche Zusammenfassung - German Summary

Die Dissertation ist in englischer Sprache verfasst und basiert auf mehreren Publikationen. Der Inhalt der sechs Manuskripte ist im Englischen in einer Übersicht eingebettet, ihr individueller Inhalt zusammengefasst und der Gesamtrahmen beschrieben. Eine kurze deutsche Zusammenfassung zum Rahmen und den Zielen der Dissertation findet sich im Folgenden.

Traditionelles Design von Computersystemen für visuelle Anwendungen beinhaltet im Regelfall eine gründliche Betrachtung der Zusammenhänge zwischen der zu lösenden Aufgabe, der auszuwählenden mathematischen Operatoren und schlussendlich der Evaluierungskriterien, die für das System und dessen Nutzer relevant sind. Beispielsweise in einer Anwendung zur Objekterkennung unter wechselhaften Lichtbedingungen könnte das System dann so gestaltet werden, dass seine Bausteine unabhängig, also invariant, von der Belichtung sind. Dies führt dazu, dass das Resultat des Systems unverändert bleibt. Allerdings wachsen im modernen digitalen Zeitalter auch die Anforderungen an die flexible Nutzbarkeit von Computersystemen für visuelle Anwendungen. Mit steigenden Anforderungen wächst somit auch die erwartete Komplexität, wenn immer schwerere und allgemeinere Aufgaben gelöst werden sollen. Es wird dann davon ausgegangen, dass das ehemals oft noch anschauliche Design des Systems infolge der immer steigenden Anzahl an Variablen der realen Welt unerreichbar wird. Eine stetig wachsende Anzahl modernen Anwendungen setzt deshalb auf maschinelles Lernen. Gerade durch das kürzlich popularisierte und erfolgreiche "deep learning" wird dieses wiederum mittlerweile als Synonym für das Trainieren von tiefen neuronalen Netzen benutzt. Obwohl "deep learning" zunächst nur auf ein Lernverfahren hindeutet, dass sich allgemein durch mehrere hintereinander folgende Berechnungen charakterisieren lässt, impliziert es mittlerweile eine spezielle Form des Lernens. Letzteres wird in englischer Sprache als "end-to-end" bezeichnet und umfasst einen Prozess der automatisch aus Daten lernt und jede einzelne Schicht eines neuronalen Netzes automatisch "von

einem Ende des Systems zum anderen" durch Fortpflanzung von Fehlern aktualisiert. Das sich daraus ergebende Versprechen ist, dass der Nutzer oder System Designer sich nicht um einzelne Schritte kümmern muss, sondern es hinreichend ist einen großen Datensatz zu sammeln und ein entsprechend tiefes neuronales Netz darauf zu trainieren. Als Konsequenz dieser vielversprechenden Entwicklung resultierte nicht nur ein scheinbarer gedanklicher Paradigmenwechsel von detailliertem Modellieren des Systems hin zu einem größeren Fokus auf die Erstellung umfangreicher Datensätze, sondern gleichzeitig auch eine beobachtbare Welle an vorstellbaren visuellen Anwendungen, die ausschließlich auf dem Prinzip des "deep learnings" basieren.

In der Praxis sind die notwendigen Arbeitsschritte für ein erfolgreiches maschinelles Lernsystem weitaus komplexer, als typischerweise in der deep learning Erfolgsgeschichte präsentiert. Diese Dissertation beschäftigt sich damit, die teils nur implizierten oder unterschlagenen Aspekte zu ermitteln, die ungenannten Annahmen hervorzuheben und letztendlich Methoden vorzustellen, die sich mit unmittelbaren Schwächen befassen. Diese aktuellen Defizite lassen sich aus Sicht des Autors darauf zurückführen, dass die Arbeitsschritte im deep learning tendenziell entkoppelt werden und der Erfolg ausschließlich daran gemessen wird, wie akkurat sich ein entwickeltes neuronales Netz in einem definierten statischen Benchmark-Test verhält. Statt aus Sicht des Gesamtsystems und der damit verbundenen Anwendungen Lösungen zu finden, werden einzelne Komponenten mit Hinblick auf die verfügbaren Daten, die Wahl der neuronalen Netzarchitektur, des exakten Lernalgorithmus und dessen Parametern, sowie der Evaluierung und Validierung des fertig trainierten neuronalen Netzes, in Isolation betrachtet und entwickelt. Folglich wurden in dieser Dissertation drei Kernthematiken ermittelt, die ursprünglich in älterer Literatur diskutiert wurden, jedoch im aktuellen Kontext der deep learning Literatur ein erneute Betrachtung erfordern. Spezifisch handelt es sich dabei um drei zuerst unabhängige Aspekte, die allerdings im Gesamtrahmen eines durch maschinelles Lernen gestützten Systems verknüpft und erforderlich sind:

- **Wahl und Flexibilität der neuronalen Netzarchitektur:** Im Wesentlichen besteht eine tiefe neuronale Netzarchitektur aus mehreren Schichten, die jeweils mit einer Wahl von mathematischen Operation und Anzahl an Parametern assoziiert ist. Überwiegend werden in der Literatur vorgeschlagene Architekturen, die sich als empirisch erfolgreich herausgestellt haben, für eine Vielzahl von Anwendungen ohne Änderungen übernommen. Die daraus entstehende Problematik ist, dass diese Architekturen weniger universell sind, als oft gewünscht, und einmal gewählt standardmäßig nicht mehr geändert werden. In Folge sind nicht nur bessere Ergebnisse erzielbar, wenn man die Aufgabe und die konkreten Daten in sein Design miteinbezieht, sondern weiterhin wenn man die Architektur im Laufe der Zeit modifiziert und erweitert.

- **Erkennen unbekannter Daten und Unterdrückung falscher Ausgaben:** Davon unabhängig ob eine neuronale Netzarchitektur in seinen Bausteinen wie oben genannt statisch oder dynamisch ist, ist seine Vorhersage und korrekte Anwendung limitiert auf die Menge der Daten die für das Training im Lernverfahren benutzt wurden. Eine altbekannte Herausforderung für neuronale Netze ist hierbei dass eine Ausgabe auf unbekannten Daten nicht nur selbstverständlich falsch ist, sondern dass diese mit hoher Wahrscheinlichkeit einem bereits bekannten Konzept zugeordnet wird. Als Beispiel klassifiziert ein neuronales Netz das trainiert wurde um Autos von LKWs zu unterscheiden, das bisher unbekannte Konzept eines Zuges mit hoher Sicherheit entsprechend als Auto oder LKW, statt wie vielleicht erwartet auszugeben dass der Inhalt unbekannter Natur ist. Dies hat zur Folge dass deep learning-Systeme zwar hervorragende Ergebnisse bei statischen Benchmark Testsets liefern, jedoch in praktischen Anwendung generell wenig vertrauenswürdig sind.

- **Kontinuierliches Lernen ohne "katastrophales Vergessen" älterer Informationen:** Wenn man das obere Beispiel von Autos, LKWs und Zügen aufgreift, so zeigt sich direkt eine weitere Komplikation in der Verwendung neuronaler Netze. Die Schwierigkeit besteht hierbei im späteren Hinzufügen neuer Klassen z.B. Zügen und deren Unterteilungen zu einem wie im letzten Absatz beschriebenen bestehenden neuronalen Netz, welches beispielsweise schon LKWs und Autos erkennen kann. Verfolgt man dies intuitiv und präsentiert dem neuronalen Netz jetzt z.B. Bilder von verschiedenen Zügen, so müssen wir leider nach dem zusätzlichen Training feststellen, dass jetzt zwar Züge erkennt werden, die im Vorfeld gelernten Autos und LKWs nun aber leider auch als Züge erkannt werden. Dieses Phänomen nennt man "katastrophales Vergessen". Es ist eine Konsequenz dessen dass basierend auf den aktuell vorliegenden Daten die Parameter des neuronalen Netzes überschrieben werden. Das neuronale Netz muss also ständig daran erinnert werden, was es eigentlich schon gelernt hat um dieses Wissen nicht abrupt wieder zu verlieren.

Zusammenfassend stellt sich also die Frage, wie man ein angemessenes neuronales Netz auswählt für eine spezifische Aufgabenstellung, wie man innerhalb dieser Anwendung erkennt was zur Aufgabenstellung gehört und welche Daten noch neue Konzepte enthalten oder eventuell sogar zu einer anderen Aufgabe gehören, und schlussendlich wie man im Laufe der Zeit das neuronale Netz mit neuen Inhalten erweitert. Aus Sicht der Erforschung einzelner Mechanismen könnte man also versuchen, jede einzelne dieser Fragen gesondert zu behandeln. Im Rahmen dieser Dissertation und aus Sicht der Systeme stellte sich allerdings schnell heraus, dass eine ganzheitliche Sichtweise nicht nur Synergien hervorhebt, sondern die Entwicklung von einheitlichen Mechanismen erlaubt, die offenstehende Fra-

gen umfassend behandeln. Der zentrale Punkt dieser Dissertation ist es also auf den existierenden Stärken aufzubauen, aber auch oben genannte Schwächen zu identifizieren, diese in Ihren Abhängigkeiten tiefer zu verstehen und eine gemeinsame Lösung zu finden. Entsprechend ist die Struktur der Dissertation orientiert an den obigen drei Fragestellungen. Kapitel 1 befasst sich somit mit der Wahl und Erweiterbarkeit der neuronalen Netzarchitektur, welche zunächst an populärer Bildklassifizierung verdeutlicht wird, gefolgt von einer konkreten Anwendung in Defekterkennung an Betonbrücken. Kapitel 2 beschäftigt sich mit den komplementären Fragen zur Erkennung von für das neuronale Netz unbekannten Konzepten und dem darauf folgenden kontinuierlichem Lernen. Kapitel 3 fasst letztlich die einzeln entwickelten Aspekte zusammen, betont ihre Wichtigkeit im Rahmen einer umfassenden Literaturrecherche, und verknüpft separat gewonnene Erkenntnisse zu einem gemeinsamen Zusammenhang. Der letztendlich präsentierte umfassende Ansatz ist somit der Beitrag der Disseration zum Fortschritt für das maschinelle Lernen gestützt durch neuronale Netze, in dem eine gemeinsame Lösung für kontinuerliches Lernen, Wahl der neuronalen Architektur und robuste Anwendung mit automatischer Erkennung unbekannter Daten vorgeschlagen wird.

Im Nachfolgenden wird ein allgemeiner kurzer Überblick über die einzelnen Manuskripte, deren Fragestellung und dem geleisteten Beitrag gelistet. Die Titel wurden im englischen Original belassen, jedoch Namen der Kapitel für eine bessere Übersicht zusätzlich sinngemäß übersetzt. Für weitere Details und eine tiefere wissenschaftliche Ausführung wird auf den englischen Hauptteil der Dissertation verwiesen.

Kapitel 1: Designing Dynamic Deep Neural Network Architectures through Meta-Learning and Representational Capacity Expansion (Design dynamischer tiefer neuronaler Netz Architekturen mit Hilfe des Meta-lernens und Kapazitätserweiterung in Bezug auf gelernte Repräsentationen der Daten)

- ***Building effective deep neural network architectures one feature at a time:***
  Neuronale Netze werden oft genutzt um anhand der Daten einer Aufgabenstellung abstrakte Repräsentationen zu lernen. Im praktischen maschinellen Lernen nutzt ein Ingenieur typischerweise für diesen Zweck vorgefertigte Architekturen. Solch eine neuronale Netzarchitektur ist aus mehreren Blöcken oder Schichte zusammengesetzt, wobei jede einzelne Schicht mit einer Wahl bezüglich seiner "Breite" verknüpft ist. Für eine feste Gesamtzahl an Schichten, spiegelt diese Breite in simplifizierter Form die Komplexität wieder. Ist diese zu niedrig, sehen wir uns mit der Gefahr konfrontiert, dass die notwendige Kombination aus Repräsentationen, um die Aufgabenstellung zu lösen, nicht gelernt werden kann. Ist diese zu hoch, so ist das Trainingsverfahren

langsam und das zusätzliche Risiko besteht, dass Datenpunkte auswendig gelernt werden und somit keinen allgemeinen Nutzen liefern. Der konventionell beschriebene Weg der Literatur ist deshalb sehr große neuronale Netze zu trainieren, und darauffolgend unnütze Repräsentationen, die nicht zur Aufgabe beitragen, wieder zu entfernen. In diesem Artikel wird eine umgekehrte Herangehensweise vorgeschlagen. Es wird empirisch gezeigt dass mit einem sehr kleinen neuronalen Netz begonnen werden kann, um dieses dann adaptiv wachsen zu lassen, um die notwendige Komplexität für die spezifische Aufgabenstellung zu erreichen. Die vorgestellte Methode erleichtert dem Anwender das Ermitteln der neuronalen Netz Schichtbreite. Im Rahmen der Validierung der Experimente wird somit ein Teil der Fragestellung zur Auswahl einer passenden neuronalen Netzarchitektur adressiert.

- ***Rethinking Layer-wise Feature Amounts in Convolutional Neural Network Architectures:***
  In praktischen Anwendungen von tiefen neuronalen Netzen ist eine Faustregel aus der Vielzahl an Experimenten entstanden. Diese Faustregel dient als Leitfaden in Bezug auf das Design von neuronalen Netzen, die hauptsächlich auf mathematischen Faltungen basieren. In solch einem Netz werden mehrere Faltungen in Reihe geschaltet, wobei der Gestalter des Netzes sowohl die Anzahl an Faltungen in Reihe, als auch die Anzahl der parallel geschalteten Berechnungen in jeder Schicht der Hierarchie bestimmen muss. Die ungeschriebene Regel besagt hier, dass die Anzahl der parallelen Berechnungen mit der Tiefe des Netzes ansteigen sollte. Dies ist inspiriert von der Hypothese, dass ein neuronales Netz zunehmend aufgabenspezifische Repräsentationen in tieferen Schichten lernt. Im Umkehrschluss wird davon ausgegangen dass weniger Operationen in niedrigen Schichten benötigt werden, da spekuliert wird, dass diese sehr elementare Merkmale lernen, so wie Kanten oder Farben in visuellen Systemen, und somit generell nutzbar für eine Vielzahl von erdenkbaren Aufgaben sind. Der vorgestellte Artikel hinterfragt diese Hypothese und insbesondere die daraus resultierende Faustregel, indem für populäre Datensätze zur Bildklassifizierung die Anzahl der maximal erlaubten lernbaren Repräsentationen in verschiedenen Schichten der neuronalen Architektur untersucht wird. Aus den spezifischen Experimenten geht hervor, dass ein umgekehrtes Phänomen zur beschriebenen Faustregel empirisch zu bevorzugen ist. In Bezug auf die überspannenden Aspekte des maschinellen Lernens werden letztendlich somit Bedenken zum aktuell überwiegenden Vorgehen geäußert, identische neuronale Netzdesigns für verschiedenartige Aufgaben zu benutzen.

- ***Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset:***
  Ein Datensatz für die Erkennung und Klassifizierung von Betondefekten in ziviler Brückeninfrastruktur wird vorgestellt. Wohingegen vorherige Datensätze einen Schwerpunkt auf Risse als Hauptgefahr setzen, erweitert der eingeführte Datensatz dies um Materialzertrümmerung (engl. spallation), Kalziumablagerungen (engl. calcium leeching), offen liegende Bewehrungsstäbe (engl. exposed reinforcement bar) und Korrosion. Diese Kategorien können alle gleichzeitig in einem Bild vorkommen. Die daraus resultierende Aufgabenstellung der Klassifizierung gemeinsam vorkommender Defekte wird aus Sicht tiefer neuronaler Netze untersucht. Es wird beobachtet dass Vorschläge für neuronale Netzarchitektur aus vorangegangener Literatur in Evaluationsexperimenten mit dem eingeführten Datensatz nicht vergleichbare Vorteile wie in den Originalexperimenten zeigen. Im Kontrast hierzu bietet das Metalernen von neuronalen Netzen, das heißt das Lernen der Bausteine der Architektur und deren Komposition selbst mit Hilfe eines zusätzlichen Lernmechanismus, der die Struktur für eine spezifische Aufgabe optimiert, eine bessere Lösung bezüglich der Genauigkeit und verringerten Größe der neuronalen Architektur. Im Rahmen der einzelnen Aspekte des maschinellen Lernens wird bekräftigt, dass ein solides neuronales Netzdesign ein explizites Hinzuziehen der Aufgabenstellung und ihrer Daten erfordert. Es sollte also nicht allgemein davon ausgegangen werden, dass ein einzelnes statisches Design ausreicht, um verschiedene Aufgaben zu lösen.

Kapitel 2: Enabling Open Set Recognition and Continual Learning in Deep Neural Network Architectures (Erkennung offener Mengen und Befähigung zum kontinuierlichen Lernen in tiefen neuronalen Netzen)

- ***Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition:***
  Eine wohlbekannte Herausforderung für tiefe neuronale Netze ist es ungesehene unbekannte Eingaben von den Trainingsdaten zu unterschieden und die aktuelle Unfähigkeit zum kontinuierlichen Lernen zu überwinden. Ohne zusätzlichen Mechanismen weist ein tiefes neuronales Netz ein unbekanntes Beispiel einem gelernten Konzept mit hoher Wahrscheinlichkeit zu. Sollte man dieses Unwissen lösen wollen, indem man die Parameter anhand neuer Beispiele aktualisiert, so überschreiben diese wiederum älteres Wissen. In diesem Artikel wird argumentiert, dass diese Schwierigkeiten verbunden sind und gemeinsam mit einem einzelnen vorgestellten Mechanismus für tiefe generative neuronale Netze angegangen werden können. Solch ein

generatives neuronales Netz lernt explizit eine Kodierung der Wahrscheinlichkeitsverteilung der observierten Daten. Zuerst werden Eingabedaten in einzelne Faktoren zerlegt, bevor diese dann rekombiniert werden, um Daten zu erzeugen. Im vorgeschlagenen Verfahren wird gezeigt, dass durch Messen der Distanz zu diesen generativen Faktoren der Trainingsdaten eine Lösung gefunden werden kann, um sowohl kontinuierliches Lernen, als auch Erkennung von unbekannten Beispielen zu ermöglichen. Um bereits gelernte Daten nicht zu vergessen, können ähnliche Datenpunkte zur ursprünglichen Datenmenge mit Hilfe der generativen Faktoren erzeugt werden und dem neuronalen Netz zur Bekräftigung nochmals präsentiert werden. Gleichzeitig kann eine falsche Ausgabe für unbekannte Daten verhindert werden, indem diese als unbekannt durch große Distanzen zu bereits bekannten Faktoren erkannt werden. Der Artikel betont das notwendige Zusammenspiel verschiedener Aspekte des maschinellen Lernens, insbesondere die Abhängigkeit zwischen robusten Ausgaben und kontinuierlichem Lernen.

- ***Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers?:***
  Der Schwerpunkt des Artikels liegt auf der Auswirkung der Wahl des neuronalen Netzes und der Wahl der Metrik auf die Fragestellung ob neue unbekannte Datenpunkte als solche erkannt werden können oder mit bekannten Konzepten verwechselt werden. Es werden Experimente durchgeführt um simple Heuristiken, basierend auf Schwellenwerten einer Klassifizierungsausgabe, mit gemessenen Distanzen im Raum der gelernten Repräsentationen zu vergleichen. Die Untersuchung ist erweitert durch optionale Schätzung von Unsicherheit, d.h. die gemessene Intensität der Schwankung in der Ausgabe, wenn Berechnungen mit einer zufällig ausgewählten Zahl an Repräsentationen wiederholt wird. Die experimentelle Analyse ist weiterhin gekoppelt an die Wahl des neuronalen Netztyps im Sinne der zugrunde liegenden statistischen Modellierung. Im Wesentlichen werden diskriminative Modelle mit ihrem generativen Gegenstück gegenübergestellt. Im Kontext neuronaler Netze geben erstere eine simple Ausgabe einer Klasse nach Eingabe ein Beispiels aus, letztere berücksichtigen hingegen weiterhin die Frage wie die Datenpunkte entstanden sind. Als Resultat der Experimente übertrifft der vorgeschlagene Ansatz, der den Generationsprozess berücksichtigt und Ähnlichkeit neuer Beispiele anhand von Distanzen zur gesehenen Verteilung der Trainingsdaten beurteilt, standard neuronale Netzklassifizierung in Hinblick auf Erkennung von unbekannten Daten. Aus Perspektive des allgemeinen Vorgehens im maschinellen Lernen verstärkt diese Arbeit somit die

Notwendigkeit eines Systemansatzes und zeigt auf, dass eine robuste Anwendung inhärent gekoppelt an die Wahl des neuronalen Netzmodells ist.

Kapitel 3: Consolidating Viewpoints: Designing Neural Networks for Continual, Active Learning in an Open World (Zusammenführen der Blickwinkel: Design von neuronalen Netzen für kontinuierliches, aktives Lernen in einer offenen Welt).

- ***A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning:***
  Der Artikel präsentiert die übergreifende Perspektive, dass verschiedene individuell behandelte Aspekte des neuronalen Netz spezifischen maschinellen Lernens gemeinsam betrachtet werden müssen. Er vertritt die Position, dass typischerweise separierte Elemente synergistisch sind und untermauert dies durch einen breiter gefächerten Literaturüberblick. Das zentrale Thema ist dabei eine Brücke zu bilden, zwischen den Herausforderungen des kontinuierlichen Lernens mit neuronalen Netzen, des Problems gelernte Konzepte von beliebigen Datenbeispielen für eine robuste Anwendung zu unterscheiden und die essenzielle Frage welche Daten für das Training benutzt werden sollten. Ein gemeinsames konzeptionelles Gerüst für diese Aspekte wird mit Hilfe der zuvor eingeführten generativen neuronalen Netze vorgestellt. Weil diese Modelle es erlauben, die Verteilung der Trainingsdaten zu approximieren, kann ein einzelner Kernmechanismus vorgestellt werden, um die beschriebenen drei Herausforderungen zu überwinden. Empirische Resultate zeigen, dass dieser Mechanismus einzelne Techniken, die ausschließlich für eine Fragestellung vorgeschlagen wurden, in der individuellen Problemstellung verbessert und gleichzeitig einen gemeinsamen übergreifenden Rahmen bietet. Dies wird demonstriert durch experimentelle Verbesserung der Genauigkeit in kontinuierlich lernenden neuronalen Netzen zur Bildklassifizierung, verbesserte Genauigkeit wenn Trainingsdaten vom Netz selbst ausgewählt werden oder wenn die Reihenfolge der Aufgaben vom neuronalen Netz selbst bestimmt wird, sowie deutlich verminderter Abfall der Genauigkeit und somit robustere Anwendung wenn korrupte Daten eingeführt werden. Der Artikel baut somit auf die Erkenntnisse vorangegangener Literatur und den vorherigen Artikeln der Dissertation auf und vereint diese. Er stellt die Notwendigkeit einer Systemperspektive heraus und demonstriert deren Vorteile aus Perspektive tiefer neuronaler Netze.

THESIS SYNOPSIS

Preamble: Models, Feature Engineering and Data-driven Approaches

Many of the traditional computer vision works build systems through inclusion of explicit invariant operators that guarantee an unchanging output under change of the condition to which they are invariant to. A popular choice for an outdoor computer vision application could be the inclusion of a photometric invariant that separates shape from illumination (Schmid and Mohr, 1997; Narasimhan et al., 2003), in order to make sure that an algorithmic prediction remains the same even when the lighting conditions vary. A similarly common choice could be invariance to scale, for applications where an object to be recognized appears at perceived different sizes due to varied distance to the camera. Ideally, a computer vision system is thus desired to produce a correct unaltered output in independence of any such nuisance variation that we do not explicitly care about, see Chin and Dyer (1986); Besl and Jain (1985); Mundy and Zisserman (1992); Mumford et al. (1994) for surveys. Often the specific application does not require full invariance or a mathematical expression for a full invariant is not trivially constructable. Quasi-invariance can then suffice to form hypotheses. Such quasi-invariance serves as an approximation with respect to full invariance, where the output now remains constant only for a specific, yet practically sufficient range of transformations, see Binford and Levitt (1993) for an overview. Based on this perspective, coupled with a view that is cohesive with rigorous engineering principles, performance characterization (Petkovic, 1989; Haralick, 1992) then conducts analyses of algorithms' behavior under conceivable perturbation models (Ramesh and Haralick, 1992a,b), see Thacker et al. (2008) overview and best practices.

However, applications have emerged where an ever increasing amount of variations in acquired data has led to an expected increasing necessity for larger complexity. If we take for instance all the possible variations of environmental conditions coupled with the generally unconstrained design of man-made objects composed of materials such as plastic, it

is assumed that these factors of variations can no longer be fully specified in an upfront system design. In an age where data has become a commodity, the advent of deep learning (DL), or rather resurgence popularized by the empirical successes on large scale computer vision tasks (LeCun et al., 1998; Krizhevsky et al., 2012; Everingham et al., 2014; Russakovsky et al., 2015), has thus shifted the focus to data-driven approaches. Initially, this has yielded works that attempt to combine or partially include the above perspectives with the design of data-driven pipelines to various degrees. For example, Girshick et al. (2014) employed traditional modules with illumination and scale invariants through inclusion of selective search preprocessing (Uijlings et al., 2013) as an initial step to further processing with neural networks. He et al. (2014) included spatial scale pyramids in deep neural networks. Analogously, scattering convolutional neural networks (Bruna and Mallat, 2013) employ a first layer of scattering wavelets (Mallat, 2012), that feature specific group invariants to e.g. rotation. These are just a few of the technical examples that combine expert designed modules with purely data-driven techniques. Nevertheless, the presumption that the respective invariance can simply be learned if enough data featuring the corresponding variation is available seems to have taken over lately and diffused into every conceivable application. In particular with deep neural networks, it is generally believed that a large amount of parameters provides enough representational power to express even the most complex concepts, leading to a task's solution if we simply allow these hidden variables to be learned through repeated iterative updates on enough acquired data. Whereas handwriting recognition has traditionally been pursued through the construction of an explicit adaptive likelihood model that takes into account expert knowledge on locations and deformations along splines of digits (Hinton et al., 1992), it is therefore now perceived as a solved challenge by training deep neural networks on ample amounts of annotated data (LeCun et al., 1998). Object detection that probabilistically factors in the physics of image formation, geometry, illumination and sensors (Tsin et al., 2001) has largely been replaced by end-to-end learnable pipelines (Girshick et al., 2014; Ren et al., 2017). Texture recognition, customarily addressed with expert constructed three dimensional assemblies of so called textons (Leung and Malik, 2001), has similarly been superseded by the DL approach (Cimpoi et al., 2015; Andrearczyk and Whelan, 2016). The purely data-driven approach is now favored in areas such as crack defect recognition (Shi et al., 2016; Kim et al., 2018; da Silva and de Lucena, 2018; Mundt et al., 2019a), where accounting for the fractal geometry was predominant in previous applications (Maaruf et al., 1993; Cao and Ren, 2006; Farhidzadeh et al., 2013). Claims even go as far as deep neural networks being able to acquire physical intuition by observing physical processes such as towers toppling due to their assembly, the blocks' mass and gravity (Lerer et al., 2016), something traditionally

described through elaborate generative factors in conjunction with e.g. graphics renderers to obtain the respective visual manifestations (Battaglia et al., 2013).

As the accomplishments of deep learning are typically perpetuated by their success on limited static benchmark datasets, see e.g. Tsotsos et al. (2019) on a discussion on the effect of dataset statistics and their potential mismatch with common traditional computer vision assumptions, this development is not perceived equally amicably by everyone. It has at the same time sparked an increasing amount of controversy, centered around the often campaigned superiority and advocated dominance of these heavily data relying techniques over traditional approaches. Several public debates were initiated, with the intent to voice valid concerns and critique about the limitations of respective viewpoints (Marcus and Bengio, 2019; Marcus and Lange, 2020). Although designed to connect and fuel the way forward, respective mindsets are however often presented as opposing one another and almost maliciously interpreted as nefarious denunciation or dismissal of other views. The central dispute seems to narrowly revolve around an asserted lack of interpretability of deep learning versus the attributed intolerable amount of human investment into engineering an expert system (Marcus, 2018; O'Mahony et al., 2019). This discussion can be further contextualized with respect to the chronicle of DARPA and Launchbury (2017) on the *three waves of AI* as the distinction between 1: descriptions through hand-crafted knowledge and 2: categorization through statistical learning. The core question is then how to proceed to the next stage 3: explanations through contextual adaptation. Just as many deep learning works assign a negative connotation to the term "hand-crafting", so do modern expert systems imply that deep learning approaches are inherently incomprehensible. In the subjective view of this thesis' author, such arguments can be attributed to the overloading of historically grown technical terms, the attached premises and the ambiguity in the terms' community specific use. Whereas ideally scientific communication should always keep facts in a grounded perspective, promoting a specific angle can be accompanied with a certain amount of in-transparency, perceived as over-claiming or even as advocacy of a technique as a "universal solvent" (Marcus, 2018). The resulting technical polymorphisms can thus serve as red herrings, that form the basis for a distorted narrative and are easily subject to misunderstandings and dispute.

One such polymorphism lies in the use and associated expectation of the concept of a *model* and the corresponding adoption in deep learning. It is clear that the notion of a model historically seems to refer to an expert designed, frequently fully parametric, system (DARPA and Launchbury, 2017). This can be rooted in causal generative factors and physics-based probabilistic approaches (Lipton, 2016; Marcus, 2018) or simply refer to computational pipelines where the task specification and context is taken into account (Petkovic, 1989;

Haralick, 1992; Ramesh and Haralick, 1992a,b; Thacker et al., 2008). Even though the computational pipeline may be deep, i.e. comprised of multiple subsequent operations, there is an anticipation of human interpretable behaviour of the model and a resulting robustness in design. Whereas deep learning is generally described as "computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction" (LeCun et al., 2015), the term however is regularly used to imply the assembly of large amounts of layers in opaque neural networks, also referred to as the *model*. As such, the wide-spread imputation is that the modern deep learning landscape is riddled with such black boxes. They are presumed to work extremely well in practice and to require significantly less expertise, as they rely on only few key mechanisms, such as training neural networks through backpropagation of errors (Rumelhart et al., 1986). The lack of human interpretability is either tolerated in favour of empirically improved benchmark results or post-hoc justified through visualization and introspection techniques (Bach et al., 2015; Simonyan et al., 2014; Erhan et al., 2009; Olah et al., 2018; Montavon et al., 2018).

So do these neural network *models* generally overcome the design efforts of traditional *models* and does this always come at the expense of interpretability?
Reality is rarely as straightforward as this painted simple picture and answering the question turns out to be far from trivial. In the process of writing this thesis' manuscripts, the realization quickly surfaced that techniques from traditional modelling that are attributed to lack "large scale" are not necessarily mutually exclusive with data-driven approaches. Conversely, deep learning is significantly more laborious for novel tasks than typically portrayed in the world of benchmarks, with many of the failure modes and modelling assumptions hidden or shifted to other aspects. Inspired by this mindset and the previously learned lessons from past literature, this thesis has set out to walk initial steps on a path towards understandable and interpretable deep neural network, potentially hybrid, models. At the time of writing this thesis this road is without doubt long, as the question of bottom-up assembly of hybrids and solving the essentially impossible to objectively define question of what constitutes deep neural network interpretability remain largely open. That is, a great amount of long-term research is still involved for the fundamental questions of whether a neural network needs to resemble human decision making (Ridgeway et al., 1998), whether causal structure in the data needs to be uncovered (Athey and Imbens, 2016), whether linear models can be attributed with interpretability (Lou et al., 2013) or whether each operation at every level of architecture compositionality requires grounding in intuitive theories of physics and psychology (Lake et al., 2017). Yet, the existence of many hidden assumptions in the machine learning workflow, that are frequently oversimplified or overlooked, also leaves room for a complementary perspective. Instead of trying to explicitly fuse traditional computer vision

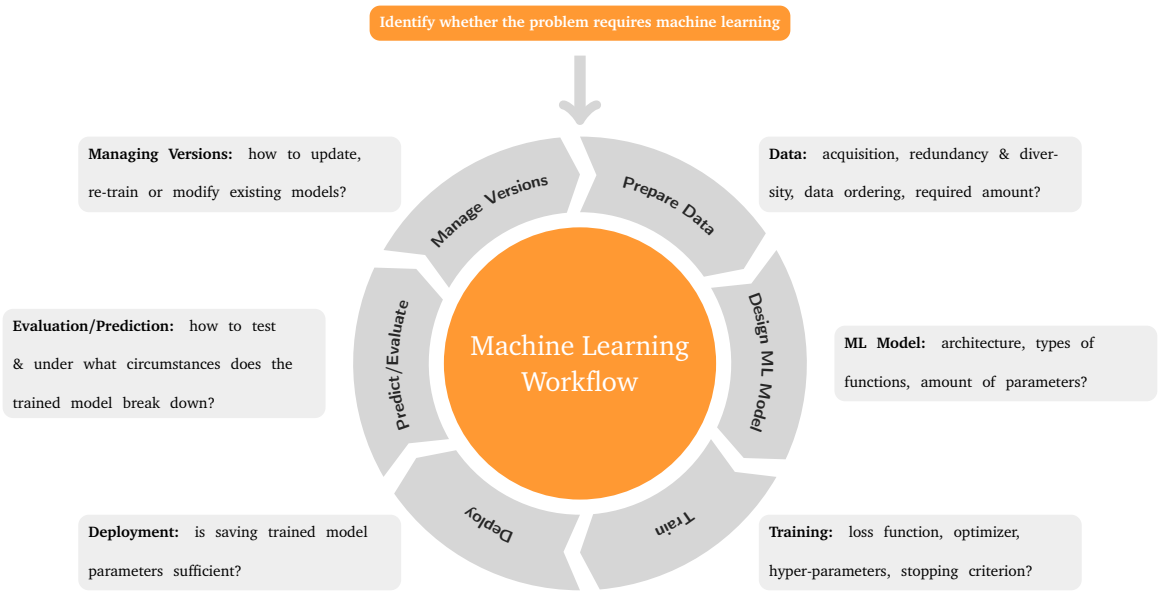


Figure 1: Core machine learning workflow in the spirit of Google Cloud (2020) illustrations. The accompanying articulated questions have been included to highlight the level of intricate detail that is involved in a revolution of the supposedly simple cycle.

techniques into deep learning systems, we can attempt to bring the common use of deep neural networks closer to that of traditional models. The main driving factor of this thesis is thus to shine light into DL assumptions, associated failure modes, and lift them to a certain extent to advance the current ways in which ML models can be exploited. The necessity for the latter has stemmed from the requirements of an accompanied real-world application of concrete infrastructure defect detection, that traditionally is not addressed through the use of machine learning due to safety prerequisites and lack of large scale datasets for machine learning.

To provide further detail, figure 1 shows a typical machine learning cycle on the basis of the illustrations of Google Cloud (2020). It is composed of seven steps that are supposed to pave the way towards any ML application, potentially by using correspondingly designed and standardized cloud software platforms. As soon as the task's suitability for machine learning has been determined, the cycle starts with preparation of the data, coding of the ML model and its training. This could be as easy as labelling the data with the desired human labels, loading an ML architecture description from the literature and employing an off-the-shelf stochastic gradient descent optimization algorithm. Once this is complete, the model can be saved and deployed and a dedicated test set for evaluation can be used to determine the model's empirical performance in order to decide whether it improves in a desirable direction towards satisfying previously set goals. The procedure is then iterated.

This is the promise and pitfall of modern deep learning: choose a model, plug in the data and iterate through further additions of data and model training.

Unfortunately this simplified view hides the many assumptions that are required to traverse the cycle. Some of the intricate details and questions are specified at each stage of the process in the figure. These encompass aspects reaching from data acquisition, required data amounts, neural network design, size and training hyper-parameters, all the way down to questions of when a trained DL model can be deemed successful, when it is expected to break down and how the model can be updated continuously without restarting the cycle from scratch at every iteration. To provide a more detailed account of the assumptions that go into a DL system that have not vanished, but have merely been shifted to other aspects that are similarly laborious to traditional model pipelines, a few examples are given in the following: ever since the proposal of convolutional neural network (CNN) LeNet (LeCun et al., 1998) and the success of Alexnet (Krizhevsky et al., 2012), almost every year has sprouted a new state-of-the-art architecture, involving networks in networks (Lin et al., 2014), deeper architectures (Simonyan and Zisserman, 2015), inclusion of skip-connections (He et al., 2016), architecture width vs. depth trade-offs (Zagoruyko and Komodakis, 2016) or dense connectivity patterns across all layers of the deep hierarchy (Huang et al., 2017). Similar architecture attempts have been made by improving activation and pooling functions (Goodfellow et al., 2013; Lee et al., 2015), replacing pooling with learnable convolutions (Springenberg et al., 2015) and normalizing activations through data mini-batch statistics (Ioffe and Szegedy, 2015). Efforts to improve stability and speed of training have been made through the proposal of adaptive optimization steps (Kingma and Ba, 2015), dropout regularization (Srivastava et al., 2014), intricate learning rate scheduling (Zagoruyko and Komodakis, 2016; Loshchilov and Hutter, 2017) and the proposal of weight initialization schemes (Glorot and Bengio, 2010; He et al., 2015).

In practice, when faced with a novel application, many of the above techniques and proposed architectures do not live up to their promise and do not show the same improvements that were conveyed in the original works' benchmark analyses (Hendrycks and Dietterich, 2019; Mundt et al., 2019a). When the trained and deployed ML model encounters data that deviates statistically from the observed training distribution the resulting prediction is generally erratic (Matan et al., 1990; Szegedy et al., 2014). When the ML model is trained continuously without repeatedly exhibiting previously seen data instances, it is bound to catastrophically forget all acquired information (McCloskey and Cohen, 1989; Ratcliff, 1990; French, 1992). To give a practical real-world example, suppose that we wish to train a neural network to distinguish different animals. We start by showing it images of dogs and cats and rapidly obtain a desired accuracy. Unfortunately, if shown images of other unseen

animals such as horses or owls, the neural network will now tell us that these animals are dogs and cats with an unreasonably high likelihood. Even more disastrously, if we now wish to update our neural network continuously to also learn about owls and horses, we need to constantly remind it of dogs and cats, as otherwise this pattern will occur in reverse and the latter suddenly become synonymous with our newly introduced animals. Surely, in the real world we don't want our vision system to immediately forget everything it has learned when we inject pieces of new information, and we certainly do not want our systems to fail at the sight of anything it hasn't previously seen.
Consequently, this thesis presents an attempt at shedding insights into, alleviating and mitigating these former aspects. The remainder of the thesis addresses individual open questions of the ML system workflow on dataset construction, neural network architecture selection, continual deep neural network learning, recognizing unknown examples and preventing overconfident false predictions with deep neural networks. Conclusively, these questions are then fused into a common larger perspective. A detailed synopsis and the specific contributions of this thesis are described in subsequent sections.

Deep Learning Design Challenges and the Convergence of Complementary Threads

The necessity to consider the entire machine learning workflow in this thesis originated from an associated practical application concerned with locating and classifying defect anomalies in concrete civil infrastructure. The open questions surrounding the machine learning workflow, as articulated in the preamble, have emerged due to the perception of a persisting disconnect between the seemingly rapid progress that is claimed in the current DL benchmarking landscape and the transfer and usefulness of such designs for practical systems. As will be argued extensively in the forthcoming thesis' manuscripts, this is assumed to be a result of the predominance of simple closed world benchmarks as a direct proxy for made advances in the field. Here, the closed world refers to the practice of constructing a limited dataset, separating a certain percentage as a dedicated test set, and hence judging an algorithm's or system's progress based on empirical performance curves within this context. Whereas many advances have certainly been achieved, the observation that such evaluation protocols are overly simplified and often unrepresentative of true application desiderata had to be made again and again throughout the course of this thesis. As such, a solid proportion of developed DL indeed seems to fit the assertion of being limited to black box data driven solutions that lack intuitive understanding and transfer to uncharted domains.

Without delving into full detail at this point, assume for an instant the later investigated task of detecting cracks on concrete material surfaces. The traditional computer vision approach would generate hypotheses on the existence of a crack based on e.g. their unique fractal like geometry, that is in stark contrast to the homogeneity of man-made edges (Maaruf et al., 1993; Cao and Ren, 2006; Farhidzadeh et al., 2013). A respective computer vision model could then be further robustified through the use of illumination invariant operator techniques to assure that the output remains consistent independent of global illumination and lighting directional changes, based on e.g. ratio computation between individual color channels (Funt and Finlayson, 1995; Nayar and Bolle, 1996; Nagao and Grimson, 1998). In deep neural network (DNN) based data-driven approaches, we do not know if such criteria are explicitly encoded and thus cannot derive similar behavioural guarantees. If we present a previously unseen image that contains a different non-fractal man-made edge, this might as well be considered as a crack due to differences in color, changes in illumination or any other unprecedented change in the environment. This is because we typically have little control over the exact invariants and the form of the features encoded in DNNs, outside of attempting to include as much as possible variety in the constructed dataset. In fact, various research has shown that it is still largely unclear how DNNs represent concepts and arrive at their predictions. Geirhos et al. (2019) have shown that DNNs appear to give more weight to textures in decision making in comparison to geometry. Ilyas et al. (2019) found that adversarial perturbations, i.e. adding minute and imperceivable noise to an image that typically leads to a complete system breakdown (Szegedy et al., 2014), can be sufficient in training a DL model. Lapuschkin et al. (2019) have investigated the possibility of DNNs exploiting so called "clever hans predictors", that is arriving at a prediction by memorizing something unique in an image, such as a photographer's signature in the corner of images taken of horses in the majority of the constructed dataset.

This argument can in principle be taken ad absurdum, as shown in a quote from a recent paper titled "Why do deep convolutional networks generalize so poorly to small image transformations?" by Azulay and Weiss (2019): "we show that the convolutional architecture does not give invariance since architectures ignore the classical sampling theorem, and data augmentation does not give invariance because the CNNs learn to be invariant to transformations only for images that are very similar to typical images from the training set." and similar contemplations by Kayhan and van Gemert (2020): "we show that CNNs can and will exploit the absolute spatial location by learning filters that respond exclusively to particular absolute locations by exploiting image boundary effects". In the same spirit, Hendrycks and Dietterich (2019) have demonstrated the lack of robustness of DNNs to a variety of conceivable common image perturbations and corruptions on a large empirical scale. These latter insights are a direct consequence of the closed world assumption, that is the idea

that evaluation needs to be conducted exclusively in a constrained environment, with the later expectation that architecture settings and features deduced from a specific dataset will somehow generalize and transfer to new other data distributions, domains and applications (Yosinski et al., 2014; Oquab et al., 2014), see e.g. Pan and Yang (2010); Weiss et al. (2016) for surveys on the prevalent assumption of DNN architecture and feature transferability. In practice however the discrepancy between closed world design and evaluation, and real open world application and usefulness forms the basis for a cascade of imperative to solve challenges.

It is undeniable that many of the recently developed deep neural network architectures have come with significant improvements across many computer vision benchmarks (LeCun et al., 1998; Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016; Zagoruyko and Komodakis, 2016; Huang et al., 2017). However, these benchmarks have been investigated for at least a decade and the question of how suitable the static architectures are for other datasets and whether similar benefits can be observed remains open. Much more importantly, when moving from the typically investigated object classification benchmarks to tasks concerning for instance textures, it can initially be unclear if commonly developed assumptions on architecture design hold, or respectively how to design a solid architecture.

Suppose that an initial neural architecture design, typically consisting of multiple layers, their precise mathematical operation and an allocated amount of learnable parameters, has been found and it is expected to perform well in practice. We are now immediately confronted with the next challenge in the cascade: there is no obvious indication of when our trained model is going to fail. To worsen this, a neural network is well known to predict falsely for unseen unknown data (Matan et al., 1990). To put this into perspective, say our above defect classification algorithm distinguishes the anomaly into cracks or exposed reinforcement bars. If a novel input image contains a third, currently unknown anomaly such as corrosion stains, then this is not only bound to be attributed to one of the former two categories, the neural network will additionally do so with considerable confidence. This is the contrast between the closed and the open world. Recognizing the latter to prevent nonsensical outputs is known as open set recognition.

Let's take our thought experiment a little further and assume that there now exists a suitable architecture and a way to protect it from nonsensical outputs by identifying scenarios on which it cannot perform well yet. Using a devised mechanism and the supervision of a human, we now employ our system in practice and set aside any data for which we have identified a failure mode. Naturally, our goal is now to improve our system and further optimize our neural network to find remedies for the unseen scenarios and include previously unconsidered concepts, such as the additional corrosion stain in our concrete defect appli-

cation. Naively, we just include these examples into the next step of our stochastic gradient descent based backpropagation optimization algorithm and unfortunately now find that the DNN performs great on the new data, but seems to have forgotten what we have previously concentrated on. This well-known phenomenon is referred to as catastrophic interference (McCloskey and Cohen, 1989; Ratcliff, 1990), a result of the combination of greedy unidirectional parameter updates coupled with the entangled nature of the dense neural network representations (French, 1992). It seems that we have to restart our process from scratch, start over by identifying a suitable architecture, as it is static in nature and doesn't allow for changes on the fly, train on the entire data set once more, and ultimately again identify what concepts are known and which are not.

Among many of the other questions compiled in figure 1, these issues have been identified many decades ago and had to be rediscovered and their persistence validated in the context of DNNs. Although the previous paragraphs may evoke a rather critical perspective, the intent of this thesis is not to dwell on these shortcomings, but rather build on the myriad of deep learning accomplishments. Accordingly, the above three essential challenges of how to design neural network architectures for a novel task, such that they can also be trained continuously as well as recognize unseen unknown data to provide a signal for their intrinsic limitations in data-derived predictions, are the subject of the investigations in the remainder of this thesis. When viewed in isolation of the preceding preamble and introduction, each of the open questions could initially appear as worthwhile of being the focus of individual segregated works. However, they are all part of the same process of devising credible machine learning systems. This is reflected in the thesis structure and the corresponding manuscripts, where multiple threads of the ML workflow have originally been loosely coupled and pursued in parallel. A gradual fusion of the strands has then ultimately led to the proposition of an integrated perspective.

The outline, in perspective of the overall machine learning workflow of figure 1, is presented in figure 2. Out of the six central iterative development stages, four key phases are involved to various degrees: preparation of data, designing the ML model, training, and prediction and evaluation. The aspects of deployment and management of versions have been omitted from the diagram, as the thesis has not further investigated questions such as how to effectively store ML models, how to compress representations, speed up computation or how to practically version individual models.

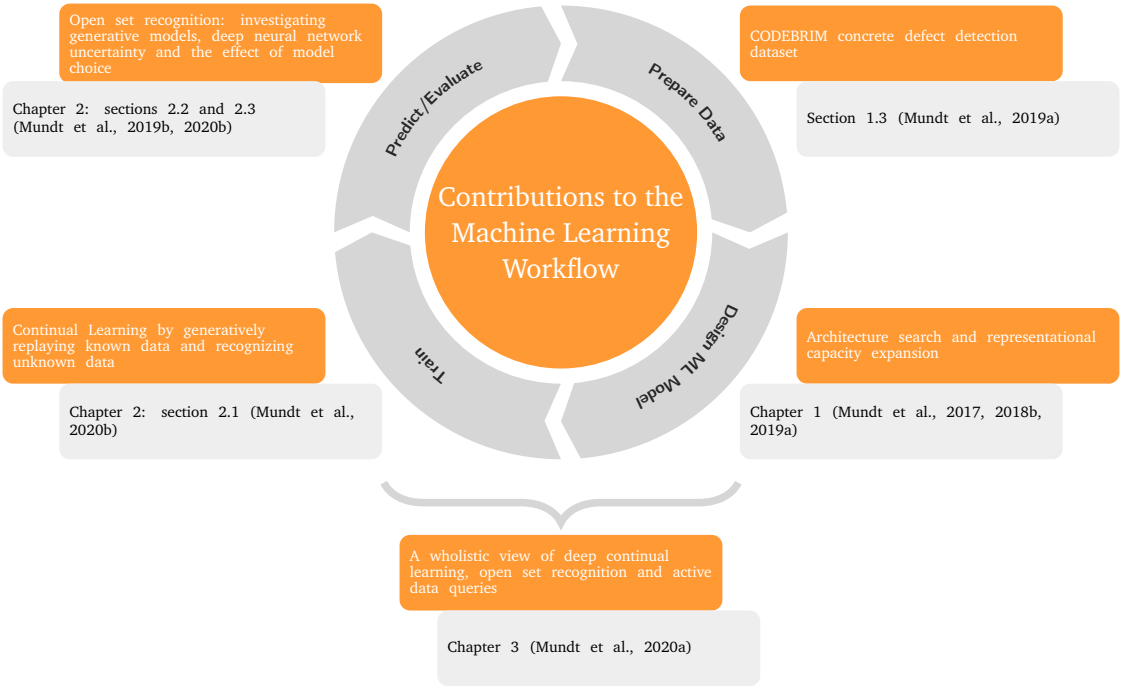


Figure 2: Key components of the machine learning workflow that have been examined in this thesis. The two phases of model deployment and versioning, as presented in figure 1, have not been investigated further and are thus ommited from this diagram. Chapter 1 focuses largely on the question of how to find and evolve suitable deep neural network architectures, whereas chapter 2 concentrates on how to enable these architectures to learn continually and identify unseen unknown examples. Chapter 3 conjoins the individual works and presents a consolidated viewpoint, its embedding in the larger ML literature landscape, a corresponding deep neural network based framework and empirical validation. Authored manuscripts have been assigned to the phase of the workflow cycle that they can best be attributed to. However, note that many of the at the time investigated challenges and questions are heavily interlaced, particularly when viewing them in retrospect of chapter 3.

Consequently, chapter 1 is concerned with the design of ML models and covers works where progressive growth of architecture parameters to find a suitable representational capacity for a task is suggested (Mundt et al., 2017), the feature distribution across the deep neural network hierarchy is examined (Mundt et al., 2018b), and reinforcement learning based neural architecture search in comparison with static literature proposed DNNs is empirically investigated in the context of a proposed concrete defect detection and classification dataset (Mundt et al., 2019a).

Chapter 2 is concerned with open set recognition and its indispensability for continual learning. First a continual learning capable deep neural network is presented, that leverages an open set recognition mechanism to identify the already known concepts and uses a generative model to rehearse previously seen data to avoid catastrophic forgetting (Mundt et al., 2020b). In a concurrent work, the proposed open set recognition scheme is empirically val-

idated and the effect of model choice and deep neural network uncertainty approximations for recognition of unknowns is investigated (Mundt et al., 2019b). Lastly, the empirical usefulness of the developed approaches (Mundt et al., 2019b, 2020b) is demonstrated for the concrete defect detection use case proposed in Mundt et al. (2019a). This constitutes the only section of the thesis that has not yet been condensed into a separate publicly accessible manuscript and contains contents from technical reports (Mundt et al., 2018c,a) of the accompanying European Union's Horizon project "AEROBI" (AErial RObotics system for in-depth Bidge Inspection, grant No. 687384).
Chapter 3 culminates in a position on why neural network architecture choice, data selection, continual learning and open set recognition are interconnected and should not be viewed in isolation. It merges the threads pursued in the thesis and places them into the overarching context. This position is manifested in an extensive literature review of the current deep learning and earlier machine learning landscape. A generic framework is presented to highlight how the identification of learned data distribution boundaries, as gauged by the learned embedding, serves as a common anchor for continual learning rehearsal of known data, identification of suitable data for future inclusion through active queries, and recognition of previously unseen unknown data instances. A respective realization of this framework in deep generative neural networks as an extension of earlier proposed work (Mundt et al., 2019b, 2020a) is introduced and subsequently empirically validated.

In retrospect of the works presented in this thesis, many of the at the time selected questions and hypotheses are inherently intertwined. It is difficult to precisely ascribe them to a single element in the machine learning system development and accounting for the overall implications of the made choices. For example, this thesis' proposed method to identify unseen unknown data greatly benefits from specific DNN model choices, which can in turn involve distinct training procedures, that can then be further modified to solve the challenge of continuous updates. This further outlines the fundamental necessity of an overarching frame of reference that takes into account the entire system design process from beginning to end. The next section presents a detailed summary of the particular scientific hypothesis that have been investigated and the notable technical contributions that have been made.

Investigated Specific Hypotheses and Detailed Scientific Contributions

Following figure 2's train of thought and the respectively outlined structure, the superordinate chapters of the thesis have been titled: *Chapter 1: Designing Dynamic Deep Neural Net-*

work Architectures through Meta-Learning and Representational Capacity Expansion, *Chapter 2: Enabling Open Set Recognition and Continual Learning in Deep Neural Network Architectures* and *Chapter 3: Consolidating Viewpoints: Designing Neural Networks for Continual, Active Learning in an Open World*. A more elaborate scientific synopsis of the manuscripts that comprise each chapter, their investigated hypotheses and contributions are given in the ensuing text segments. In addition to this technically more detailed summary, the individual articles are then further preceded by general abstracts, both as a mean for the reader to recall the remaining thesis' structure and to reinforce the overarching perspective as presented in the synopsis.

Dynamic Deep Neural Network Architectures through Meta-Learning and Representational Capacity Expansion

Chapter 1 essentially covers the question of how to exploit neural network architecture designs and adapt them to be suitable for application to tasks for which they have not originally been designed or validated.

The first question in a sequence of examinations has been to what extent typical neural networks are parametrized appropriately, assuming that the designed order and type of operations in the hierarchy, i.e. the neural network layers, are desirable. A corresponding manuscript *Building effective deep neural network architectures one feature at a time* (Mundt et al., 2017) (technical report arXiv:1705.06778) has thus been motivated from a typically occurring mismatch between a neural network's specified representational capacity, that is the maximum amount of features a designer dedicates to each layer in the hierarchy, and the actual effective representational capacity that is put to use in the training process (Goodfellow et al., 2016). Predominantly, this imbalance is resolved post-hoc by employing an over-parametrized model that gets heavily regularized (Srivastava et al., 2014; Ioffe and Szegedy, 2015) to avoid overfitting and later pruning and compressing redundant or obsolete parameters (Hinton et al., 2014; Han et al., 2015, 2016; Kang et al., 2016; Shrikumar et al., 2016; Alvarez and Salzmann, 2016; Hao et al., 2017; Rodriguez et al., 2017; Han et al., 2017). This is not only problematic because it is inefficient, but also because it inherently lacks mechanisms to dynamically adapt architectures that go beyond repeatedly training abundant amounts of parameters and then removing the unused parts.
In the suggested work (Mundt et al., 2017), inspiration is thus taken from the inverse perspective of neurogenesis (Ash, 1989; Gross, 2000; Vadodaria and Jessberger, 2014), the bottom-up alternative that suggests to grow the amount of parameters as required during learning. For this, the core assumption is that there exists an inherent regularization effect

of stochastic gradient descent that leads to unused features not adapting from their state at initialization. As such, the key hypotheses that is then formed is that a structural change in a feature with respect to its state at initialization is an indicator of feature importance. A non-changing feature is hypothesized to be either due to: the initialization being a perfect solution that does not require any prospective alteration to solve the task, or alternatively a variety of complexly interacting effects such as too high representational capacity, the nature of the cost function, explicitly imposed or implicit regularization or the type of optimization algorithm resulting in a feature not being updated and thus effectively being obsolete. As the former is highly unlikely in very high dimensions, the work has proposed a novel algorithm to add extra features to a dynamically growing neural network as long as structural change is identified among all features of a layer. This algorithm relies on an introduced measure that captures a feature's self-resemblance over time. It makes use of a normalized cross-correlation between feature weight tensors at their initial time step of initialization and the eventually changed tensor at any other point in time of training. With the aid of L_2-norms, the normalized cross-correlation metric is constructed such that the self-resemblance is invariant to translation or rescaling. This effectively discounts changes observed in the weight tensors that are a sole result of e.g. subtraction of fixed regularization terms when gradients have vanished. Based on deep neural networks introduced in previous literature (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016; Zagoruyko and Komodakis, 2016), multiple architecture skeleton designs are taken and the proposed expansion algorithm experimentally validated on popular image classification datasets by progressively growing the representational capacity from a bare minimum of one feature at initialization (LeCun et al., 1998; Krizhevsky, 2009; Russakovsky et al., 2015). The evolved architectures are found to empirically outperform the literature baselines on these datasets either in terms of rivalling accuracy with significantly less parameters, or with respect to improved accuracy at the expense of additional parameter allocation. Although quite intuitive, this further highlights the necessity of flexible architectures that scale with task complexity instead of static task-agnostic neural network designs.

One emerging observation from the experiments on adaptively growing neural network representational capacity is that the distribution of feature amounts across layers does not seem to resemble the typical design patterns of the literature. Whereas evolved architectures systematically allocate the majority of their representational capacity to early operations in the hierarchy, the previously prevalent design increases the number of features with increasing depth of the neural network architecture, often simply by doubling the amount multiple times (Simonyan et al., 2014; He et al., 2016; Zagoruyko and Komodakis, 2016). LeCun et al. (2015) summarizes the rationale for the latter convolutional neural network composition as drawing inspiration from simple and primitive cells in the brain (Hubel and Wiesel,

1962) and being reminiscent of the the visual cortex' structure (Felleman and Van Essen, 1991; Cadieu et al., 2014). As such, it could be argued that lower layers require little capacity because they are hypothesized to encode generically usable features such as edge or color filters, and in turn the deeper layers in the hierarchy necessitating a rich encoding complexity to derive concept specific complex abstractions. In a consecutive work entitled *Rethinking Layer-wise Feature Amounts in Convolutional Neural Network Architectures* (Mundt et al., 2018b) (NeurIPS critiquing and correcting trends in machine learning workshop) the previously predominant assumption on capacity allocation and respectively hypothesized rule of thumb of deep neural network feature distribution is thus challenged. This is done by defining a simple three-parameter univariate skew normal distribution to parametrize the family of neural networks with respect to their distribution of feature amounts across layers. The result of the empirical characterization across popular image classification benchmark datasets (LeCun et al., 1998; Krizhevsky, 2009; Clanuwat et al., 2018) is that an almost inverse pattern to the prevalent design assumption is found to be consistently preferable. Hypothetically this could be due to previous underspecification of low-level primitive features or over-parametrization of high-level abstractions leading to high degrees of memorization. Ultimately, whether or not the actual features in the hierarchy truly correspond to primitive and abstract features, the work points out the necessity for more thorough future analysis of proposed architectures. Rather than assuming that a designed neural network is suitable across many tasks on the basis of inspiration and intuition from previous experiments, a deeper analysis on the network and task specific information flow is required.

Although the previously described capacity expansion technique and the analysed network feature topologies show empirical potential, there are two central limitations without further modifications. First, it is not immediately apparent how to extend the architecture expansion to also determine the amount of required layers, or at an even more profound level, how to select the mathematical operation itself. Second, there are currently no guarantees for the expansion procedure's success, as a respective grounding in theory remains open. A concurrently emergent trend of meta-learning, that is learning to learn, comes with the promise of solving these challenges. In the context of neural network architectures, meta-learning can be associated with neural architecture search (Baker et al., 2016; Real et al., 2017; Zoph and Le, 2017), where a learning algorithm is used to discover suitable neural network designs that subsequently learn the task itself. This is often framed as a reinforcement learning problem, where the validation accuracy of a trained architecture corresponds to the reward to be maximized and the combination of individual layer operations constitutes the search space (Baker et al., 2016; Zoph and Le, 2017). Whereas the earlier capacity expansion technique has initially found use beyond the standard computer vision benchmarks in practical application to concrete defect detection (Mundt et al., 2018d), the

ability of meta-learning to derive entire neural architectures has thus been favoured in practice later. A corresponding dataset for this concrete defect detection and classification task, together with an investigation of suitable neural architectures has been published in the work titled *Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset* (Mundt et al., 2019a) (CVPR). In essence, the work's contributions are three-fold. The first contribution is the established high-resolution dataset, which is not only significant because it adds to the corpus of available texture (Dana et al., 1999; Sharan et al., 2009; Hayman et al., 2004; Bell et al., 2015) and object (Everingham et al., 2014; Russakovsky et al., 2015) benchmarks, but also because it expands upon previously limited domain-specific data advances (Shi et al., 2016; Yang et al., 2017). Based on previous concrete defect detection datasets (Shi et al., 2016; Yang et al., 2017), prior machine learning applications are typically constrained to crack versus non-crack classification (Li et al., 2018; da Silva and de Lucena, 2018; Kim et al., 2018). In contrast, the introduced COncrete DEfect BRidge IMage (CODEBRIM) dataset features high-resolution images that display multiple defect categories with frequent overlaps. The resulting application is thus a substantially more challenging multi-class and multi-target task. From a civil engineering perspective, this is imperative because the taxonomy of defects and their severity, judged e.g. by the co-occurrence of multiple defects and the interplay of particular defect classes, plays a significant role with respect to a structure's integrity (Koch et al., 2015). Second, on the basis of the CODEBRIM dataset, an investigation of best-practice static neural network architectures (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Andrearczyk and Whelan, 2016; Zagoruyko and Komodakis, 2016; Huang et al., 2017) together with a thorough examination of hyper-parameter selection, i.e. the correlation between the trained model's accuracy and image size, stochastic gradient descent mini-batch size, learning rate, is conducted. This includes an examination of transfer learning from commonly exploited datasets such as the object centered ImageNet (Russakovsky et al., 2015) or texture focused "materials in natural context" (MINC) (Bell et al., 2015) database. Due to their astonishingly large amount of images in the millions, such pre-training is frequently speculated to be advantageous for applications where scarcer data amounts are available. Perhaps not surprisingly, recall the introduction's statements on the closed-world assumption, this form of feature transfer is empirically observed to be futile. Analogously, claimed performance advantages of one static literature neural network baseline over the other are not mirrored in the CODEBRIM scenario. Consequently, the efficacy of two meta-learning architecture search procedures (Baker et al., 2016; Pham et al., 2018) is studied for CODEBRIM. For this purpose, two variants, Meta-QNN (Baker et al., 2016) and ENAS (efficient neural architecture search) (Pham et al., 2018), are adapted. The former is based on tabular Q-learning (Watkins and Dayan, 1992), the latter is based

on policy gradients (Sutton et al., 1999) and an additional long short-term memory (LSTM) neural network (Hochreiter and Schmidhuber, 1997) that learns the policy. The search space is extended to involve spatial pyramidal pooling (He et al., 2014), both to find scale quasi-invariant neural architectures, as well as enabling processing of varying input sizes as a result of an adaptive mapping from flexible input dimensionality to fixed size output vectors. Most of the best-practice literature baseline architectures, even those claimed to be specifically designed towards texture based problems (Andrearczyk and Whelan, 2016), cannot compare with the meta-learned architectures. For the specific application, the latter contain significantly less parameters, generally fewer layers, while rivalling or outperforming human designed counterparts in terms of accuracy. The design of the common search space across the Meta-QNN and ENAS methods further allows for a direct comparison of the two methods. For the specified task both methods yield comparable architectures. This is intriguing in foresight of future application to deep continual learning as the ENAS procedure already continuously shares partially learned features throughout the course of its architecture search, whereas Meta-QNN treats each suggested architecture as a blank slate.

Just like in the other two manuscripts that are accumulated in this thesis chapter, the last work again highlights that the machine learning workflow requires an all-encompassing view that transcends the ubiquitous narrative of conveniently using a deep neural network as an advisable out-of-the-box solution for any task. The successive chapters further substantiate this perspective.

Enabling Open Set Recognition and Continual Learning in Deep Neural Network Architectures

Chapter 2 covers a related question of how to exploit neural network architectures in practice. Rather than pursuing the former chapter's question of how to adequately construct a neural architecture or adapt its capacity to obtain the desired accuracy on a new task, this chapter addresses the suitability of DNNs from a different angle. It is the perspective of enabling deep neural networks to overcome their closed set training and enable a continuous learning process. In other words, how can we protect DNNs from the persisting threat of overly confident false predictions on unseen unknown data instances that are distinct from the observed training distribution (Matan et al., 1990; Scheirer et al., 2013; Bendale and Boult, 2016; Nalisnick et al., 2019; Ovadia et al., 2019) and alleviate the catastrophic forgetting hazard when such data is consecutively trained (McCloskey and Cohen, 1989; Ratcliff, 1990)? As both of these phenomena are deeply rooted in the workhorse of modern deep learning, i.e. backpropagation based stochastic gradient descent in conjunction

with densely entangled hierarchical representations (French, 1992), these aspects are thus inherently tied to the design of DNNs. In essence, this forms a double-edged sword with the dilemma of whether to abandon the prevalent deep learning representations in favor of traditional computer vision models at the potential cost of expressivity, think of deep learning's capacity to capture the complex real-world interaction of multiple overlapping defect classes of the earlier introduced CODEBRIM dataset, or being confronted with the above limitations. Whereas both paths are certainly worthy of pursuit, this thesis naturally focuses on overcoming the latter. However, in contrast to most of the to this day prevalent literature, see Boult et al. (2019) for a review on deep open set recognition and Parisi et al. (2019) for a review on deep continual learning, the works presented in this chapter do not attempt to address the DNN open set and continual learning problems individually. This stems from an early realization that identification of the boundary between the known data population and samples from unknown distributions naturally grants the means to precisely protect acquired information. Hence it seemed natural to attempt to find a solution that merges open set recognition and continual learning from the start, especially since the prevention of overconfident false predictions would emerge as a limitation for robust application in continually trained neural networks anyway. Instead of following recent trends to find situational techniques that can demonstrate alleviated catastrophic forgetting in terms of accuracy on specific benchmarks, see Kemker et al. (2018); Farquhar and Gal (2018); Parisi et al. (2019); Lesort et al. (2019); Pfülb and Gepperth (2019); De Lange et al. (2019) for overviews of the catastrophic forgetting centric perspective, the goal has therefore directly been broadened to encapsulate learning and application beyond the closed world.

The backbone of this chapter is formed by two interconnected works. The first work, *Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition* (Mundt et al., 2020b) (under review, preprint arXiv:1905.12019), introduces a natural mechanism for open set recognition in deep neural networks, that is shared across both works, and proposes its principled role in mitigating catastrophic forgetting in continual learning. The second work, *Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers?* (Mundt et al., 2019b) (ICCV, first workshop on statistical deep learning for computer vision), compares and contrasts the empirical efficacy of the suggested open set recognition mechanism in relation to model choices and imaginable alternative techniques for recognition of unseen unknown data, such as commonly applied heuristics or approximations of uncertainty in deep neural networks. Although this latter work has eventually wound up being published first, it is chronologically anteceded.

Correspondingly, *Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition* (Mundt et al., 2020b) introduces a single deep model for continual learning that is naturally capable of open set recognition. The work draws its inspiration from two streams of prior work, one on extreme value theory (EVT) based meta-recognition to identify unseen unknown data (Scheirer et al., 2013, 2014), and the other on diminishing catastrophic interference in DNNs by rehearsing already learned concepts with the help of a deep generative model (Robins, 1995; Shin et al., 2017). Here, extreme value theory based meta-recognition for open set recognition is formally defined as limiting the risk of examples that are outside of a union of balls of a particular radius that include all training examples in some feature space (Scheirer et al., 2013, 2014). Accordingly, a distributional fit with respect to the observed extreme distances of the occupied feature space can be employed to formalize an outlier likelihood. This outlier likelihood is then large for individual instances for which the distance in this feature space does not meet the probabilistic expectation. This idea has been transferred to use with deep neural network classifiers by Bendale and Boult (2016). They have formulated the OpenMax algorithm, which fits a per-class Weibull distribution on the basis of distances in a deep neural network's penultimate layer's feature space and subsequently lowers the output confidence for novel instances whose distance surpasses the acquired distribution's heavy tail. Although shown to be empirically preferable to rather simple heuristics, the approach in this form nevertheless comes with a major limitation: there is little control over the type of information encoded in the penultimate layer of a deep neural network classifier. Most importantly, crucial information that describes the full data distribution is purposefully discarded if it doesn't aid in minimizing the formulated classification loss surrogate. This is in addition to the general open questions concerning the encoded patterns' nature, i.e. the encoded patterns could simply resemble noise (Ilyas et al., 2019) and thus not be beneficial to distinguish the desired human taxonomy of concepts. In the spirit of these former works, the key contribution of this chapter's first work is an EVT based meta-recognition variant that is rooted in the actual data distribution and its generative factors. For this purpose, a deep generative neural network model that captures the joint data and label distribution in a latent space is constructed and the EVT meta-recognition procedure respectively adapted to rely on the generative factors of variation. This is enabled through auto-encoding of variational Bayes (Kingma and Welling, 2013), which seeks to match the data's approximate posterior with a specified prior distribution through a probabilistic neural network encoder, and simultaneously trains a probabilistic decoder to produce a distribution over possible data values based on the encoded generative factors. To ensure that the generative factors are clustered according to a supervised signal, a linear separability objective for label attribution is added to the latent space. Besides now attempting to encapsulate all representations that characterize the data

distribution and thus the open set recognition yielding empirically impressive results, the identification of the known generative factors and regions of the latent space has an additional inherent advantage. It clarifies the essential question of how to robustly sample from the prior in semi-supervised variational auto-encoders (VAE) (Kingma et al., 2014) in order to generate instances that correspond to the original data and hence use them for rehearsal in continual learning. This is because identification of known high-density latent space regions conveniently circumvents the dispute around the unavoidable mismatch between the specified prior and the actually optimized approximate posterior distribution. In reality, the latter is never desired to actually fully resemble the prior, at least if the specified prior is very simplistic such as a unit Gaussian distribution (Tomczak and Welling, 2018; Bauer and Mnih, 2019), as this would collapse any innately present and captured structure in the data (Hoffman and Johnson, 2016; Burgess et al., 2017; Mathieu et al., 2018, 2019). In fact, the proposed open set recognition procedure identifies the boundaries of this so called aggregate posterior and allows to sample and generate data that veritably resembles actually observed instances, without excessive interpolation or ambiguity. For continual learning, the generated data can then serve to protect previously acquired information from being catastrophically forgotten. Not only is the requirement of training separate neural network classifiers and generative models thus lifted, in contrast to e.g. Shin et al. (2017) who always train additional generative adversarial networks (GAN) (Goodfellow et al., 2014), but resulting continual learning accuracy is substantially improved on several visual (LeCun et al., 1998; Krizhevsky, 2009; Xiao et al., 2017) and audio (Becker et al., 2018) benchmarks. The capacity for deep continual learning is thus provided, all while being able to identify unseen unknown data and prevent a subsequent misprediction in robust application. The proposed unified framework is ultimately shown to scale to high-resolution color images (Nilsback and Zisserman, 2006), by adopting further literature advances on auto regressively modelling the dependency between pixels in an image (van den Oord et al., 2016; Larsen et al., 2016; Chen et al., 2017; Gulrajani et al., 2017) and adversarial VAE training mechanisms (Ulyanov et al., 2018; Huang et al., 2018).

The second work in this chapter continues this thread with further quantification and empirical assessment of open set recognition. In *Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers?* (Mundt et al., 2019b) it is investigated to what extent a generative classifier, and thus an explicit approximation of the data distribution, is required for successful open set recognition. Consequently three model choices are contrasted: a conventional deep neural network blackbox classifier, a variational discriminative classifier and a deep variational Bayesian generative model. For each model choice two further aspects are explored, the contrast between the outlier detection criterion hinging on a predictive heuristic such as predictive entropy and

the multiple realizations of the EVT based meta-recognition method. In dependence on the exact model choice, the latter corresponds to the proposed open set recognition variants of Bendale and Boult (2015) and Mundt et al. (2020b). For each combination of model variant and metric, a further examination with respect to uncertainty approximations in deep neural networks is conducted with respect to their role in recognition of unseen unknown data. The corresponding implementation to gauge uncertainty is based on the arguments of Gal and Ghahramani (2015), who suggest that employing a Dropout operation (Srivastava et al., 2014) at each layer and conducting stochastic forward passes at the prediction stage can be viewed as a variational approximation. The respectively termed Monte-Carlo Dropout is argued to treat the model weights as a random variable that is marginalized. Given that individual weights are set to either zero or one with a certain probability in Dropout, the assumed distribution is then a Bernoulli distribution. The obtained results across multiple datasets (LeCun et al., 1998; Krizhevsky, 2009; Netzer et al., 2011; Xiao et al., 2017; Clanuwat et al., 2018; Becker et al., 2018) indicate that a pure discriminative model is insufficient to address the open set challenge, independently of whether uncertainty is accounted for or not. The deep neural network uncertainty is further empirically observed to be inadequate for recognition of unknown data in combination with prediction values. This may be due to the lacking approximation quality of Monte-Carlo Dropout or a deeper limitation of unseen unknown data not being expressible by Bayes rule, as argued extensively in Boult et al. (2019). In conjunction with the previously summarized manuscript, this work further highlights the requirement to consider the complete data distribution beyond features that only aid in a classification objective, even if the latter initially seems to appear as the exclusive goal.

The chapter is concluded with the only section in this thesis that is not comprised of a separate publicly available manuscript. It does however contain non-verbatim elements of the results presented in technical reports *AEROBI - D3.3 Deliverable: Cognitive Vision System V2* (Mundt et al., 2018c) and *AEROBI - D3.6 Deliverable: Online Learning* (Mundt et al., 2018a) of the European Union's Horizon 2020 "AEROBI" project under grant agreement No. 687384. The section demonstrates the advantages of the historically later developed techniques of this chapter on the earlier introduced CODEBRIM dataset. Hence, qualitative illustrations are shown for semantic segmentation with simultaneous application of the proposed open set recognition techniques to practical concrete defect detection.

Consolidating Viewpoints: Designing Neural Networks for Continual, Active Learning in an Open World

The third and last manuscript based chapter of the thesis provides the final overarching perspective and widens the established viewpoint even further. In its essence, the respective work *A Wholistic View of Continual Learning with Deep Neural Networks* (Mundt et al., 2020a) (under review, preprint arXiv:2009.01797) puts forward an extensive position and arguments for the necessity of such a consolidated view. It highlights the connections between neural architecture choices, continual learning and open set recognition. It extends the framework of Mundt et al. (2020b) to additionally encompass the challenges of active data and task selection in continuous learning processes, i.e. ordering and choosing data instances for consecutive training steps to maximize the expected performance gain.

In order to adequately portray the explored synergies, the work first starts by recalling literature definitions of continual learning and related paradigms, summarizing recent static and continual benchmark evaluation practices and critiquing their limited value due to a mismatch with the practical desiderata of transferable and robust real-world application. To embed this into a grander frame of reference, a large scale review of the typically isolated advances in deep continual learning, active learning and learning in an open world is contributed. For each of these fields in the literature, individual techniques are grouped into a taxonomy of methods. This taxonomy serves the essential purpose of highlighting shortcomings. The latter are pointed out to have been identified rather early in more mature machine learning literature, but seem to be frequently overlooked or forgotten in a resurgence of respective methodology in deep neural network learning. Consequently, these are grouped into five insights from past literature, that have rather provocatively been termed forgotten lessons. The precise content of these forgotten lessons revolves around: 1. the repeatedly disregarded closed world assumption and the fact that discriminative neural networks are almost guaranteed to falsely and overconfidently predict outside of their closed world; 2. the misleading premise that neural network uncertainty heuristics serve as a principled mean to identify the open set and query novel data without being susceptible to meaningless or uninformative outliers; 3. the somewhat naive assumption that explicit designation of what should constitute novel data solves the challenge through calibration methods; 4. the generally neglected importance and relevance of data and task order in continual learning and learning curricula in general; 5. the inherent role of architecture and parameter growth in the move from the small data regime to ever increasing amounts of information in contrast to the generally assumed static neural network nature. In the manuscript, each of these findings is corroborated in detail and tied to the preceding literature review.

The second half of the article then focuses on the natural interface and interconnections between these statements and their respective open challenges. The mechanism to identify the learned distribution boundaries is identified as the common denominator among the previously seemingly distinct questions. In a deep neural network implementation this can equate to the earlier proposed approximate posterior based boundaries in the latent space of a deep variational generative neural network as practically gauged through the use of extreme value theory (Mundt et al., 2019b, 2020b). Based on this single unified standpoint, several algorithmic variations to address specific goals are formulated. On the one hand, recognition of the open set can be adopted in analogy to the previously presented works to protect the neural network from generating ambiguous examples for continual learning and discard nonsensical overconfident predictions. On the other hand, knowledge of the boundary between known and unknown can be further extended to use cases in two additional directions. Rather than just rejecting nonsensical overconfident predictions, the measure of outlier likelihood can be used to rank-order novel data instances. They can correspondingly be grouped into instances that are expected to yield little future improvement, i.e. redundant samples with high likelihood of being inliers of the seen distribution, instances that are expected to provide additional task-relevant information, i.e. samples that show some relation and moderate statistical deviation with respect to already seen data, and instances that are prone to yielding little task improvement, i.e. new inputs that are complete statistical outliers due to consisting of uninformative noise or featuring other extraneous content. This can be used for robust active learning or the construction of an intuitive curriculum of tasks and data, where difficulty is measured according to overlap with the already observed data distribution. On the flip side of the same coin, already seen data instances can be picked according to their distance distribution to the known latent space boundary. As a proxy to sampling from the actual data distribution, the posterior approximation can thus be used to retain subsets of the original data that more accurately reflect the real data distribution than a simple uniform sub-sampling would. This can be used for robust continual learning, as a drop-in rehearsal replacement or in addition to previously proposed generative replay. Finally, all these algorithmic variants with a common core are empirically evaluated, both from the isolated perspective in comparison with active and continual learning methods proposed in the literature on closed world fixed-order benchmarks, as well as in situations where the closed world assumption is partially lifted through inclusion of corrupted and perturbed data instances and where task order can be selected freely. The proposed unified perspective and its realization in deep neural networks is not only shown to outperform other methods in their original environment, it is also shown to have a significant edge over these methods when unexpected data can be encountered.

The article ends with a proposition on a refined definition of continual deep learning that includes the innate challenges of data selection, architecture modification and learning robustly in an open world. One conceivable suggestion for a more comprehensive and system oriented design and evaluation strategy is presented in the outlook. In its core, this suggestion coincides with the voiced concerns and requirement of considering the entire machine learning workflow, from data to architecture specifics and training to evaluation, as presented in the synopsis of this thesis.

General Article Abstracts: Reinforcing the Synopsis

The foregoing thesis synopsis has given a detailed view into the individual technical topics of the upcoming thesis' articles and their larger machine learning context. In order to facilitate navigation for the reader, the previously detailed specific scientific contributions of each work are summarized once more from a general perspective. These abstracts are intended to reinforce the investigated topics and recall the overarching view.

Chapter 1: Designing Dynamic Deep Neural Network Architectures through Meta-Learning and Representational Capacity Expansion

- ***Building effective deep neural network architectures one feature at a time:*** Neural networks are often used to automatically learn some abstract representations to solve a task given data. In practical machine learning, an engineer typically uses a pre-made architecture for this purpose. Such a neural architecture is composed of multiple building blocks called layers, where each individual building block comes with a choice of width. In simplified terms and for a fixed amount of layers, this width mirrors the learnable complexity. If too little, we are faced with the threat of not being able to learn the necessary combination of representations required to solve a task. If too large, the training is slow and the additional danger is a pure memorization of every data point with little practical value. The conventional literature approach is thus to train a very large neural network and consecutively discard representations that do not contribute much to the task at hand. In the proposed work a reverse approach is shown to be empirically successful, where a very small neural network is used at the start and the width is increased adaptively during training to increase the neural network's complexity. To the extent that the experimental validation holds, this addresses the high-level machine learning workflow question of how to select a suitable neural network for a new task, as the proposed method will aid the engineer in automatically determining the required layer widths.

- ***Rethinking Layer-wise Feature Amounts in Convolutional Neural Network Architectures:***
  In practical use of deep neural networks a rule of thumb seems to have emerged as a result of successful experiments. This rule of thumb serves as a guideline to the machine learning engineer with respect to the design of convolutional neural network architectures. In the latter, multiple mathematical convolution operations are executed in sequence, and the engineer has to decide both on the number of sequential operations, as well as the number of operations that are computed in parallel at the same level of the hierarchy. The respectively evolved unwritten rule is to increase the number of parallel computations as the pipeline's depth increases, inspired by the hypothesis that the neural network learns increasingly task specific features at deeper levels. Conversely, as the first operations in the sequence are speculated to learn very basic features such as edges or colors in computer vision, it is hypothesized that a lesser amount is required, as they can be shared across various imaginable tasks. For a selection of popular computer vision image classification datasets, the article questions this hypothesizes and analyzes this rule of thumb by modifying the maximum number of learnable features at various levels of the neural network hierarchy. In the experiments, a reverse phenomenon to the above described design pattern can be observed. Regarding the overarching machine learning workflow, this thus raises concerns with respect to the practice of using the same neural network design across various tasks.

- ***Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset:***
  A dataset for detection and classification of concrete defects in civil bridge infrastructure is introduced. Whereas previous datasets concentrate on cracks as the key threat, the proposed work provides an extension by also including spallation, calcium leeching, exposed reinforcement bar and corrosion stain defects. These can all occur simultaneously within a captured image. The resulting task of classifying these co-occurring defects into their categories is investigated from a perspective of deep neural networks. For this introduced application, it is observed that neural network architecture designs of previous literature do not provide similar benefits as discovered in original experiments on popular image classification benchmarks. In contrast, meta-learned neural architectures, that is neural networks whose building block composition is learned with the aid of an additional learning mechanism that optimizes the neural network's structure for the particular task, are found to be favorable in terms of final task performance and smaller neural network size. In context of the machine

learning workflow, this reinforces the point that neural network design needs to take into account the particular task at hand and it should not be assumed that a single static design suffices across various applications.

Chapter 2: Enabling Open Set Recognition and Continual Learning in Deep Neural Network Architectures

- ***Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition:***
  A well-known challenge of deep neural networks is their current inability to learn continuously and to distinguish unknown input from what has been presented to the network during training. Without additional mechanisms, a neural network will attribute a new unknown example to any of the old concepts with high confidence. Trying to solve this by updating the network exclusively on new examples will simply overwrite these older concepts. In the work it is argued that these challenges are intertwined and can be commonly addressed with a single proposed mechanism in deep generative neural networks. Such a generative neural network attempts to explicitly encode the data distribution of the observed data population. It first decomposes inputs into individual factors, that are then taken together to recompose and generate data. In the article it is shown how measuring distances to these generative factors that describe the already seen data can serve as a solution towards both continuous learning and recognition of unknown examples. To avoid forgetting of previously seen data, generation can be used to rehearse data that lies close to past inputs and thus reinforce older concepts. At the same time, a false prediction for entirely unknown inputs can be prevented when a large dissimilarity to the already seen data distribution is noticed. The work highlights the interplay of different aspects of the machine learning workflow, such as continuous learning and robust predictions being interdependent, and provides first steps towards a common solution.

- ***Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers?:***
  This article focuses on the effect of neural network choice and selection of metric to determine whether a newly seen data point belongs to known concepts or is presently unknown. The work conducts experiments that investigate how simple threshold heuristics on a neural network's classification output perform in contrast to measuring distances in its learned feature space. This investigation is augmented with an optional assessment of uncertainty, that is the intensity of fluctuations in the output

when computation is repeated with random deactivation of some of the learned features. The experimental analysis is coupled to the choice of neural network type in terms of the underlying statistical modelling. In essence, discriminative models are compared to their generative counterparts. In the context of deep neural networks, the generally used former variant simply predicts a class given a data sample, whereas the latter also explicitly considers the question of how the data was generated. As a result of the experiments, the proposed approach that takes into account the data formation process and measures data similarity according to distance to the seen data distribution outperforms simple classifiers in detection of unknown examples. From a perspective of the overall machine learning workflow, this reinforces how robust application is inherently tied to the choice of machine learning model.

Chapter 3: Consolidating Viewpoints: Designing Neural Networks for Continual, Active Learning in an Open World.

- ***A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning:***
  The article presents a comprehensive perspective of how various aspects of the neural network based machine learning workflow need to be treated together. It presents the position that typically separately treated elements are synergistic, based on a broader review of the literature and insights from older works. The central theme is a bridge between the challenges faced when neural networks need to learn continuously, when asked to separate known learned concepts from arbitrary data input for robust prediction, and the essential question of what data to include for training. A common framework is suggested based on previously developed generative neural networks. Because these models allow to approximate the distribution of the seen data population, a single key mechanism is used to derive algorithmic variants to overcome the previously formulated three-fold challenge. The results show that the introduced framework improves upon previous techniques that are tailored towards only one of the challenges, while providing a common frame of reference. This is demonstrated in experiments to increase accuracy in continual learning image classification, improve accuracy when selecting limited amounts of data for training, select which task to learn next and how to diminish performance degradation when data is corrupted. The article thus builds upon the insights from various former works and ties them together, advocates the necessity of a systems perspective and demonstrates its utility in experiments from a deep neural network perspective.

Chapter 1

Designing Dynamic Deep Neural Network Architectures through Meta-Learning and Representational Capacity Expansion

BUILDING EFFECTIVE DEEP NEURAL NETWORK ARCHITECTURES ONE FEATURE AT A TIME

Martin Mundt
Frankfurt Institute for Advanced Studies
Ruth-Moufang-Str. 1
60438 Frankfurt, Germany
mundt@fias.uni-frankfurt.de

Tobias Weis
Goethe-University Frankfurt
Theodor-W.-Adorno-Platz 1
60323 Frankfurt, Germany
weis@ccc.cs.uni-frankfurt.de

Kishore Konda
INSOFE
Janardana Hills, Gachibowli
500032 Hyderabad, India
kishore.konda@insofe.edu.in

Visvanathan Ramesh
Frankfurt Institute for Advanced Studies
Ruth-Moufang-Str. 1
60438 Frankfurt, Germany
ramesh@fias.uni-frankfurt.de

ABSTRACT

Successful training of convolutional neural networks is often associated with sufficiently deep architectures composed of high amounts of features. These networks typically rely on a variety of regularization and pruning techniques to converge to less redundant states. We introduce a novel bottom-up approach to expand representations in fixed-depth architectures. These architectures start from just a single feature per layer and greedily increase width of individual layers to attain effective representational capacities needed for a specific task. While network growth can rely on a family of metrics, we propose a computationally efficient version based on feature time evolution and demonstrate its potency in determining feature importance and a networks' effective capacity. We demonstrate how automatically expanded architectures converge to similar topologies that benefit from lesser amount of parameters or improved accuracy and exhibit systematic correspondence in representational complexity with the specified task. In contrast to conventional design patterns with a typical monotonic increase in the amount of features with increased depth, we observe that CNNs perform better when there is more learnable parameters in intermediate, with falloffs to earlier and later layers.

1 INTRODUCTION

Estimating and consequently adequately setting representational capacity in deep neural networks for any given task has been a long standing challenge. Fundamental understanding still seems to be insufficient to rapidly decide on suitable network sizes and architecture topologies. While widely adopted convolutional neural networks (CNNs) such as proposed by Krizhevsky et al. (2012); Simonyan & Zisserman (2015); He et al. (2016); Zagoruyko & Komodakis (2016) demonstrate high accuracies on a variety of problems, the memory footprint and computational complexity vary. An increasing amount of recent work is already providing valuable insights and proposing new methodology to address these points. For instance, the authors of Baker et al. (2016) propose a reinforcement learning based meta-learning approach to have an agent select potential CNN layers in a greedy, yet iterative fashion. Other suggested architecture selection algorithms draw their inspiration from evolutionary synthesis concepts (Shafiee et al., 2016; Real et al., 2017). Although the former methods are capable of evolving architectures that rival those crafted by human design, it is currently only achievable at the cost of navigating large search spaces and hence excessive computation and time. As a trade-off in present deep neural network design processes it thus seems plausible to consider layer types or depth of a network to be selected by an experienced engineer based on prior knowledge and former research. A variety of techniques therefore focus on improving already well established architectures. Procedures ranging from distillation of one network's knowledge into another (Hinton et al., 2014), compressing and encoding learned representations Han et al. (2016),

pruning alongside potential retraining of networks (Han et al., 2015; 2017; Shrikumar et al., 2016; Hao et al., 2017) and the employment of different regularization terms during training (He et al., 2015; Kang et al., 2016; Rodriguez et al., 2017; Alvarez & Salzmann, 2016), are just a fraction of recent efforts in pursuit of reducing representational complexity while attempting to retain accuracy. Underlying mechanisms rely on a multitude of criteria such as activation magnitudes (Shrikumar et al., 2016) and small weight values (Han et al., 2015) that are used as pruning metrics for either single neurons or complete feature maps, in addition to further combination with regularization and penalty terms.

Common to these approaches is the necessity of training networks with large parameter quantities for maximum representational capacity to full convergence and the lack of early identification of insufficient capacity. In contrast, this work proposes a bottom-up approach with the following contributions:

- We introduce a computationally efficient, intuitive metric to evaluate feature importance at any point of training a neural network. The measure is based on feature time evolution, specifically the normalized cross-correlation of each feature with its initialization state.
- We propose a bottom-up greedy algorithm to automatically expand fixed-depth networks that start with one feature per layer until adequate representational capacity is reached. We base addition of features on our newly introduced metric due to its computationally efficient nature, while in principle a family of similarly constructed metrics is imaginable.
- We revisit popular CNN architectures and compare them to automatically expanded networks. We show how our architectures systematically scale in terms of complexity of different datasets and either maintain their reference accuracy at reduced amount of parameters or achieve better results through increased network capacity.
- We provide insights on how evolved network topologies differ from their reference counterparts where conventional design commonly increases the amount of features monotonically with increasing network depth. We observe that expanded architectures exhibit increased feature counts at early to intermediate layers and then proceed to decrease in complexity.

2 Building neural networks bottom-up feature by feature

While the choice and size of deep neural network model indicate the *representational capacity* and thus determine which functions can be learned to improve training accuracy, training of neural networks is further complicated by the complex interplay of choice of optimization algorithm and model regularization. Together, these factors define define the *effective capacity*. This makes training of deep neural networks particularly challenging. One practical way of addressing this challenge is to boost model sizes at the cost of increased memory and computation times and then applying strong regularization to avoid over-fitting and minimize generalization error. However, this approach seems unnecessarily cumbersome and relies on the assumption that optimization difficulties are not encountered. We draw inspiration from this challenge and propose a bottom-up approach to increase capacity in neural networks along with a new metric to gauge the effective capacity in the training of (deep) neural networks with stochastic gradient descent (SGD) algorithms.

2.1 Normalized weight-tensor cross-correlation as a measure for neural network effective capacity

In SGD the objective function $J(\mathbf{\Theta})$ is commonly equipped with a penalty on the parameters $R(\mathbf{\Theta})$, yielding a regularized objective function:

$$\hat{J}(\mathbf{\Theta}) = J(\mathbf{\Theta}) + \alpha R(\mathbf{\Theta}) \ . \tag{1}$$

Here, α weights the contribution of the penalty. The regularization term $R(\mathbf{\Theta})$ is typically chosen as a L_2-norm, coined weight-decay, to decrease model capacity or a L_1-norm to enforce sparsity. Methods like dropout (Srivastava et al., 2014) and batch normalization (Ioffe & Szegedy, 2015) are typically employed as further implicit regularizers.
In principle, our rationale is inspired by earlier works of Hao et al. (2017) who measure a complete feature's importance by taking the L_1-norm of the corresponding weight tensor instead of operating

on individual weight values. In the same spirit we assign a single importance value to each feature based on its values. However we do not use the weight magnitude directly and instead base our metric on the following hypothesis: While a feature's absolute magnitude or relative change between two subsequent points in time might not be adequate measures for direct importance, the relative amount of change a feature experiences with respect to its original state provides an indicator for how many times and how much a feature is changed when presented with data. Intuitively we suggest that features that experience high structural changes must play a more vital role than any feature that is initialized and does not deviate from its original states' structure. There are two potential reasons why a feature that has randomly been initialized does not change in structure: The first being that its form is already initialized so well that it does not need to be altered and can serve either as is or after some scalar rescaling or shift in order to contribute. The second possibility is that too high representational capacity, the nature of the cost function, too large regularization or the type of optimization algorithm prohibit the feature from being learned, ultimately rendering it obsolete. As deep neural networks are commonly initialized from using a distribution over high-dimensional space the first possibility seems unlikely (Goodfellow et al., 2016).
As one way of measuring the effective capacity at a given state of learning, we propose to monitor the time evolution of the normalized cross-correlation for all weights with respect to their state at initialization. For a convolutional neural network composed of layers $l = 1, 2, \ldots, L-1$ and complementary weight-tensors $\mathbf{W}^l_{f^l j^l k^l f^{l+1}}$ with spatial dimensions $j^l \times k^l$ defining a mapping from an input feature-space $f^l = 1, 2, ...F^l$ onto the output feature space $f^{l+1} = 1, 2, ...F^{l+1}$ that serves as input to the next layer, we define the following metric:

$$\mathbf{c}^l_{f^{l+1},t} = 1 - \frac{\sum_{f^l, j^l, k^l} \left[\left(\mathbf{W}^l_{f^l j^l k^l f^{l+1}, t_0} - \bar{\mathbf{W}}^l_{f^{l+1}, t_0} \right) \circ \left(\mathbf{W}^l_{f^l j^l k^l f^{l+1}, t} - \bar{\mathbf{W}}^l_{f^{l+1}, t} \right) \right]}{\left\| \mathbf{W}^l_{f^l j^l k^l, t_0} \right\|_{2, f^{l+1}} \cdot \left\| \mathbf{W}^l_{f^l j^l k^l, t} \right\|_{2, f^{l+1}}} \tag{2}$$

which is a measure of self-resemblance. In this equation, $\mathbf{W}^l_{f^l j^l k^l f^{l+1}, t}$ is the state of a layer's weight-tensor at time t or the initial state after initialization t_0. $\bar{\mathbf{W}}^l_{f^{l+1}, t}$ is the mean taken over spatial and input feature dimensions. $\circ$ depicts the Hadamard product that we use in an extended fashion from matrices to tensors where each dimension is multiplied in an element-wise fashion analogously. Similarly the terms in the denominator are defined as the L_2-norm of the weight-tensor taken over said dimensions and thus resulting in a scalar value. Above equation can be defined in an analogous way for multi-layer perceptrons by omitting spatial dimensions.
The metric is easily interpretable as no structural changes of features lead to a value of zero and importance approaches unity the more a feature is deviating in structure. The usage of normalized cross-correlation with the L_2-norm in the denominator has the advantage of having an inherent invariance to effects such as translations or rescaling of weights stemming from various regularization contributions. Therefore the contribution of the sum-term in equation 1 does not change the value of the metric if the gradient term vanishes. This is in contrast to the measure proposed by Hao et al. (2017), as absolute weight magnitudes are affected by rescaling and make it more difficult to interpret the metric in an absolute way and find corresponding thresholds.

2.2 Bottom-up construction of neural network representational capacity

We propose a new method to converge to architectures that encapsulate necessary task complexity without the necessity of training huge networks in the first place. Starting with one feature in each layer, we expand our architecture as long as the effective capacity as estimated through our metric is not met and all features experience structural change. In contrast to methods such as Baker et al. (2016); Shafiee et al. (2016); Real et al. (2017) we do not consider flexible depth and treat the amount of layers in a network as a prior based on the belief of hierarchical composition of the underlying factors. Our method, shown in algorithm 1, can be summarized as follows:

1. For a given network arrangement in terms of function type, depth and a set of hyper-parameters: initialize each layer with one feature and proceed with (mini-batch) SGD.
2. After each update step evaluate equation 2 independently per layer and increase feature dimensionality by F_{exp} (one or higher if a complexity prior exists) if all currently present features in respective layer are differing from their initial state by more than a constant ϵ.
3. Re-initialize all parameters if architecture has expanded.

Algorithm 1 Greedy architecture feature expansion algorithm

Require: Set hyper-parameters: learning rate λ_0, mini-batch size, maximum epoch t_{end}, ...
Require: Set expansion parameters: $\epsilon = 10^{-6}$, $F_{exp} = 1$ (or higher)
1: Initialize parameters: $t = 1$, $F^l = 1 \forall\, l = 1, 2, \ldots L-1$, $\mathbf{\Theta}$, $reset = false$
2: **while** $t \leq t_{end}$ **do**
3: **for** mini-batches in training set **do**
4: $reset \leftarrow false$
5: Compute gradient and perform update step
6: **for** $l = 1$ to $L-1$ **do**
7: **for** $i = f^{l+1}$ to F^{l+1} **do**
8: Update $\mathbf{c}_{i,t}^l$ according to equation 2 } in parallel.
9: **end for**
10: **if** $\max(\mathbf{c}_t^l) < 1 - \epsilon$ **then**
11: $F^{l+1} \leftarrow F^{l+1} + F_{exp}$
12: $reset \leftarrow true$
13: **end if**
14: **end for**
15: **if** $reset == true$ **then**
16: Re-initialize parameters $\mathbf{\Theta}$, $t = 0$, $\lambda = \lambda_0$, ...
17: **end if**
18: **end for**
19: $t \leftarrow t + 1$
20: **end while**

The constant ϵ is a numerical stability parameter that we set to a small value such as 10^{-6}, but could in principle as well be used as a constraint. We have decided to include the re-initialization in step 3 (lines $15 - 17$) to avoid the pitfalls of falling into local minima[1]. Despite this sounding like a major detriment to our method, we show that networks nevertheless rapidly converge to a stable architectural solution that comes at less than perchance expected computational overhead and at the benefit of avoiding training of too large architectures. Naturally at least one form of explicit or implicit regularization has to be present in the learning process in order to prevent infinite expansion of the architecture. We would like to emphasize that we have chosen the metric defined in equation 2 as a basis for the decision of when to expand an architecture, but in principle a family of similarly constructed metrics is imaginable. We have chosen this particular metric because it does not directly depend on gradient or higher-order term calculation and only requires multiplication of weights with themselves. Thus, a major advantage is that computation of equation 2 can be parallelized completely and therefore executed at less cost than a regular forward pass through the network.

3 Revisiting popular architectures with architecture expansion

We revisit some of the most established architectures "GFCNN" (Goodfellow et al., 2013) "VGG-A & E" (Simonyan & Zisserman, 2015) and "Wide Residual Network: WRN" (Zagoruyko & Komodakis, 2016) (see appendix for architectural details) with batch normalization (Ioffe & Szegedy, 2015). We compare the number of learnable parameters and achieved accuracies with those obtained through expanded architectures that started from a single feature in each layer. For each architecture we include all-convolutional variants (Springenberg et al., 2015) that are similar to WRNs (minus the skip-connections), where all pooling layers are replaced by convolutions with larger stride. All fully-connected layers are furthermore replaced by a single convolution (affine, no activation function) that maps directly onto the space of classes. Even though the value of more complex type of sub-sampling functions has already empirically been demonstrated (Lee et al., 2015), the amount of features of the replaced layers has been constrained to match in dimensionality with the preceding convolution layer. We would thus like to further extend and analyze the role of layers involving sub-sampling by decoupling the dimensionality of these larger stride convolutional layers.

[1] We have empirically observed promising results even without re-initialization, but deeper analysis of stability (e.g. expansion speed vs. training rate), initialization of new features during training (according to chosen scheme or aligned with already learned representations?) is required.

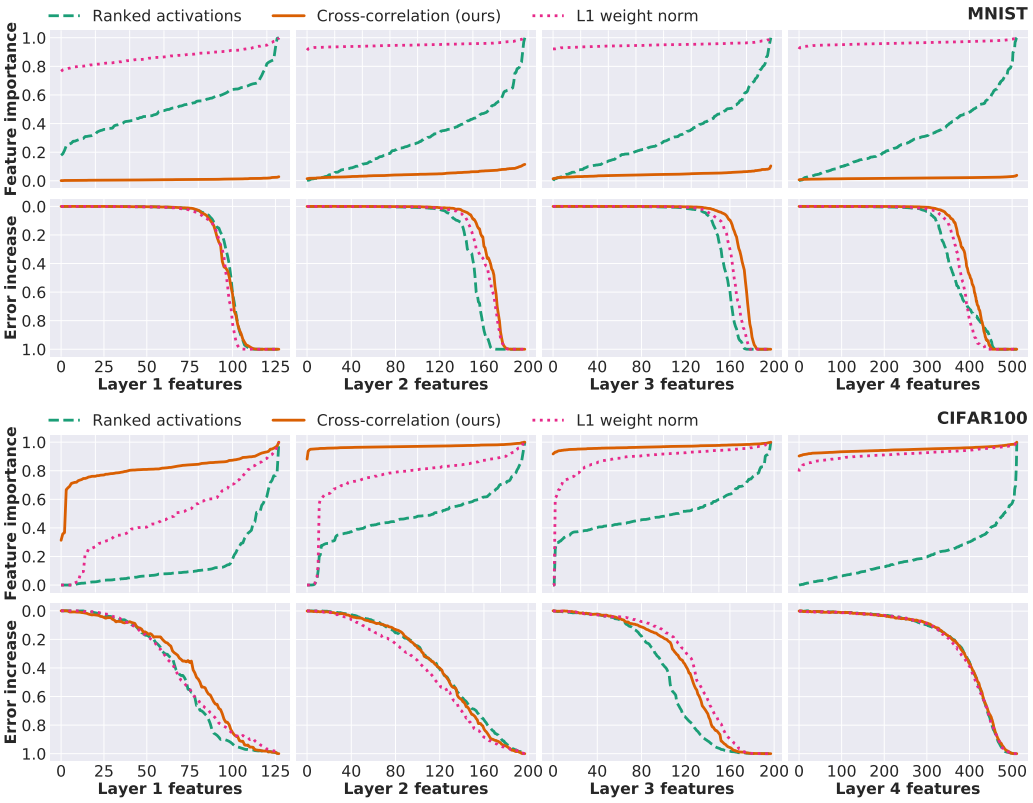


Figure 1: Pruning of complete features for the GFCNN architecture trained on MNIST (top panel) and CIFAR100 (bottom panel). Top row shows sorted feature importance values for every layer according to three different metrics at the end of training. Bottom row illustrates accuracy loss when removing feature by feature in ascending order of feature importance.

We consider these architectures as some of the best CNN architectures as each of them has been chosen and tuned carefully according to extensive amounts of hyper-parameter search. As we would like to demonstrate how representational capacity in our automatically constructed networks scales with increasing task difficulty, we perform experiments on the MNIST (LeCun et al., 1998), CIFAR10 & 100 (Krizhevsky, 2009) datasets that intuitively represent little to high classification challenge. We also show some preliminary experiments on the ImageNet (Russakovsky et al., 2015) dataset with "Alexnet" (Krizhevsky et al., 2012) to conceptually show that the algorithm is applicable to large scale challenges. All training is closely inspired by the procedure specified in Zagoruyko & Komodakis (2016) with the main difference of avoiding heavy preprocessing. We preprocess all data using only trainset mean and standard deviation (see appendix for exact training parameters). Although we are in principle able to achieve higher results with different sets of hyper-parameters and preprocessing methods, we limit ourselves to this training methodology to provide a comprehensive comparison and avoid masking of our contribution. We train all architectures five times on each dataset using a Intel i7-6800K CPU (data loading) and a single NVIDIA Titan-X GPU. Code has been written in both Torch7 (Collobert et al., 2011) and PyTorch (http://pytorch.org/) and will be made publicly available.

3.1 The top-down perspective: Feature importance for pruning

We first provide a brief example for the use of equation 2 through the lens of pruning to demonstrate that our metric adequately measures feature importance. We evaluate the contribution of the features by pruning the weight-tensor feature by feature in ascending order of feature importance values and re-evaluating the remaining architecture. We compare our normalized cross-correlation metric 2 to the L_1 weight norm metric introduced by Hao et al. (2017) and ranked mean activations evaluated

over an entire epoch. In figure 1 we show the pruning of a trained GFCNN, expecting that such a network will be too large for the easier MNIST and too small for the difficult CIFAR100 task. For all three metrics pruning any feature from the architecture trained on CIFAR100 immediately results in loss of accuracy, whereas the architecture trained on MNIST can be pruned to a smaller set of parameters by greedily dropping the next feature with the currently lowest feature importance value. We notice how all three metrics perform comparably. However, in contrast to the other two metrics, our normalized cross-correlation captures whether a feature is important on absolute scale. For MNIST the curve is very close to zero, whereas the metric is close to unity for all CIFAR100 features. Ultimately this is the reason our metric, in the way formulated in equation 2, is used for the algorithm presented in 1 as it doesn't require a difficult process to determine individual layer threshold values. Nevertheless it is imaginable that similar metrics based on other tied quantities (gradients, activations) can be formulated in analogous fashion.
As our main contribution lies in the bottom-up widening of architectures we do not go into more detailed analysis and comparison of pruning strategies. We also remark that in contrast to a bottom-up approach to finding suitable architectures, pruning seems less desirable. It requires convergent training of a huge architectures with lots of regularization before complexity can be introduced, pruning is not capable of adding complexity if representational capacity is lacking, pruning percentages are difficult to interpret and compare (i.e. pruning percentage is 0 if the architecture is adequate), a majority of parameters are pruned only in the last "fully-connected" layers (Han et al., 2015), and pruning strategies as suggested by Han et al. (2015; 2017); Shrikumar et al. (2016); Hao et al. (2017) tend to require many cross-validation with consecutive fine-tuning steps. We thus continue with the bottom-up perspective of expanding architectures from low to high representational capacity.

3.2 The bottom-up perspective: Expanding architectures

We use the described training procedure in conjunction with algorithm 1 to expand representational complexity by adding features to architectures that started with just one feature per layer with the following additional settings:

Architecture expansion settings and considerations: Our initial experiments added one feature at a time, but large speed-ups can be introduced by means of adding stacks of features. Initially, we avoided suppression of late re-initialization to analyze the possibility that rarely encountered worst-case behavior of restarting on an almost completely trained architecture provides any benefit. After some experimentation our final report used a stability parameter ending the network expansion if half of the training has been stable (no further change in architecture) and added $F_{exp} = 8$ and $F_{exp} = 16$ features per expansion step for MNIST and CIFAR10 & 100 experiments respectively.

We show an exemplary architecture expansion of the GFCNN architecture's layers for MNIST and CIFAR100 datasets in figure 2 and the evolution of the overall amount of parameters for five different experiments. We observe that layers expand independently at different points in time and more features are allocated for CIFAR100 than for MNIST. When comparing the five different runs we can identify that all architectures converge to a similar amount of network parameters, however at different points in time. A good example to see this behavior is the solid (green) curve in the MNIST example, where the architecture at first seems to converge to a state with lower amount of parameters and after some epochs of stability starts to expand (and re-initialize) again until it ultimately converges similarly to the other experiments.
We continue to report results obtained for the different datasets and architectures in table 1. The table illustrates the mean and standard deviation values for error, total amount of parameters and the mean overall time taken for five runs of algorithm 1 (deviation can be fairly large due to the behavior observed in 2). We make the following observations:

- Without any prior on layer widths, expanding architectures converge to states with at least similar accuracies to the reference at reduced amount of parameters, or better accuracies by allocating more representational capacity.
- For each architecture type there is a clear trend in network capacity that is increasing with dataset complexity from MNIST to CIFAR10 to CIFAR100 [2].

[2] For the WRN CIFAR100 architecture the * signifies hardware memory limitations due to the arrangement of architecture topology and thus expansion is limited. This is because increased amount of early-layer features requires more memory in contrast to late layers, which is particularly intense for the coupled WRN architecture.

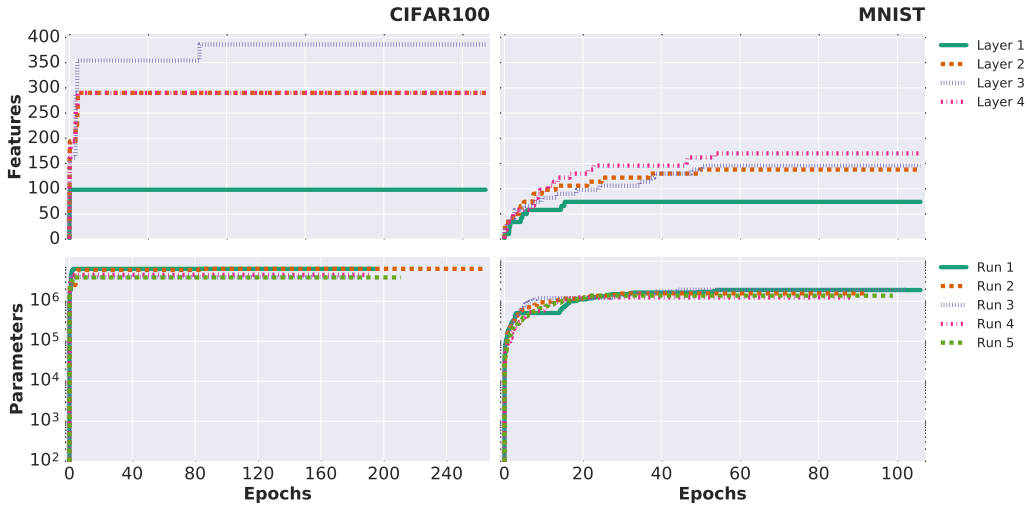


Figure 2: Exemplary GFCNN network expansion on MNIST and CIFAR100. Top panel shows one architecture's individual layer expansion; bottom panel shows the evolution of total parameters for five runs. It is observable how different experiments converge to similar network capacity on slightly different time-scales and how network capacity systematically varies with complexity of the dataset.

Table 1: Mean error and standard deviation and number of parameters (in millions) for architectures trained five times using the original reference and automatically expanded architectures respectively. For MNIST no data augmentation is applied. We use minor augmentation (flips, translations) for CIFAR10 & 100. All-convolutional (all-conv) versions have been evaluated for each architecture (except WRN where convolutions are stacked already). The * indicates hardware limitation.

			GFCNN		VGG-A		VGG-E		WRN-28-10	
	type		original	expanded	original	expanded	original	expanded	original	expanded
MNIST	standard	error [%]	0.487	0.528 ±0.03	0.359	0.394 ±0.05	0.386	0.388 ±0.03	overfit	0.392±0.05
		params [M]	4.26	1.61 ±0.31	13.70	3.35 ±0.45	20.57	6.49 ±0.89	36.48	4.83 ±0.57
		time [h]	0.29	0.90 ±0.32	0.59	5.35 ±1.95	0.48	9.8 ±3.02	2.47	2.91 ±0.28
	all-conv	error [%]	0.535	0.552 ±0.03	0.510	0.502 ±0.04	0.523	0.528 ±0.04		
		params [M]	3.71	2.19 ±0.55	10.69	3.79 ±0.57	21.46	6.02 ±0.73		
		time [h]	0.39	1.68 ±1.05	0.64	3.06 ±0.97	0.52	6.58 ±2.81		
CIFAR10+	standard	error [%]	11.32	11.03 ±0.19	7.18	6.73 ±0.05	7.51	5.64 ±0.11	4.04	3.95 ±0.12
		params [M]	4.26	4.01 ±0.62	13.70	8.54 ±1.51	20.57	27.41 ±4.09	36.48	25.30 ±1.62
		time [h]	0.81	1.40 ±0.22	1.61	3.95 ±0.59	1.32	16.32 ±5.39	8.22	21.18 ±2.12
	all-conv	error [%]	8.78	8.13 ±0.11	6.71	6.56 ±0.18	7.46	5.42 ±0.11		
		params	3.71	10.62 ±1.91	10.69	8.05 ±1.57	21.46	44.98 ±7.31		
		time [h]	1.19	3.38 ±0.76	1.74	5.24 ±1.11	1.54	26.46 ±9.77		
CIFAR100+	standard	error [%]	34.91	34.23 ±0.29	25.01	25.17 ±0.34	29.43	25.06 ±0.55	18.51	18.44*
		params [M]	4.26	6.82 ±1.08	13.70	8.48 ±1.40	20.57	28.41 ±2.26	36.48	27.75*
		time [h]	0.81	1.83 ±0.56	1.61	3.83 ±0.47	1.32	16.67 ±2.89	8.22	13.9*
	all-conv	error [%]	29.83	28.34 ±0.43	24.30	23.95 ±0.28	31.94	24.87 ±0.16		
		params [M]	3.71	21.40 ±3.71	10.69	10.84 ±2.41	21.46	44.59 ±4.49		
		time [h]	1.19	4.72 ±1.15	1.74	5.38 ±1.46	1.54	22.76 ±3.94		

- Even though we have introduced re-intialization of the architecture the time taken by algorithm 1 is much less than one would invest when doing a manual, grid- or random-search.
- Shallow GFCNN architectures are able to gain accuracy by increasing layer width, although there seems to be a natural limit to what width alone can do. This is in agreement with observations pointed out in other works such as Ba & Caurana (2014); Urban et al. (2017).
- The large reference VGG-E (lower accuracy than VGG-A on CIFAR) and WRN-28-10 (complete overfit on MNIST) seem to run into optimization difficulties for these datasets. However, expanded alternate architecture clearly perform significantly better.

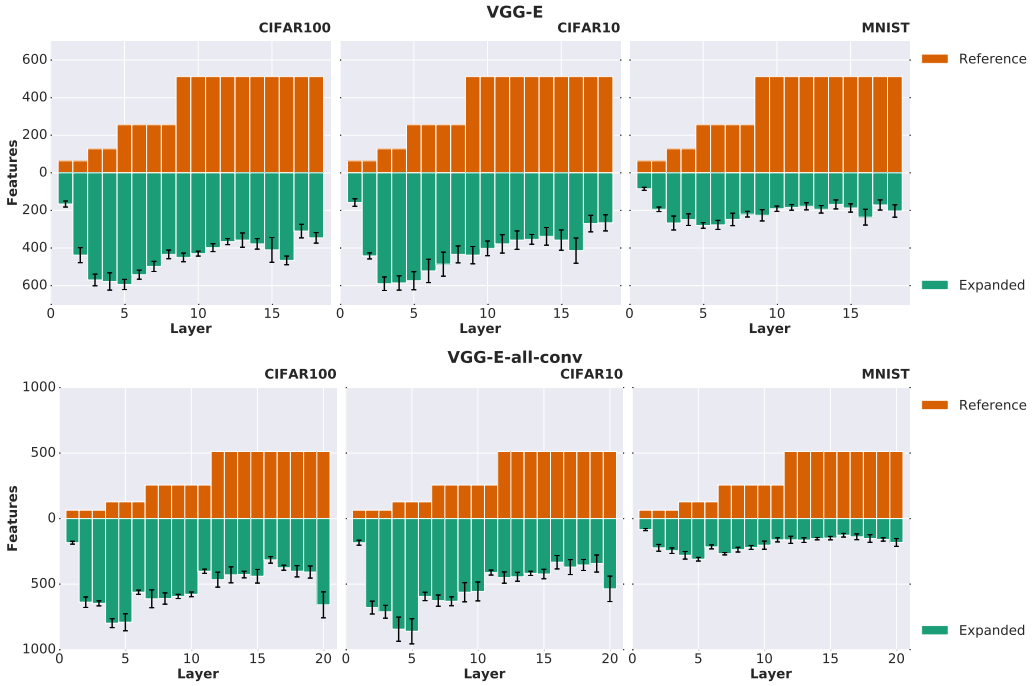


Figure 3: Mean and standard deviation of topologies as evolved from the expansion algorithm for a VGG-E and VGG-E all-convolutional architecture run five times on MNIST, CIFAR10 and CIFAR100 datasets respectively. Top panels show the reference architecture, whereas bottom shows automatically expanded architecture alternatives. Expanded architectures vary in capacity with dataset complexity and topologically differ from their reference counterparts.

In general we observe that these benefits are due to unconventional, yet always coinciding, network topology of our expanded architectures. These topologies suggest that there is more to CNNs than simply following the rule of thumb of increasing the number of features with increasing architectural depth. Before proceeding with more detail on these alternate architecture topologies, we want to again emphasize that we do not report experiments containing extended methodology such as excessive preprocessing, data augmentation, the oscillating learning rates proposed in Loshchilov & Hutter (2017) or better sets of hyper-parameters for reasons of clarity, even though accuracies rivaling state-of-the-art performances can be achieved in this way.

3.3 Alternate formation of deep neural network topologies

Almost all popular convolutional neural network architectures follow a design pattern of monotonically increasing feature amount with increasing network depth (LeCun et al., 1998; Goodfellow et al., 2013; Simonyan & Zisserman, 2015; Springenberg et al., 2015; He et al., 2016; Zagoruyko & Komodakis, 2016; Loshchilov & Hutter, 2017; Urban et al., 2017). For the results presented in table 1 all automatically expanded network topologies present alternatives to this pattern. In figure 3, we illustrate exemplary mean topologies for a VGG-E and VGG-E all-convolutional network as constructed by our expansion algorithm in five runs on the three datasets. Apart from noticing the systematic variations in representational capacity with dataset difficulty, we furthermore find topological convergence with small deviations from one training to another. We observe the highest feature dimensionality in early to intermediate layers with generally decreasing dimensionality towards the end of the network differing from conventional CNN design patterns. Even if the expanded architectures sometimes do not deviate much from the reference parameter count, accuracy seems to be improved through this topological re-arrangement. For architectures where pooling has been replaced with larger stride convolutions we also observe that dimensionality of layers with sub-sampling changes independently of the prior and following convolutional layers suggesting that highly-complex sub-sampling operations are learned. This an extension to the proposed

all-convolutional variant of Springenberg et al. (2015), where introduced additional convolutional layers were constrained to match the dimensionality of the previously present pooling operations.
If we view the deep neural network as being able to represent any function that is limited rather by concepts of continuity and boundedness instead of a specific form of parameters, we can view the minimization of the cost function as learning a functional mapping instead of merely adopting a set of parameters (Goodfellow et al., 2016). We hypothesize that evolved network topologies containing higher feature amount in early to intermediate layers generally follow a process of first mapping into higher dimensional space to effectively separate the data into many clusters. The network can then more readily aggregate specific sets of features to form clusters distinguishing the class subsets. Empirically this behavior finds confirmation in all our evolved network topologies that are visualized in the appendix. Similar formation of topologies, restricted by the dimensionality constraint of the identity mappings, can be found in the trained residual networks.
While He et al. (2015) has shown that deep VGG-like architectures do not perform well, an interesting question for future research could be whether plainly stacked architectures can perform similarly to residual networks if the arrangement of feature dimensionality is differing from the conventional design of monotonic increase with depth.

3.4 An outlook to ImageNet

We show two first experiments on the ImageNet dataset using an all-convolutional Alexnet to show that our methodology can readily be applied to large scale. The results for the two runs can be found in table 2 and corresponding expanded architectures are visualized in the appendix. We observe that the experiments seem to follow the general pattern and again observe that topological rearrangement of the architecture yields substantial benefits. In the future we would like to extend experimentation to more promising ImageNet architectures such as deep VGG and residual networks. However, these architectures already require 4-8 GPUs and large amounts of time in their baseline evaluation, which is why we presently are not capable of evaluating these architectures and keep this section at a very brief proof of concept level.

Table 2: Two experiments with all-convolutional Alexnet on the large scale Imagenet dataset comparing the reference implementation with our expanded architecture.

	Alexnet - 1				Alexnet - 2			
	top-1 error	top-5 error	params	time	top-1 error	top-5 error	params	time
original	43.73 %	20.11 %	35.24 M	27.99 h	43.73 %	20.11 %	35.24 M	27.99 h
expanded	37.84 %	15.88 %	34.76 M	134.21 h	38.47 %	16.33 %	32.98 M	118.73 h

4 Conclusion

In this work we have introduced a novel bottom-up algorithm to start neural network architectures with one feature per layer and widen them until a task depending suitable representational capacity is achieved. For the use in this framework we have presented one potential computationally efficient and intuitive metric to gauge feature importance. The proposed algorithm is capable of expanding architectures that provide either reduced amount of parameters or improved accuracies through higher amount of representations. This advantage seems to be gained through alternative network topologies with respect to commonly applied designs in current literature. Instead of increasing the amount of features monotonically with increasing depth of the network, we empirically observe that expanded neural network topologies have high amount of representations in early to intermediate layers.
Future work could include a re-evaluation of plainly stacked deep architectures with new insights on network topologies. We have furthermore started to replace the currently present re-initialization step in the proposed expansion algorithm by keeping learned filters. In principle this approach looks promising but does need further systematic analysis of new feature initialization with respect to the already learned feature subset and accompanied investigation of orthogonality to avoid falling into local minima.

ACKNOWLEDGEMENTS

This work has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 687384. Kishora Konda and Tobias Weis received funding from Continental Automotive GmbH. We would like to further thank Anjaneyalu Thippaiah for help with execution of ImageNet experiments.

REFERENCES

Jose M. Alvarez and Mathieu Salzmann. Learning the Number of Neurons in Deep Networks. In *NIPS*, 2016.

Lei J. Ba and Rich Caurana. Do Deep Nets Really Need to be Deep ? *arXiv preprint arXiv:1312.6184*, 2014.

Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing Neural Network Architectures using Reinforcement Learning. *arXiv preprint arXiv:1611.02167*, 2016.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. *BigLearn, NIPS Workshop*, 2011.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout Networks. In *ICML*, 2013.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both Weights and Connections for Efficient Neural Networks. In *NIPS*, 2015.

Song Han, Huizi Mao, and William J. Dally. Deep Compression - Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *ICLR*, 2016.

Song Han, Huizi Mao, Enhao Gong, Shijian Tang, William J. Dally, Jeff Pool, John Tran, Bryan Catanzaro, Sharan Narang, Erich Elsen, Peter Vajda, and Manohar Paluri. DSD: Dense-Sparse-Dense Training For Deep Neural Networks. In *ICLR*, 2017.

Li Hao, Asim Kadav, Hanan Samet, Igor Durdanovic, and Hans Peter Graf. Pruning Filters For Efficient Convnets. In *ICLR*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning Workshop*, 2014.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arxiv preprint arXiv:1502.03167*, 2015.

Guoliang Kang, Jun Li, and Dacheng Tao. Shakeout: A New Regularized Deep Neural Network Training Scheme. In *AAAI*, 2016.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu. Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree. 2015.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent With Warm Restarts. In *ICLR*, 2017.

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Quoc Le, and Alex Kurakin. Large-Scale Evolution of Image Classifiers. *arXiv preprint arXiv:1703.01041*, 2017.

Pau Rodriguez, Jordi González, Guillem Cucurull, Josep M. Gonfaus, and Xavier Roca. Regularizing CNNs With Locally Constrained Decorrelations. In *ICLR*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

Mohammad J. Shafiee, Akshaya Mishra, and Alexander Wong. EvoNet: Evolutionary Synthesis of Deep Neural Networks. *arXiv preprint arXiv:1606.04393*, 2016.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not Just a Black Box: Interpretable Deep Learning by Propagating Activation Differences. In *ICML*, 2016.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

Jost T. Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. In *ICLR*, 2015.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *JMLR*, 15, 2014.

Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Abdel-rahman Mohamed, Matthai Philipose, Matthew Richardson, and Rich Caruana. Do Deep Convolutional Nets Really Need To Be Deep And Convolutional? In *ICLR*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *BMVC*, 2016.

A APPENDIX

A.1 DATASETS

- MNIST (LeCun et al., 1998): 50000 train images of hand-drawn digits of spatial size 28×28 belonging to one of 10 equally sampled classes.
- CIFAR10 & 100 (Krizhevsky, 2009): 50000 natural train images of spatial size 32×32 each containing one object belonging to one of 10/100 equally sampled classes.
- ImageNet (Russakovsky et al., 2015): Approximately 1.2 million training images of objects belonging to one of 1000 classes. Classes are not equally sampled with 732-1300 images per class. Dataset contains 50 000 validation images, 50 per class. Scale of objects and size of images varies.

A.2 TRAINING HYPER-PARAMETERS

All training is closely inspired by the procedure specified in Zagoruyko & Komodakis (2016) with the main difference of avoiding heavy preprocessing. Independent of dataset, we preprocess all data using only trainset mean and standard deviation. All training has been conducted using cross-entropy as a loss function and weight initialization following the normal distribution as proposed by He et al. (2015). All architectures are trained with batch-normalization with a constant of $1 \cdot 10^{-3}$, a batch-size of 128, a L_2 weight-decay of $5 \cdot 10^{-4}$, a momentum of 0.9 and nesterov momentum.

Small datasets: We use initial learning rates of 0.1 and 0.005 for the CIFAR and MNIST datasets respectively. We have rescaled MNIST images to 32×32 (CIFAR size) and repeat the image across color channels in order to use architectures without modifications. CIFAR10 & 100 are trained for 200 epochs and the learning rate is scheduled to be reduced by a factor of 5 every multiple of 60 epochs. MNIST is trained for 60 epochs and learning rate is reduced by factor of 5 once after 30 epochs. We augment the CIFAR10 & 100 training by introducing horizontal flips and small translations of up to 4 pixels during training. No data augmentation has been applied to the MNIST dataset.

ImageNet: We use the single-crop technique where we rescale the image such that the shorter side is equal to 224 and take a centered crop of spatial size 224×224. In contrast to Krizhevsky et al. (2012) we limit preprocessing to subtraction and divison of trainset mean and standard deviation and do not include local response normalization layers. We randomly augment training data with random horizontal flips. We set an initial learning rate of 0.1 and follow the learning rate schedule proposed in Krizhevsky et al. (2012) that drops the learning rate by a factor of 0.1 every 30 epochs and train for a total of 74 epochs.

The amount of epochs for the expansion of architectures is larger due to the re-initialization. For these architectures the mentioned amount of epochs corresponds to training during stable conditions, i.e. no further expansion. The procedure is thus equivalent to training the converged architecture from scratch.

A.3 ARCHITECTURES

GFCNN (Goodfellow et al., 2013) Three convolution layer network with larger filters (followed by two fully-connected layers, but without "maxout". The exact sequence of operations is:

1. Convolution 1: $8 \times 8 \times 128$ with padding $= 4 \rightarrow$ batch-normalization $\rightarrow$ ReLU $\rightarrow$ max-pooling 4×4 with stride $= 2$.
2. Convolution 2: $8 \times 8 \times 198$ with padding $= 3 \rightarrow$ batch-normalization $\rightarrow$ ReLU $\rightarrow$ max-pooling 4×4 with stride $= 2$.
3. Convolution 3: $5 \times 5 \times 198$ with padding $= 3 \rightarrow$ batch-normalization $\rightarrow$ ReLU $\rightarrow$ max-pooling 2×2 with stride $= 2$.
4. Fully-connected 1: $4 \times 4 \times 198 \rightarrow 512 \rightarrow$ batch-normalization $\rightarrow$ ReLU.
5. Fully-connected 2: $512 \rightarrow$ classes.

Represents the family of rather shallow "deep" networks.

VGG (Simonyan & Zisserman, 2015) "VGG-A" (8 convolutions) and "VGG-E" (16 convolutions) networks. Both architectures include three fully-connected layers. We set the number of features in the MLP to 512 features per layer instead of 4096 because the last convolutional layer of these architecture already produces outputs of spatial size 1×1 (in contrast to 7×7 on ImageNet) on small datasets. Batch normalization is used before the activation functions. Examples of stacking convolutions that do not alter spatial dimensionality to create deeper architectures.

WRN (Zagoruyko & Komodakis, 2016) Wide Residual Network architecture: We use a depth of 28 convolutional layers (each block completely coupled, no bottlenecks) and a width-factor of 10 as reference. When we expand these networks this implies an inherent coupling of layer blocks due to dimensional consistency constraints with outputs from identity mappings.

Alexnet (Krizhevsky et al., 2012) We use the all convolutional variant where we replace the first fully-connected large $6 \times 6 \times 256 \rightarrow 4096$ layer with a convolution of corresponding spatial filter size and 256 filters and drop all further fully-connected layers. The rationale behind this decision is that previous experiments, our own pruning experiments and those of Hao et al. (2017); Han et al. (2015), indicate that original fully-connected layers are largely obsolete.

A.4 Automatically expanded architecture topologies

In addition to figure 3 we show mean evolved topologies including standard deviation for all architectures and datasets reported in table 1 and 2. In figure 4 and 5 all shallow and VGG-A architectures and their respective all-convolutional variants are shown. Figure 6 shows the constructed wide residual 28 layer network architectures where blocks of layers are coupled due to the identity mappings. Figure 7 shows the two expanded Alexnet architectures as trained on ImageNet.
As explained in the main section we see that all evolved architectures feature topologies with large dimensionality in early to intermediate layers instead of in the highest layers of the architecture as usually present in conventional CNN design. For architectures where pooling has been replaced with larger stride convolutions we also observe that dimensionality of layers with sub-sampling changes independently of the prior and following convolutional layers suggesting that highly-complex pooling operations are learned. This an extension to the proposed all-convolutional variant of Springenberg et al. (2015), where introduced additional convolutional layers were constrained to match the dimensionality of the previously present pooling operations.

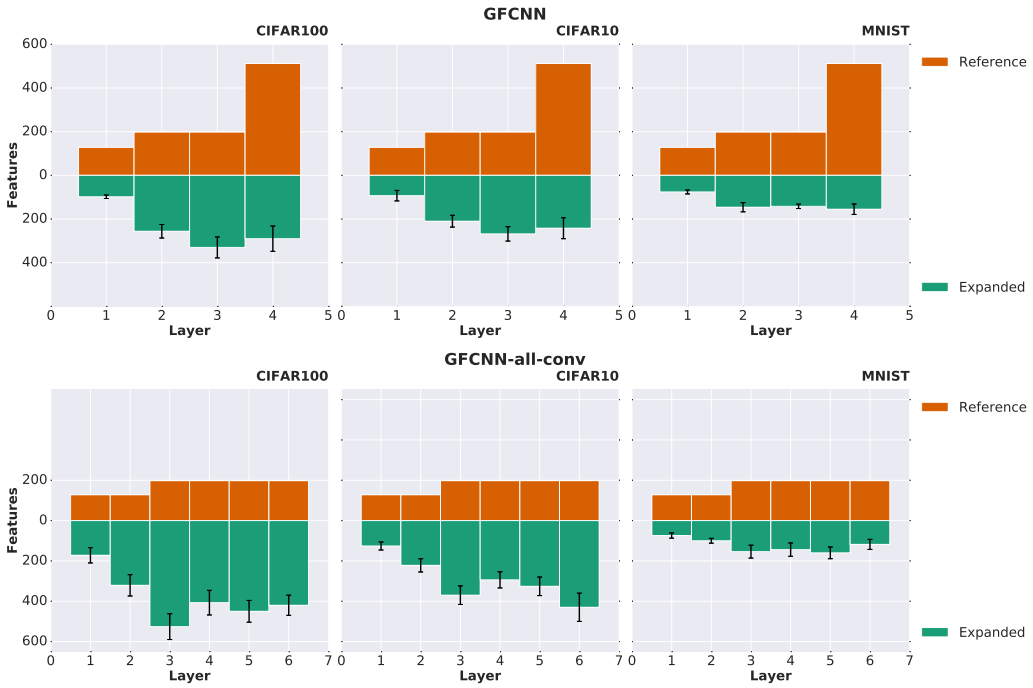


Figure 4: Mean and standard deviation of topologies as evolved from the expansion algorithm for the shallow networks run five times on MNIST, CIFAR10 and CIFAR100 datasets respectively.

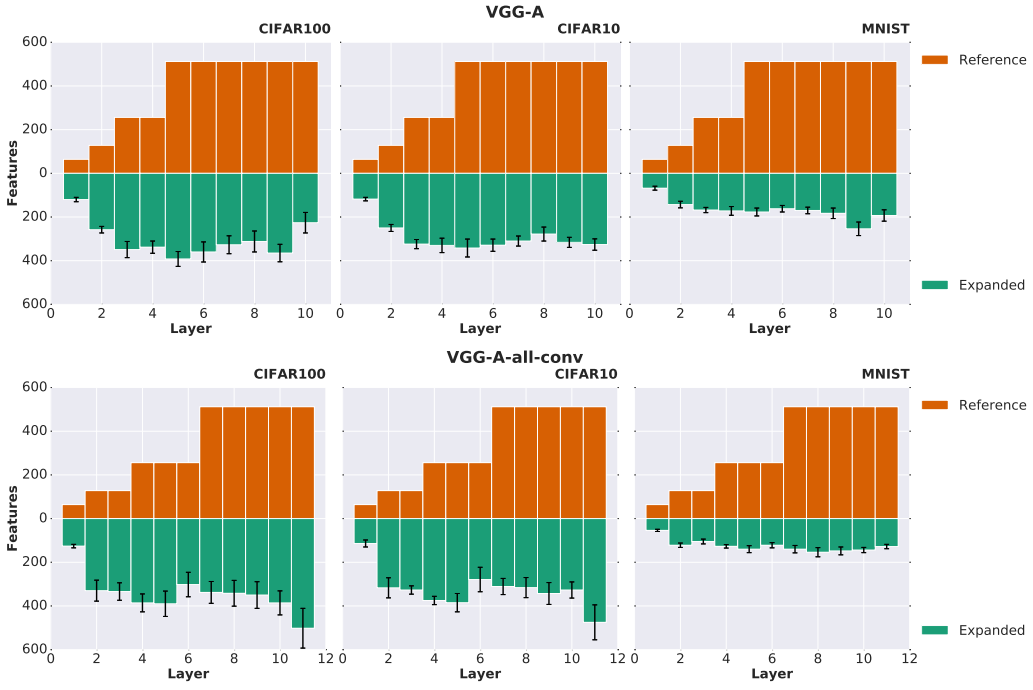


Figure 5: Mean and standard deviation of topologies as evolved from the expansion algorithm for the VGG-A style networks run five times on MNIST, CIFAR10 and CIFAR100 datasets respectively.

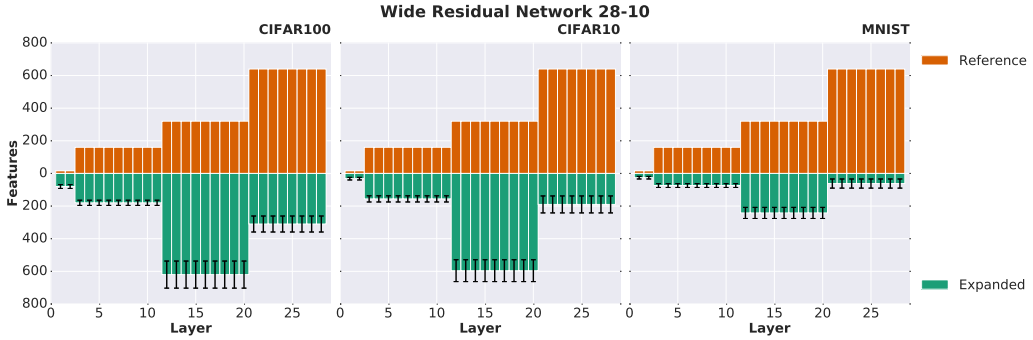


Figure 6: Mean and standard deviation of topologies as evolved from the expansion algorithm for the WRN-28 networks run five times on MNIST, CIFAR10 and CIFAR100 datasets respectively. Note that the CIFAR100 architecture was limited in expansion by hardware.

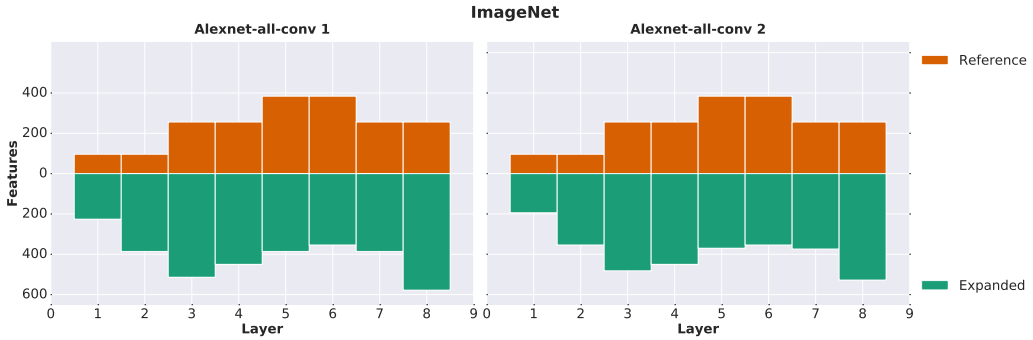


Figure 7: Architecture topologies for the two all-convolutional Alexnets of table 2 as evolved from the expansion algorithm on ImageNet.

Rethinking Layer-wise Feature Amounts in Convolutional Neural Network Architectures

Martin Mundt
Goethe University
Frankfurt Institute for Advanced Studies
mmundt@em.uni-frankfurt.de

Sagnik Majumder
Goethe University
majumder@ccc.cs.uni-frankfurt.de

Tobias Weis
Goethe University
weis@ccc.cs.uni-frankfurt.de

Visvanathan Ramesh
Goethe University
Frankfurt Institute for Advanced Studies
vramesh@em.uni-frankfurt.de

Abstract

We characterize convolutional neural networks with respect to the relative amount of features per layer. Using a skew normal distribution as a parametrized framework, we investigate the common assumption of monotonously increasing feature-counts with higher layers of architecture designs. Our evaluation on models with VGG-type layers on the MNIST, Fashion-MNIST and CIFAR-10 image classification benchmarks provides evidence that motivates rethinking of our common assumption: architectures that favor larger early layers seem to yield better accuracy.

1 Introduction and motivation

Deep learning practices that are empirically confirmed to be valuable often turn into rules of thumb to be used by the community. One such rule of thumb is the historically grown custom of increasing the number of features (for convolutions synonymous with kernel or filter) with increasing depth of a convolutional neural network (CNN). Perpetuated by perhaps the simplicity and large success of the VGG architecture [1], more recent work such as residual networks [2] or densely connected networks [3] still follow this design principle. While such works achieve progress through modifying connectivity structure, changing the task or depth of the network, we, the machine learning community, tend to leave the principle of stacking 3×3 convolutions with monotonously increasing feature amounts per layer untouched. For other advances in tasks such as semantic image segmentation [4, 5], the encoder strictly follows this pattern and on top mirrors the pattern in the decoder. Even though some work, such as the "network in network" architecture [6], deviates and explores alternatives in design, many architectures [2–5] seem to inherit the simple VGG-style of keeping or doubling the amount of features from one layer to another. Apart from the empirically demonstrated effectiveness, a core assumption can be hypothesized as follows: lower layers of CNNs learn more primitive features whereas higher layers learn more abstract features. Thus, our assumption could be to increase the amount of learnable features in higher layers to in turn provide enough representational capacity for a rich encoding.

In this work we propose a simple three-parameter univariate skew normal distribution to parametrize a family of neural networks. By changing the distribution's parameters, we shift a constant amount of features and map them to architectures with monotonously decreasing, increasing and normally distributed feature amounts per layer. While the exact choice of distribution is of empirical nature, a three-dimensional mathematical description allows for an intuitive model characterization. We train 200 model variants by conducting a grid-search on the distribution's parameters on three

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

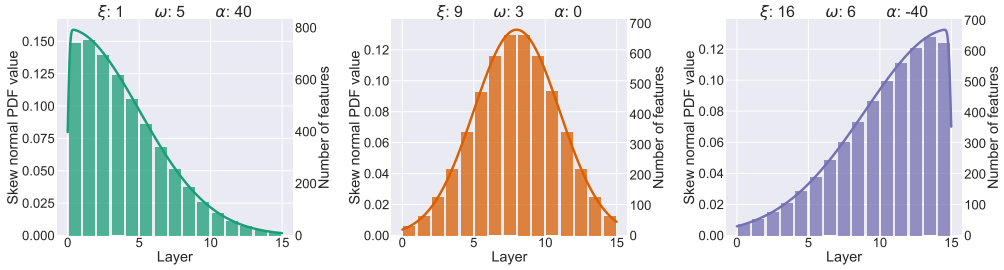


Figure 1: Three examples of skew normal PDFs (solid line), integrated layer bins and mapped amount of features. All three architectures have the same amount of overall features. Architectures with PDF parameters as depicted in the right panel most resemble traditional CNN designs. Architectures parametrized by the mid and specifically the left panel are not commonly found in the literature.

popular image classification datasets: MNIST [7], CIFAR-10 [8], Fashion-MNIST [9]. We show that the commonly picked subset of monotonously increasing feature amounts per layer seems to be outperformed in terms of accuracy by architectures that favor larger early layers. We hope to inspire to rethink our CNN design intuition and to stimulate further analysis for future models.

2 Parametrizing distribution of features across layers

For the purpose of parametrization and characterization of common and uncommon CNN design, we have chosen the probability density function (PDF) of the univariate three parameter skew normal distribution. We use three parameters in order to be able to generate curves with varying location of the maximum peak with different sharpness, as defined by the location (mean) ξ and scale (variance) ω respectively. We also require the shape (skew) α to adjust the slope in positive or negative direction. This results in the following PDF:

$$\frac{1}{2\pi}\frac{2}{\omega\sqrt{2\pi}}e^{-\frac{(x-\xi)^2}{2\omega^2}}\frac{1}{2}[1+erf(\frac{\alpha\frac{x-\xi}{\omega}}{\sqrt{2}})] \tag{1}$$

Here, $erf(x)$ is the error function. To apply this PDF to CNNs, we specify the number of layers and overall features, e.g. the total number of features in a 16 layered 3×3 convolutional VGG-D architecture. We then use trapezoidal integration to calculate one integrated value per layer. The resulting discretized distribution is scaled by the overall number of features. Using this process we can generate a family of architectures while keeping the number of layers and overall amount of features constant. We visualize three examples of the PDF, the integrated discretized layer bins, as well as the amount of features per layer in figure 1. The figure shows three examples, corresponding to architectures with maximum amount of features in the first, middle and last layers. The latter, depicted in the right panel, is an example that is similar to the original design of the VGG-D architecture.

3 Characterization of VGG filter distributions

Generated grid of architectures: We generate a set of architectures using previously described process by creating a discretized grid of ξ, ω, α values. Specifically we let ξ be in the interval $[1, 16]$, in steps of 1, to generate 16 layer VGG-D like architectures with different feature maximum locations. We vary the scale ω in the interval $[0.5, 5.5]$ in steps of 0.5 and the shape α in the interval $[-40, 40]$ in steps of 4. That is, we keep the network's functional sequence (including pooling and activation functions and last two fully-connected layers) the same as the original VGG-D architecture and only redistribute the features across different layers. Not all combinations of ξ, ω, α are considered "valid" as the resulting integral would violate the assumption of keeping the amount of features constant. We thus only take into account combinations that do not lower the total amount of features by more than 5%. These parameters result in 203 architectures trained on each dataset.

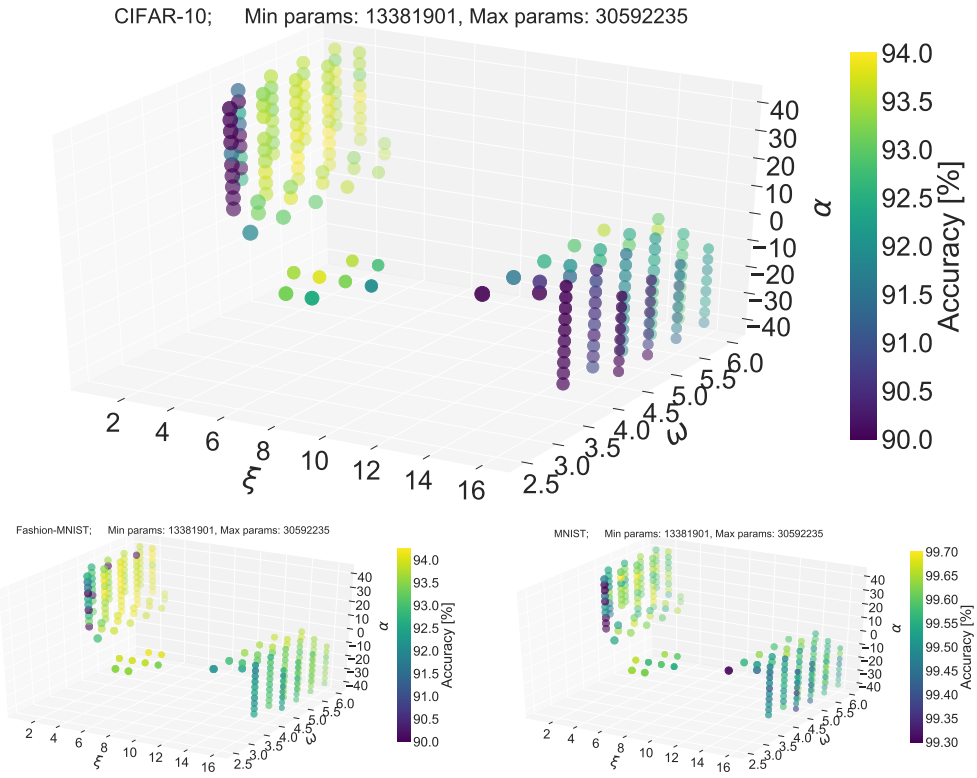


Figure 2: Validation accuracy (in color) of 16-layer VGG-type architectures parametrized through combinations of parameters ξ, ω, α of a univariate skew normal distribution. While all models have approximately the same amount of features, total parameter amounts can vary as indicated by marker size. The accuracy range has been cut-off at the bottom for better visual perception.

Training hyper-parameters: We train all networks for 150 epochs for CIFAR-10, and 30 epochs for MNIST and Fashion-MNIST using the weight initialization of He et al [10]. To make sure that all networks are able to train to convergence, we include batch-normalization with a value of 10^{-4} [11] and cycle the learning rate with warm restarts [12]. We start with an initial learning rate of 10^{-2} and continuously lower it to 10^{-5} with a restart cycle of 10 epochs, that is then doubled after each restart. To be consistent with evaluation in the literature, we train using a batch size of 128, a weight-decay of $5 \cdot 10^{-4}$ and apply horizontal flip and four pixel random translation data augmentation to the CIFAR-10 data. MNIST and Fashion-MNIST images are resized to 32×32 to allow for the use of the same architectures. No further data pre-processing or augmentation is applied.

Results: The validation accuracy for trained architectures parametrized by ξ, ω, α is shown in figure 2. We remind the reader that all architectures approximately have the same amount of overall features. Depending on the precise distribution of features the representational capacity can vary. The total amount of parameters is therefore also encoded by marker sizes. Note that for the majority of architectures there is minor variation, with exception of the edge cases where a large amount of features is attributed in the very last two fully-connected layers, where the overall amount of parameters is then smaller. In all three examples, most clearly for CIFAR-10 due to larger accuracy variation, we observe the following trends:

- The accuracy rises with lower ξ value, i.e. architectures that favor larger amounts of features in early layers seem to achieve better accuracy.
- The accuracy rises with higher ω value. This is because low scale values lead to tails of the distribution that map to very little overall amount of features, e.g. only 2 features in a layer.

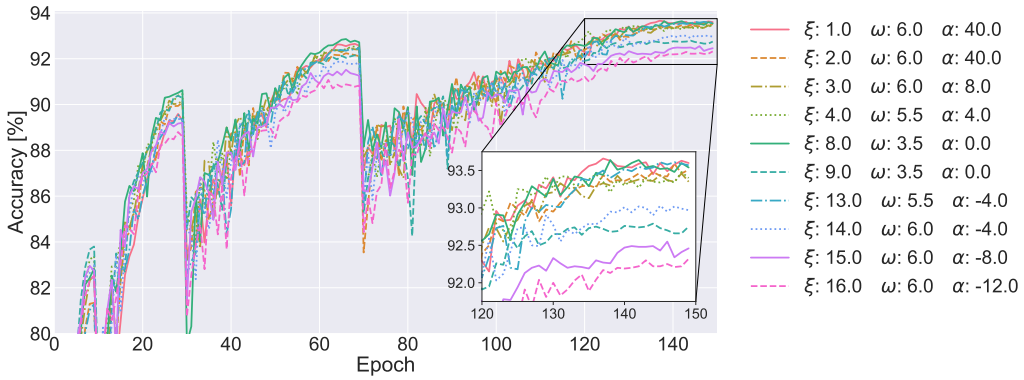


Figure 3: Validation accuracy per epoch of 16-layer VGG-type architectures parametrized through combinations of parameters ξ, ω, α of a univariate skew normal distribution. The best architectures per ξ are visualized to demonstrate the trend that lower values of ξ are correlated with higher accuracy.

For small or large values of ξ, we also observe a constraint of α to a positive or negative range respectively, to make sure that the overall amount of features stays approximately constant. The changing number of total parameters at constant feature amount highlights a different (mal)practice in CNN design, where we generally design amounts of unique features independently of spatial kernel dimensions or questioning the effects on overall parameter count. To emphasize the differences in accuracy, we visualize the best CIFAR-10 architecture per ξ in figure 3. The tendency of rising accuracy with lower ξ is in contrast with our common assumption of increasing, or even doubling the amount of features as we progress deeper into the CNN layers. We remark that all models converge after 150 epochs. However, the hyper-parameters are selected based on original VGG architectures (i.e. large ξ) and not tuned to best fit presented small ξ variants. Additional experiments with 10 VGG-type layers confirm described trends. Due to space constraints we include these results with the open-source code for this work: `https://github.com/MrtnMndt/Rethinking_CNN_Layerwise_Feature_Amounts`.

After analysis of the results presented in figure 3, we note that the middle range of ξ is difficult to parametrize due to a non constant total number of features for many distribution parameter configurations. In hindsight, one idea could thus be to use a distribution with constant probability mass as the parameters change. One such distribution for further experimentation could be the Beta distribution, with layers binned to equally sized intervals in the $[0, 1]$ range.

4 Conclusion

We have parametrized CNN architectures with respect to their relative amounts of features per layer using a skew normal distribution. Although further investigation with larger datasets is necessary, our experiments indicate that our historically grown assumption of increasing layer-wise feature counts with increasing network depth is challenged by architectures that favor large early layers. While it isn't emphasized in the original work, architectures generated through the recent trend of reinforcement learning based search seem to be in favor of this trend [13]. It will thus be interesting to extend our examination to models with skip connections to see if a similar conclusion hold. A remaining crucial open question is the reason behind the observed pattern. Is it simply that using too few features in initial layers acts as a bottleneck, making it harder for the remaining layers to retrieve the information about the image that is necessary for classification? Or is there a deeper reason? We motivate to rethink this design principle and more thoroughly analyse future CNN designs.

5 Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 687383 .

References

[1] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.243.

[4] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, Vol. 9351:234–241, 2015.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *PAMI*, 39(12):2481–2495, 2017.

[6] M. Lin, C. Qiang, and Y. Shuicheng. Network In Network. In *ICLR*, 2014.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

[8] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Toronto, 2009.

[9] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv: 1708.07747*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[11] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, volume 37, pages 448–456, 2015.

[12] I. Loshchilov and F. Hutter. SGDR: Stochastic Gradient Descent With Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[13] Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Meta-learning Convolutional Neural Architectures for Multi-target Concrete Defect Classification with the COncrete DEfect BRidge IMage Dataset

Martin Mundt[1*], Sagnik Majumder[1], Sreenivas Murali[1*], Panagiotis Panetsos[2], Visvanathan Ramesh[1*]
1. Goethe University 2. Egnatia Odos A. E.
{mmundt, vramesh}@em.uni-frankfurt.de {majumder, murali}@ccc.cs.uni-frankfurt.de
ppane@egnatia.gr

Abstract

Recognition of defects in concrete infrastructure, especially in bridges, is a costly and time consuming crucial first step in the assessment of the structural integrity. Large variation in appearance of the concrete material, changing illumination and weather conditions, a variety of possible surface markings as well as the possibility for different types of defects to overlap, make it a challenging real-world task. In this work we introduce the novel COncrete DEfect BRidge IMage dataset (CODEBRIM) for multi-target classification of five commonly appearing concrete defects. We investigate and compare two reinforcement learning based meta-learning approaches, MetaQNN and efficient neural architecture search, to find suitable convolutional neural network architectures for this challenging multi-class multi-target task. We show that learned architectures have fewer overall parameters in addition to yielding better multi-target accuracy in comparison to popular neural architectures from the literature evaluated in the context of our application.

1. Introduction

To assess a concrete bridge's structural safety, it is desirable to determine the level of degradation by accurately recognizing all defect types. Defects tend to be small with respect to bridge elements and often occur simultaneously with overlap of defect categories. Although one could imagine treating each defect category independently, overlapping defects are more severe with respect to the structural safety. The requirement to recognize these multi-class multi-target defects forms the basis for a challenging real-world task that is further complicated by a variety of environmental factors. Concrete, as a composite material, has a wide range of variation in surface reflectance, roughness, color and, in some cases, applied surface coatings. Changing lighting conditions, weather dependent wetness of the surface and a diverse set of safety irrelevant surface alterations like small holes, markings, stains or graffiti, add to the factors of variation. This necessitates computer vision techniques that are capable of addressing such rich appearance spaces.

Deep learning techniques in conjunction with labelled datasets have turned out to be ideal candidates for recognition tasks of similar complexity. Especially convolutional neural networks (CNNs) [21, 32, 1, 37, 16] have been shown to excel at object and material recognition benchmarks [29, 10, 35, 3]. Unfortunately, defect recognition in concrete bridges is largely yet to benefit from deep learning approaches. Due to the necessity of expert knowledge in the annotation process along with tedious image acquisition, the task is traditionally focused on cracks with algorithms based on domain specific modelling or manual inspection by a human. Recently, datasets [31, 36, 26] and corresponding deep learning applications [36, 23, 18, 8] have presented significant efforts towards data-driven approaches in this domain. Their work focuses on cracks as only a subset of structurally relevant defects and concentrates on CNNs proposed in the object recognition literature, that might not be the best choice for material defect recognition.

In this work we address two crucial open aspects of concrete defect recognition: the establishment of a labelled multi-target dataset with overlapping defect categories for use in machine learning and the design of deep neural networks that are tailored to the task. For this purpose we introduce our novel COncrete DEfect BRidge IMage (CODEBRIM) dataset and employ meta-learning of CNN architectures specific to multi-class multi-target defect classification. CODEBRIM features six mutually non-exclusive classes: crack, spallation, efflorescence, exposed bars, corrosion (stains) and non-defective background. Our images were acquired at high-resolution, partially using an unmanned aerial vehicle (UAV) to gain close-range access, and feature varying scale and context. We evaluate a variety of best-practice CNN architectures [21, 32, 1, 37, 16] in the literature on the CODEBRIM's multi-target defect

* work conducted while at Frankfurt Institute for Advanced Studies

recognition task. We show that meta-learned neural architectures achieve equivalent or better accuracies, while being more parameter efficient, by investigating and comparing two reinforcement learning neural architecture search approaches: MetaQNN [2] and "efficient neural architecture search" (ENAS) [27]. The CODEBRIM dataset is publicly available at: https://doi.org/10.5281/zenodo.2620293 . We also make the code for training the CNN baselines and both meta-learning techniques available open-source at: https://github.com/MrtnMndt/meta-learning-CODEBRIM . To summarize our contributions:

- We introduce a novel high-resolution multi-class multi-target dataset featuring images of defects in context of concrete bridges.
- We evaluate and compare best-practice CNN architectures for the task of multi-target defect classification.
- We adapt and contrast two reinforcement learning based architecture search methods, MetaQNN and ENAS, on our multi-target scenario. We show how resulting meta-learned architectures from both methods improve the presented task in terms of higher accuracy and lower model parameter count.

2. Prior and related work

Datasets. Image classification and object detection benchmarks predominantly focus on the single-target scenario. Popular examples are the ImageNet [29], Pascal VOC [10] or the scene understanding SUN dataset [35], where the task is to assign a specific class to an image, area or pixel. Much of the recent computer vision deep learning research is built upon improvements based on these publicly available datasets. The "materials in context" database (MINC) [3] followed in spirit and has created a dataset for material and texture related recognition tasks. To a large degree MINC has extended previous datasets and applications built upon prior work of the (CUReT) database [9], the FMD dataset [30] and KTH-TIPS [11, 5]. With respect to defects in concrete structures, or bridges in particular, openly available datasets remain scarce. Depending on the defect type that needs to be recognized, our task combines texture anomalies such as efflorescence or cracks with objects such as exposed reinforcement bars. Domain specific dataset contributions were very recently proposed with the "CrackForest" dataset [31], the CSSC database [36] and SDNET2018 [26]. However, as all of the former works feature a single-target and in fact single-class task, we have decided to extend existing work with the multi-class multi-target CODEBRIM dataset.

Defect (crack) recognition. Koch *et al*. [20] provide a comprehensive review on the state of computer vision in concrete defect detection and open aspects. In summary, the majority of approaches follow task specific modelling. Data-driven applications are still the exception and are yet to be leveraged fully. Recent works [23, 8, 18] show application to crack versus non-crack classification using images with little clutter and lack of structural context. An additional defect class of spalling is considered by the authors of [36]. Similar to other works, they focus on the single-target scenario and evaluation of well-known CNN baselines from prior object recognition literature. We extend their work by meta-learning more task specific neural architectures for more defect categories and overlapping defects.

Convolutional neural networks. A broader review of deep learning, its history and neural architecture innovations is given by LeCun *et al*. [22]. We recall some CNN architectures that serve as baselines and give a frame of reference for architectures produced by meta-learning on our task. Alexnet [21] had a large success on the ImageNet [29] challenge that was later followed by a set of deeper architectures commonly referred to as VGG [32]. Texture-CNN [1] is an adapted version of the Alexnet design that includes an energy-based adaptive feature pooling and FV-CNN [7] augments VGG with Fisher Vector pooling for texture classification. Recent works address information flow in deeper networks by adding skip connections with residual networks [14], wide residual networks (WRN) [37] and densely connected networks (DenseNet) [16].

Meta-learning neural architectures. Although deep neural networks empirically work well in many practical applications, networks have initially been designed for different tasks. A recent trend to bypass the human design intuition is to treat neural architectures themselves from a meta-learning perspective and conduct a black-box optimization on top of the training of weights to find suitable task-specific architecture designs. Several works in the literature have posed architecture meta-learning from a variety of perspectives based on reinforcement learning (RL) controllers [2, 38, 27, 4], differentiable methods [24] or evolutionary strategies [28]. In our work, we evaluate and adapt two RL based approaches to multi-target defect classification: MetaQNN [2] and "efficient neural architecture search" (ENAS) [27]. We pick these two approaches as they share underlying principles of training RL controllers. This allows us to pick a common reward metric determined by proposed CNN candidate accuracies. The main differences lie in the RL agents' nature: MetaQNN employs Q-Learning to learn to suggest increasingly accurate CNNs, whereas ENAS uses policy gradients [34] to train an auto-regressive recurrent neural network that samples individual layers based on previous input.

(a) Top row from left to right: 1.) exposed bars, spallation, cracks (hard to see) 2.) hairline crack with efflorescence 3.) efflorescence 4.) defect-free concrete. Bottom row from left to right: 1.) large spalled area with exposed bars and corrosion 2.) crack with graffiti 3.) corrosion stain, minor onset efflorescence 4.) defect-free concrete with dirt and markings.

(b) From left to right: 1.) spalled area with exposed bar, advanced corrosion and efflorescence 2.) exposed corroded bar 3.) larger crack 4.) partially exposed corroded bars, cracks 5.) hairline crack 6.) heavy spallation, exposed bars, corrosion 7.) wet/damp crack with efflorescence on the top 8.) efflorescence 9.) spalled area 10.) hairline crack with efflorescence.

Figure 1: Dataset examples. Top figure: full high-resolution images. Images heavily down-sampled for view in pdf. Bottom figure: Image patches cropped from annotated bounding boxes (not corresponding to top images). Images resized for view in pdf but with original aspect ratio.

3. The CODEBRIM dataset

The acquisition of the COncrete DEfect BRidge IMage: CODEBRIM dataset was driven by the need for a more diverse set of the often overlapping defect classes in contrast to previous crack focused work [31, 36, 26]. In particular, deep learning application to a real-world inspection scenario requires sampling of real-world context due to the many factors of variation in visual defect appearance. Our dataset is composed of five common defect categories: crack, spallation, exposed reinforcement bar, efflorescence (calcium leaching), corrosion (stains), found in 30 unique bridges (disregarding bridges that did not have defects). The bridges were chosen according to varying overall deterioration, defect extent, severity and surface appearance (e.g. roughness and color). Images were taken under changing weather conditions to include wet/stained surfaces with multiple cameras at varying scales. As most defects tend to be very small one crucial requirement was the acquisition at high-resolution. Considering that large parts of bridges are not accessible for a human, a subset of our dataset was acquired by UAV. We continue with the requirements and rationale behind the camera choices, the annotation process that led to the dataset and finally give a summary of important dataset properties.

3.1. Image acquisition and camera choice

Image acquisition and camera choices were motivated by typical concrete cracks in bridges having widths as small as $0.3\,\mathrm{mm}$ [20]. Resolving such defects on a pixel level

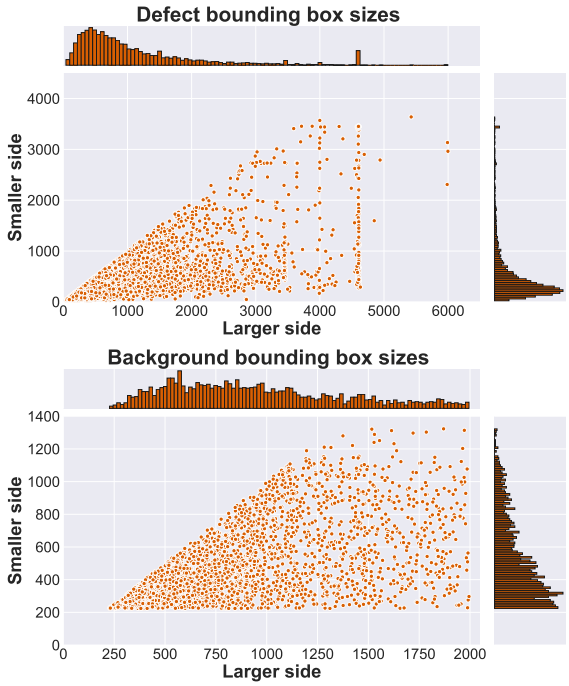


Figure 2: Top panel: distribution of annotated bounding box sizes for defects. Bottom panel: distribution of sizes for sampled non-overlapping background bounding boxes.

imposes a strong constraint on the distance and resolution at which the images are acquired. In a naive calculation for a conventional consumer-grade camera with an example chip of size $23.50 \times 15.60\,\mathrm{mm}$ and maximum resolution 6000×4000, this translates to around $0.1\,\mathrm{mm}$ per pixel at a focal length of $50\,\mathrm{mm}$ and a distance of roughly $1.5\,\mathrm{m}$ (assuming a pinhole camera model and viewing axis perpendicular to the surface). Based on this requirement our dataset was gathered with four different cameras at high resolution and large focal lengths under varying distance and angles. In addition, to homogeneously illuminate the darker bridge areas, we made use of diffused flash. Exact camera models and corresponding detailed parameters can be found in the supplementary material.

3.2. Dataset properties

We employed a multi-stage annotation process by first curating acquired images, annotating bounding boxes per defect and sequentially labelling each class separately. The rationale and exact annotation process is outlined in the supplementary material. The acquisition and annotation process resulted in a dataset with the following properties:

- 1590 high-resolution images with defects in context of 30 unique bridges, acquired at different scales and resolutions.
- 5354 annotated defect bounding boxes (largely with overlapping defects) and 2506 generated non-overlapping background bounding boxes.
- Defect numbers for the following classes: crack - 2507, spallation - 1898, efflorescence - 833, exposed bars - 1507 and corrosion stain - 1559.

Examples of images and extracted patches from bounding boxes featuring a variety of overlapping and non-overlapping defects can be seen in figure 1a and 1b respectively. We point out that in contrast to most object and texture based benchmarks, the majority of our dataset has more than one class occurring at once. We show a corresponding histogram for the number of defect classes per individual bounding box annotation in the supplementary material.

Apart from the multi-target nature making our dataset more challenging than single-class recognition, the task is difficult because of large variations in aspect ratio, scale and resolution of the different defects and their bounding boxes. This is true especially at a scene level, considering that cracks can be very fine and elongated, whereas spalled areas can vary almost arbitrarily. To illustrate these variations we visualize the distributions of defect bounding box sizes and the sampled background bounding box sizes in figure 2. Further details about distributions of image sizes, bounding box size distributions per category (with overlaps due to the multi-target nature) and distribution of aspect ratios per defect can be found in the supplementary material.

4. Meta-learning convolutional neural networks for multi-target defect classification

We use meta-learning to discover models tailored to multi-target defect classification on the CODEBRIM dataset. In order to find a suitable set of hyper-parameters for the meta-learning search space and training of neural architectures we start with the T-CNN [1] and VGG-A [32] baselines and investigate the influence of learning rate, batch size and patch size. For this we separate the dataset into train and validation splits and set aside a final test set for evaluation. We then adapt the MetaQNN [2] and ENAS [27] architecture meta-learning approaches and contrast the obtained results with the following set of CNN architectures proposed in the literature: Alexnet [21], T-CNN [1], VGG-A and VGG-D [32], wide residual network (WRN) [37] and densely connected convolutional networks (DenseNet) [16]. We want to point out that even though bounding box annotations are present in our dataset, we do not evaluate any bounding box detection algorithms because our goal at this stage is the establishment of the already challenging multi-target classification task. We have also evaluated

Architecture	Batch size	Multi-target accuracy [%] depending on learning rate schedule: max to min								
		$[10^{-1}, 10^{-5}]$			$[5 \cdot 10^{-2}, 5 \cdot 10^{-4}]$			$[10^{-2}, 10^{-5}]$		
		best val	bv-test	bv-train	best val	bv-test	bv-train	best val	bv-test	bv-train
T-CNN	16	64.62	69.51	80.27	63.67	65.71	83.38	64.30	67.93	93.91
	32	64.78	66.19	87.66	63.36	68.72	94.49	62.84	66.35	96.22
	64	63.36	70.14	95.21	63.52	67.93	98.10	62.26	66.82	95.85
	128	63.67	67.45	98.31	63.36	66.82	98.63	60.53	65.08	94.47
VGG-A	16	60.22	62.08	75.74	63.67	68.24	94.78	64.93	70.45	98.29
	32	63.05	67.77	93.88	63.05	66.35	94.27	65.40	69.51	97.01
	64	63.36	69.66	98.00	63.37	70.45	90.64	59.90	63.82	97.01
	128	63.20	61.29	92.99	63.52	68.07	98.55	58.80	61.29	92.99

Table 1: Grid-search conducted on different batch sizes and different learning rate schedules for the T-CNN and VGG-A models. The multi-target best validation accuracy (best val) is shown together with each model's accuracy on the test set at the point in time of achieving the best validation accuracy (bv-test). The analogous training accuracy (bv-train) is shown to demonstrate that models do not under-fit. These validation accuracies have been used to determine training hyper-parameters.

transfer-learning from the ImageNet and MINC datasets, albeit without improvements and therefore report these experiments in the supplementary material.

4.1. Dataset training, validation and test splits

We have randomly chosen 150 unique defect examples per class for validation and test sets respectively. To avoid over-fitting due to very similar context, we make sure that we always include all annotated bounding boxes from one image in one part of the dataset split only. An alternative way to split the dataset is to separate train, validation and test sets according to unique bridges. However, it is infeasible to balance such a split with respect to equal amount of occurrences per defect due to individual bridges not featuring defect classes uniformly (particularly with class overlaps) and thus makes an unbiased training and reporting of average losses or accuracies difficult. Nevertheless, to investigate the importance of over-fitting global properties, we investigate and further discuss the challenges of such splits in the supplementary material.

4.2. Training procedure

The challenging multi-class multi-target nature of our dataset makes the following measures necessary:

1. **Multi-class multi-target.** For a precise estimate of a model's performance in a multi-target scenario, a classification is considered as correct if, and only if, all the targets are predicted correctly. To adapt all neural networks for this scenario we use a Sigmoid function for every class in conjunction with the binary cross entropy loss function. When we calculate classification accuracies we binarize the Sigmoid output with a threshold of 0.5. Note that this could be treated as a hyper-parameter to potentially obtain better results.

2. **Variations in scale and resolution.** We address the variation in scale and resolution of bounding boxes by following the common literature approach based on previous datasets such as ImageNet [29] and the models presented in [21, 32, 37, 16]. Here, the smaller side of the extracted patch is rescaled to a pre-determined patch size and random quadratic crops of patch size are taken to extract fixed size images during training.

3. **Train set imbalance.** We balance the training dataset by virtually replicating the under-represented class examples such that the overall defect number per class is on the same scale to make sure defect classes are sampled equally during training. Note that test and validation sets are balanced by design.

The reason for adopting step two is to allow for a direct comparison with CNNs proposed in prior literature without making modifications to their architectures. We do not use individual class accuracies as a performance metric as it is difficult to compare models that don't capture overlaps adequately. Nevertheless we provide an example table with multi-target versus per-class accuracy of later shown CNN literature baselines in the supplementary material.

4.2.1 Common hyper-parameters

We conduct an initial grid-search to find a suitable common set of hyper-parameters for CNNs (meta-learned or not) trained with stochastic gradient descent based on the T-CNN [1] and VGG-A [32] architectures. For this we use learning rate schedules with warm restarts (SGDWR) according to the work of [25]. The grid search features three cycles with ranges inspired by previous work [25, 27]: $[10^{-1}, 10^{-5}]$, $[5 \cdot 10^{-2}, 5 \cdot 10^{-4}]$ and $[10^{-2}, 10^{-5}]$,

a warm restart cycle length of 10 epochs that is doubled after every restart, and four different batch sizes: $128, 64, 32$ and 16. All networks are trained for four warm restart cycles and thus 150 overall epochs after which we have noticed convergence. Other hyper-parameters are a momentum value of 0.9, a batch-normalization [17] value of 10^{-4} to accelerate training and a dropout rate [33] of 0.5 in the penultimate classification layer. Weights are initialized according to the Kaiming-normal distribution [13].

We determine a suitable set of hyper-parameters using cross-validation, that is according to the best validation accuracy during the entire training. We then report the test accuracy based on this model. We show the multi-target accuracy's dependency on learning rate and batch size for the two CNN architectures in table 1. We notice that the general trend is in favor of lower batch sizes and a learning rate schedule in the range of $\left[10^{-2}, 10^{-5}\right]$. While the evaluated best validation model's test accuracy generally follows a similar trend, the best test accuracies aren't always correlated with a higher validation accuracy, showing a light distribution mismatch between the splits. We further note that the absolute best test accuracy doesn't necessarily coincide with the point of training at which the model achieves the best validation accuracy. In general, the models seem to have a marginally higher accuracy for the test split. The table also shows that validation and test sets are reasonably different from the train set, on which all investigated models achieve an over-fit.

After determining a suitable set of hyper-parameters, a batch size of 16 and a learning rate cycle between $\left[10^{-2}, 10^{-5}\right]$, we have proceeded with the selection of patch sizes determined through an additional experiment based on best multi-target validation accuracy. We again emphasize that we do not pick hyper-parameters based on test accuracy, even if a model with lower validation accuracy has a better test score.

4.2.2 Selection of patch size

Whereas most CNN architectures proposed in the literature are designed for patch sizes of 224×224, we also evaluate a range of different patch sizes by modifying the number of parameters in the T-CNN model's first fully-connected layer according to the last convolution's spatial output resolution (we do not modify the outgoing feature amounts). In figure 3 we show the multi-target best validation and corresponding test accuracies for different patch and batch sizes. The perceivable trend is that models trained on patch sizes smaller than 224 yield less accuracy, whereas the validation accuracy seems to plateau or feature an upwards trend for larger patch sizes. The corresponding test accuracies mirror this trend. We leave the evaluation of even larger patch sizes for future work. For the remainder of this work, we continue

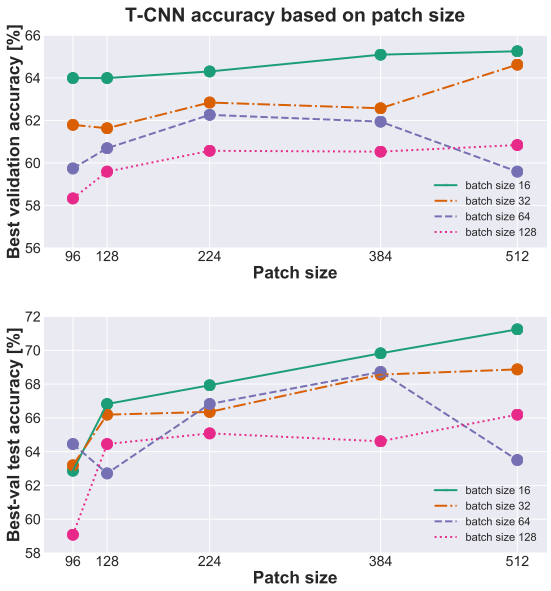


Figure 3: T-CNN multi-target validation accuracy (top panel) and best validation model's multi-target test accuracy (bottom panel) in dependence on patch size.

to use a patch size of 224. Although larger patch sizes seem promising they prevent a direct comparison and contrasting of meta-learning approaches with neural network models proposed in the literature without making modifications to their architectures.

4.2.3 Meta-learning specific parameters

We design the reward for both MetaQNN and ENAS to fit our multi-target scenario by setting it to the multi-target validation accuracy. We re-iterate that using a per-class accuracy as a metric and particularly to design an RL reward, could lead to controllers being biased towards naively raising the reward by generating models that predict (the easiest) subsets of classes correctly without considering the multi-target overlap properly. We try to set the method specific hyper-parameters of the two meta-learning methods as similar as possible to allow for a direct comparison. We therefore train all child CNN models using the SGDWR schedules and SGD hyper-parameters specified earlier.

MetaQNN: We employ an ϵ-greedy schedule for the Q-learning approach. We train an overall amount of 200 architectures and start with a full exploration phase of 100 architectures for $\epsilon = 1.0$. We continue with 10 architectures for ϵ values of 0.9 to 0.3 in steps of 0.1 and finish with 15 architectures for ϵ values of 0.2 and 0.1. Our search space

is designed to allow neural architectures with at least 3 and a maximum of 10 convolutional layers. We include choices for quadratic filters in the sizes of $3, 5, 7, 9, 11$ with possible number of features per layer of $32, 64, 128, 256$. We use a Q-learning rate of 0.1, a discount-factor of 1.0 and an initial Q-value of 0.15. The latter is motivated by a 15% validation accuracy early-stopping criterion at the end of the first SGDWR cycle. In analogy to [2], if an architecture doesn't pass this threshold, it is discarded and a new one is sampled and trained.

Apart from the different reward design, we also make several extensions to the MetaQNN [2]: We cover downsampling with an option for convolution stride $s = 2$ for filter sizes larger than 5. Convolutional layers are further followed by an adaptive pooling stage using spatial-pyramidal pooling (SPP) [12] of allowed scales $3, 4, 5$ and the possibility to pick a hidden fully-connected layer with size $32, 64$ or 128 before adding the final classification stage. All layers are followed by batch-normalization and a ReLU nonlinearity to accelerate training. We also include the possibility to add ResNet-like skip connections between two padded 3×3 convolutions that do not change spatial dimensionality. If the number of convolutional output features is the same the skip connection is a simple addition, whereas an extra parallel convolution (that isn't counted as an additional layer) is added if the amount of output features needs to change. We make these extensions to provide a fairer comparison to the architecture search of ENAS, that by design contains batch-normalization, adaptive pooling and the possibility of adding skip-connections.

ENAS: In contrast to MetaQNN where the number of layers of each architecture is flexible, network depth in ENAS is pre-determined by the specification of number of nodes in the directed acyclic graph (DAG). Each node defines a possible set of feature operations that the RNN controller samples at each step together with connection patterns. In the process of the search, the same DAG is used to generate architectures with candidates sharing weights through sharing of feature operations. We choose to let the search evolve through alternate training of the CNNs' shared weights on the CODEBRIM train set and the RNN controller's weights on the validation set, where the controller samples one architecture per mini-batch. We design the DAG such that each architecture has 7 convolutional layers and 1 classification layer that is followed by a Sigmoid function. We choose this depth to have a direct comparison to the average depth of MetaQNN architectures. The allowed feature operations are convolutions with square filters of size 3 and 5, corresponding depth-wise separable convolutions [6], max-pooling and average-pooling with kernel size 3×3. Each layer is followed by batch-normalization and a ReLU nonlinearity. Because ENAS uses shared weights in the search,

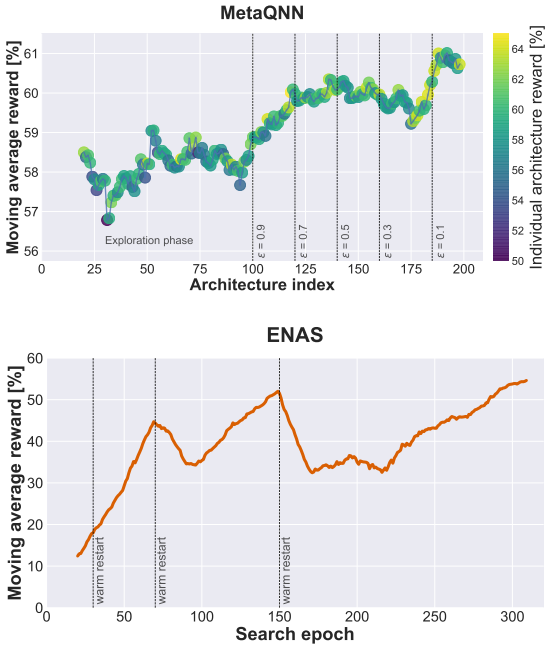


Figure 4: Evolution of the moving average reward defined as the multi-target validation accuracy of architectures proposed through meta-learning. The top panel additionally shows individual architecture accuracies for the MetaQNN in color. ENAS in the bottom panel has shared model weights during training and thus requires a final end-to-end re-training step for final validation accuracies of individual architectures.

a final re-training step of proposed architectures is necessary. We use a feature amount of 64 during the search for all layers and use a DenseNet growth-pattern [16] of $k = 2$ in the final training consistent with the work of Pham *et al.* [27]. The total number of search epochs is 310 (5 SGDWR cycles) after which we have experienced convergence of the controller. The RNN controller consists of an LSTM [15] with two hidden-layers of 64 features that is trained with a learning rate of 10^{-3} using ADAM [19].

4.3. Results and discussion

We demonstrate the effectiveness of neural architecture search with MetaQNN and ENAS for multi-target concrete defect classification on the CODEBRIM dataset. We show respective moving average rewards based on a window size of 20 architectures in figure 4. Individual architecture accuracies for MetaQNN are shown in color for each step in the top panel. We observe that after the initial exploration phase, the Q-learner starts to exploit and architec-

Architecture	Multi-target accuracy [%]		Params [M]	Layers
	best val	bv-test		
Alexnet	63.05	66.98	57.02	8
T-CNN	64.30	67.93	58.60	8
VGG-A	64.93	70.45	128.79	11
VGG-D	64.00	70.61	134.28	16
WRN-28-4	52.51	57.19	5.84	28
Densenet-121	65.56	70.77	11.50	121
ENAS-1	65.47	70.78	3.41	8
ENAS-2	64.53	68.91	2.71	8
ENAS-3	64.38	68.75	1.70	8
MetaQNN-1	66.02	68.56	4.53	6
MetaQNN-2	65.20	67.45	1.22	8
MetaQNN-3	64.93	72.19	2.88	7

Table 2: Comparison of popular CNNs from the literature with the top three architectures of MetaQNN and ENAS in terms of best multi-target validation accuracy (best val), best validation model's test accuracies (bv-test), overall amount of parameters (Params) in million and amount of trainable layers. For WRN we use a width factor of 4 and a growth rate of $k = 32$ for DenseNet.

tures improve in multi-target validation accuracy. In the bottom panel of the figure we show corresponding rewards for the shared-weight ENAS DAG. We observe that both methods learn to suggest architectures with improved accuracy over time. We remind the reader that in contrast to the MetaQNN, a final re-training step of the top architectures is needed for ENAS to obtain the task's final accuracy values.

The multi-target validation and test accuracies, again reported at the point in time of best validation, the number of overall architecture parameters and layers for the top three MetaQNN and ENAS architectures can be found in table 2. We also evaluate and provide these values for popular CNN baselines: Alexnet [21], VGG [32], Texture-CNN [1], wide residual networks (WRN) [37] and densely connected networks (DenseNet) [16]. We see that the Texture-CNN variant of Alexnet slightly outperforms the latter. The connectivity pattern of the DenseNet architecture also boosts the performance in contrast to the VGG models. Lastly, we note that we were only able to achieve heavy over-fitting with WRN configurations (even with other hyper-parameters and other configurations such as WRN-28-10 or WRN-40).

The accuracies obtained by all of our meta-learned architectures, independently of the underlying algorithm, outperform most baseline CNNs and feature at least similar performance in comparison to DenseNet. Moreover, they feature much fewer parameters with fewer overall layers and are thus more efficient than their computationally heavy counterparts. Our best meta-learned models have validation accuracies as high as 66%, while the test accuracies go up to 72% with total amount of parameters less than 5 million. In contrast to literature CNN baselines these architectures are thus more tailored to our specific task and its multi-target nature. Interestingly, previously obtained improvements from one literature CNN baseline to another on ImageNet, such as Alexnet 81.8% to VGG-D 92.8% top-5 accuracies, do not show similar improvements when evaluated on our task. This underlines the need for diverse datasets in evaluation of architectural advances and demonstrates how architectures that were hand-designed, even with incredible care and effort, for one dataset such as ImageNet may nonetheless be inferior to meta-learned neural networks.

Between the two search strategies we do not observe a significant difference in performances. We believe this is due to previously mentioned modifications to MetaQNN, mainly the addition of skip-connections and batch-normalization that make proposed architectures more similar to those of ENAS. We point the reader to the supplementary material for exact definitions of meta-learned architectures. There, we also include a set of image patches that are commonly classified as correct for all targets, images where only part of the overlapping defect classes is predicted and completely misclassified examples.

5. Conclusion

We introduce a novel multi-class multi-target dataset called CODEBRIM for the task of concrete defect recognition. In contrast to previous work that focuses largely on cracks, we classify five commonly occurring and structurally relevant defects through deep learning. Instead of limiting our evaluation to common CNN models from the literature, we adapt and compare two recent meta-learning approaches to identify suitable task-specific neural architectures. Through extension of the MetaQNN, we observe that the two meta-learning techniques yield comparable architectures. We show that these architectures feature fewer parameters, fewer layers and are more accurate than their human designed counterparts on our presented multi-target classification task. Our best meta-learned models achieve multi-target test accuracies as high as 72%. Our work creates prospects for future work such as multi-class multi-target concrete defect detection, semantic segmentation and system applications like UAV based real-time inspection of concrete structures.

Acknowledgements: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 687384 "AEROBI". We would like to thank everyone involved in the AEROBI project. Particular appreciation is given for the civil engineering team of Egnatia Odos A.E. and Netivei NTIC, without whom the annotation of the dataset wouldn't have been possible. We further thank FADA-CATEC, Tobias Weis and Sumit Pai for their support in parts of the data acquisition and Hieu Pham for valuable discussion of ENAS hyper-parameters.

References

[1] Vincent Andrearczyk and Paul F Whelan. Using filter banks in Convolutional Neural Networks for texture classification. *Pattern Recognition Letters*, 84:63–69, 2016. 1, 2, 4, 5, 8

[2] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing Neural Network Architectures using Reinforcement Learning. *International Conference on Learning Representations (ICLR)*, 2016. 2, 4, 7

[3] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the Materials in Context Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[4] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient Architecture Search by Network Transformation. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[5] Barbara Caputo, Eric Hayman, and P Mallikarjuna. Class-specific material categorisation. In *International Conference on Computer Vision (ICCV)*, 2005. 2

[6] Francois Chollet. Xception: Deep Learning with Depthwise Seperable Convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017. 7

[7] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep convolutional filter banks for texture recognition and segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[8] Wilson R. L. da Silva and Diogo S. de Lucena. Concrete Cracks Detection Based on Deep Learning Image Classification. In *International Conference on Experimental Mechanics (ICEM18)*, 2018. 1, 2

[9] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34, 1999. 2

[10] Mark Everingham, S. M.Ali Ali Eslami, Luc Van Gool, Christopher K.I. I Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2014. 1, 2

[11] Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh. On the Significance of Real-World Conditions for Material Classification. In *European Conference on Computer Vision (ECCV)*, 2004. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer Vision (ECCV)*, pages 346–361, 2014. 7

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 6

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. 7

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 1, 2, 4, 5, 7, 8

[17] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, volume 37, pages 448–456, 2015. 6

[18] Hyunjun Kim, Eunjong Ahn, Myoungsu Shin, and Sung-Han Sim. Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning. *Structural Health Monitoring*, 2018. 1, 2

[19] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 7

[20] Christian Koch, Kristina Georgieva, Varun Kasireddy, Burcu Akinci, and Paul Fieguth. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, 29(2):196–210, 2015. 2, 3

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems (NeurIPS)*, volume 25, pages 1097–1105, 2012. 1, 2, 4, 5, 8

[22] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 2

[23] Yundong Li, Hongguang Li, and Hongren Wang. Pixel-wise crack detection using deep local pattern predictor for robot application. *Sensors*, 18(9), 2018. 1, 2

[24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search. *International Conference on Learning Representations (ICLR)*, 2019. 2

[25] I. Loshchilov and F. Hutter. SGDR: Stochastic Gradient Descent With Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[26] Marc Macquire, Sattar Dorafshan, and Robert J. Thomas. SDNET2018: A concrete crack image dataset for machine learning applications. *https://digitalcommons.usu.edu/all_datasets/48 (last access: 06.11.18)*, Paper 48, 2018. 1, 2, 3

[27] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient Neural Architecture Search via Parameters Sharing. In *International Conference on Machine Learning (ICML)*, 2018. 2, 4, 5, 7

[28] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Quoc Le, and Alex Kurakin. Large-Scale Evolution of Image Classifiers. In *International Conference on Machine Learning (ICML)*, 2017. 2

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. In *International Journal of Computer Vision (IJCV)*, volume 115, pages 211–252, 2015. 1, 2, 5

[30] Lavanya Sharan, Ruth Rosenholtz, and Edward H. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision (JOV)*, 9(8), 2009. 2

[31] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 17(12):3434–3445, 2016. 1, 2, 3

[32] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 4, 5, 8

[33] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMRL)*, 15:1929–1958, 2014. 6

[34] S. Richard Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems (NeurIPS)*, pages 1057–1063, 1999. 2

[35] Jianxiong Xiao, James Hays, Krista A Ehinger, and Antonio Torralba. SUN Database : Large-scale Scene Recognition from Abbey to Zoo. In *Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2

[36] Liang Yang, Bing Li, Wei Li, Zhaoming Liu, Guoyong Yang, and Jizhong Xiao. Deep Concrete Inspection Using Unmanned Aerial Vehicle Towards CSSC Database. In *International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1, 2, 3

[37] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *British Machine Vision Conference (BMVC)*, pages 87.1–87.12, 2016. 1, 2, 4, 5, 8

[38] Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, 2017. 2

Meta-learning Convolutional Neural Architectures for Multi-target Concrete Defect Classification with the COncrete DEfect BRidge IMage Dataset: Supplementary Material

Martin Mundt[1*], Sagnik Majumder[1], Sreenivas Murali[1*], Panagiotis Panetsos[2], Visvanathan Ramesh[1*]
1. Goethe University 2. Egnatia Odos A. E.
{mmundt, vramesh}@em.uni-frankfurt.de {majumder, murali}@ccc.cs.uni-frankfurt.de
ppane@egnatia.gr

A. Content overview

The supplementary material contains further details for material presented in the main body.

We start with an extended description of the CODEBRIM dataset in section B. Here, we provide the specific settings for the employed cameras for dataset image acquisition. In addition to the histogram presented in the main body that shows number of different defects per bounding box, we further provide a histogram with amount of bounding box annotations per image. Additional material reveals specifics of the main body's figure depicting the large variations in distribution of bounding boxes by illustrating the individual nuances of this distribution per defect class. The supplementary dataset material is concluded with a brief discussion on background patch generation.

In supplementary section C we provide a brief discussion on why multi-target accuracy is a better reward metric than naive single-class accuracies and show what multi-target accuracies would translate to in terms of a naive average single-class accuracy. We give detailed descriptions and graphs of the six meta-learned architectures for the top three models obtained through MetaQNN and ENAS. Although it isn't an immediate extension to the main body, but rather additional content, we provide a compact section on transfer learning with experiments conducted with models pre-trained on the ImageNet and MINC datasets. We have decided to move these experiments to the the supplementary material for the interested reader as they do not show any improvements over the content presented in the main body. We conclude the supplementary material with examples for images that are commonly classified correctly as well as showing some typical false multi-target classifications to give the reader a better qualitative understanding of the dataset complexity and challenges.

* work conducted while at Frankfurt Institute for Advanced Studies

B. CODEBRIM dataset

B.1. Delamination as a defect class

Some of the CODEBRIM dataset features images that have a defect that is typically referred to as delimation. It is a stage where areas start to detach from the surface. Delamination can thus be recognized by a depth offset of a layer from the main surface body. However, in images acquired by a single camera, especially if the images were acquired using a camera view direction that is orthogonal to the surface, these boundaries are often visually not distinguishable from cracks. Without further information, even a civil engineering expert faces major difficulty in such a distinction between these categories. We have thus decided to label eventual occurrences of delamination together with the crack category.

B.2. Cameras

We show the four cameras used for acquisition of dataset images in table 1. All chosen cameras have a resolution above Full-HD, with the highest resolution going up to 6000×4000 pixels. For two cameras we have used a lens with varying focal length, whereas two cameras had a lens with fixed focal length of 50 and $55\,\mathrm{mm}$ respectively. We have further systematically varied aperture in conjunction with the use of diffused flash modules to homogeneously illuminate dark bridge areas, while also adjusting for changing global illumination (avoiding heavy over or under-exposure). A different very crucial aspect was the employed exposure time. Pictures acquired by UAV were generally captured with a much shorter exposure time to avoid blurring of the image due to out of focus acquisition or inherent vibration and movement of the UAV. One of our cameras, Sony α-6000 has thus exclusively been used in the context of UAV based image acquisition with an exposure time of $1/1000$ seconds.

We show how the CODEBRIM dataset is practically

Camera	Resolution [pixels]	Exposure [s]	f [mm]	F-value [f/]	ISO	Flash
Canon IXUS 870 IS	2592×1944	flexible	$5 - 20$	$2.8 - 5.8$	$100 - 800$	none
Panasonic DMC-FZ72	4608×3456	$1/250$	$4 - 42$	5.6	400	built-in
Nikon D5200	6000×4000	$1/200$	55	11.0	200	built-in
Sony α-6000	6000×3376	$1/1000$	50	$2.0 - 5.6$	$50 - 400$	HVL-F43M

Table 1: Description of cameras, including resolution, exposure time in fraction of a second, focal length f in mm, the aperture or F-value in terms of focal length, ISO speed rating and information on potentially used flash.

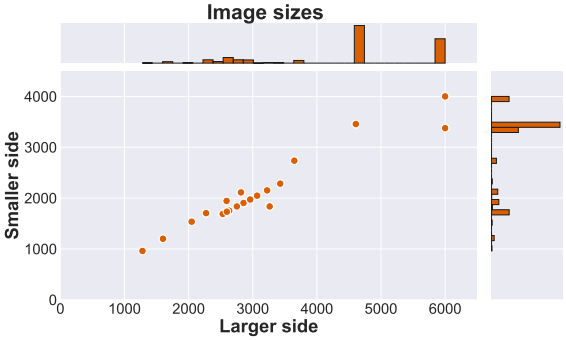


Figure 1: Distribution of image resolutions. Smaller and larger side refer to the image's larger and smaller dimension.

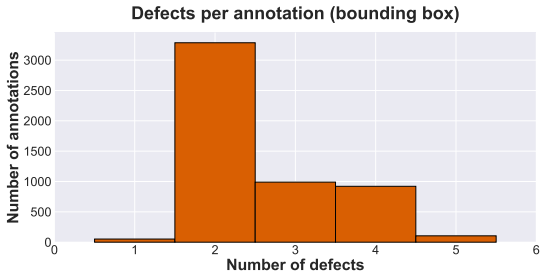


Figure 2: Histogram of number of simultaneously occurring defect classes per annotated bounding box.

comprised of the varying resolutions resulting from use with different cameras and settings in figure 1. We can observe that the aspect ratio is almost constant with changes in absolute resolution and that the large majority of images has been acquired at very high resolutions.

B.3. Annotation process

After curating acquired images by excluding the majority of images that do not have defects, we employed a multi-stage annotation process to create a multi-class multi-target classification dataset using the annotation tool LabelImg [2]

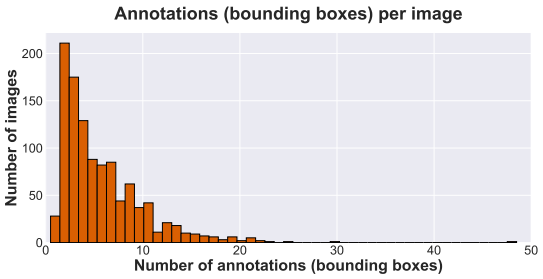


Figure 3: Distribution of number of bounding box annotations per image.

in consultation with civil engineering experts:

1. We first annotated bounding boxes for areas containing defects in the Pascal format [1].

2. Each individual bounding box was analyzed with respect to one defect class and a corresponding label was set if the defect is present.

3. After finishing the entire set of bounding boxes for one class, we repeated step 2 for the remaining classes and arrived at a multi-class multi-target annotation.

4. In the last stage, we sampled bounding boxes containing background (concrete without defects as well as non-concrete) according to absolute count, aspect ratios and size of annotated defect bounding boxes.

The reason for staging the process is that we found the annotation process to be less error prone if annotators had to concentrate on the presence of one defect at a time.

B.4. Further dataset statistics

We show additional information and statistics of the dataset. In figure 2 we show a histogram that demonstrates how one bounding box annotation typically contains more than one defect class at a time. In figure 3 the complementary histogram of the number of annotated bounding boxes per image can be found. Here, we can further observe

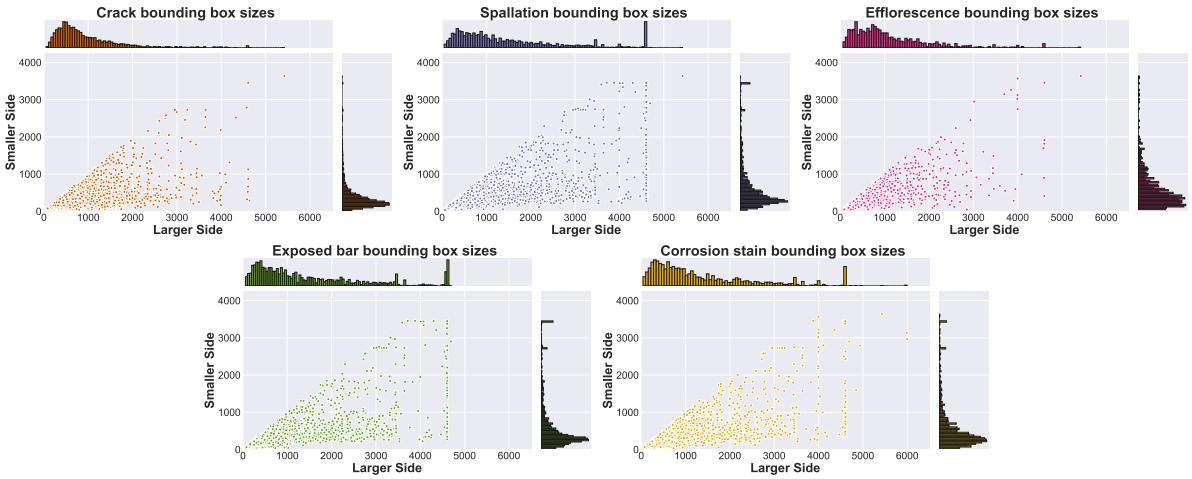


Figure 4: Individual distributions of annotated bounding box sizes for each of the 5 defect classes.

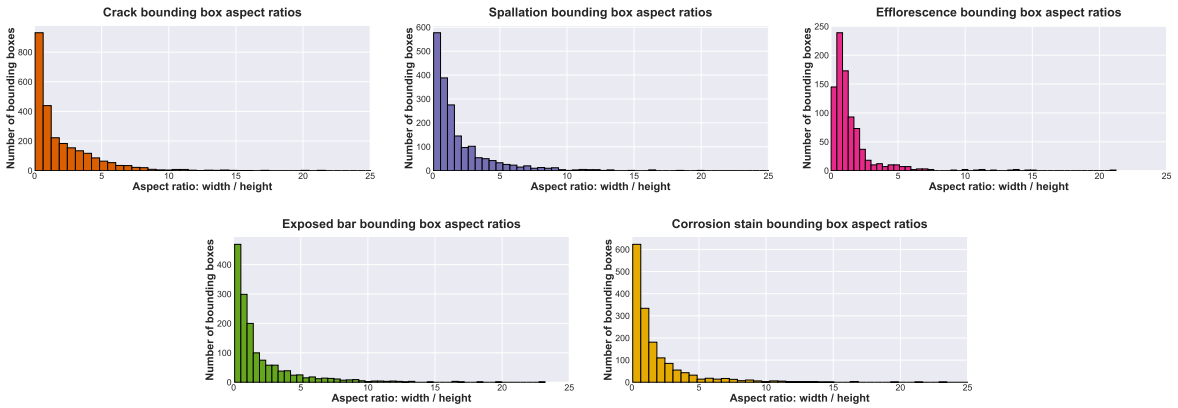


Figure 5: Individual distributions of number of bounding box annotations for different aspect ratios for each of the 5 defect classes.

that our choice of bridges led to image acquisition scenarios where one acquired image generally contains multiple different defect locations. While this is not impacting our classification task, we believe it is a crucial precursor for future extensions to a realistic semantic segmentation scenario.

In addition to the distribution of the annotated bounding box sizes for background and for all the defects combined as shown in the main body, the reader might be interested in the specific distribution per defect class. In figure 4 the corresponding distribution of annotated bounding boxes per-class is shown. Similarly, figure 5 contrasts the aspect ratio distributions for the individual defects. It is to be noted that these per-class distributions are not mutually exclusive because of multi-target overlap in the bounding box annotations. All individual classes have a similarly distributed bounding box size per defect including a long tail towards large resolutions. A major difference for individual classes can be found at high resolutions between the crack and efflorescence classes and the spallation, exposed bar and corrosion stain classes. The latter sometimes span an entire image. While this of course depends on the acquisition distance, we point out that in images acquired at a similar distance spalled and corroded areas including bar exposition are larger on average.

B.5. Random generation of background bounding boxes

We emphasize that the CODEBRIM dataset has many factors that add to the complexity. Acquired images have

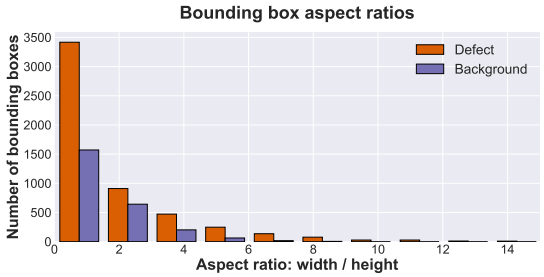


Figure 6: Distribution of number of defect and background bounding box annotations for different aspect ratios.

large variations depending on the target geometry, types of defects and their overlapping behavior, the camera pose relative to the photographed surface (particularly if captured by UAV), as well as global scene properties such as illumination. From a machine learning classification dataset point of view it is thus interesting to capture this complexity in the generation of image patches for the background class.

We therefore devote this supplementary section to provide further details to the reader on generation of background bounding boxes. Before administering the final dataset, the last dataset creation step of sampling areas containing background has been validated. Specifically, we have checked whether the distribution of sizes (shown in the main body) as well as the distribution of sampled areas' aspect ratios approximately follow those of the human annotated defects. In figure 6 we show the aspect ratios for the annotated defects together with the sampled bounding box aspect ratios for background. Whereas the overall count for background is less than the integrated total amount of defects (number of background samples roughly corresponds to occurrence of each individual defect class), the distribution of aspect ratios is confirmed to have the same trend. We have further made sure that none of the background bounding boxes have any overlap with bounding boxes annotated for defects and that bounding boxes for background are evenly distributed among images. In summary, this methodology captures the complexity of surface variations, target geometry, global illumination and makes sure that image patch resolution and sizes reflect those of defect annotations.

C. Deep convolutional neural networks for multi-target defect classification

C.1. Per-class and multi-target CNN accuracies

As mentioned in the main body of this work, most image classification tasks focus on the single target scenario and an easy pitfall would be to treat our task in a similar fashion. This would imply reporting classification accuracies independently per class and not treating the task in the multi-target fashion. We remind the reader that this would not represent the real-world scenario appropriately, where one is interested in the severity of the degradation of the inspected concrete structure. This severity is magnified if two or more different defect classes are mutually occurring and overlapping. Nevertheless, one idea could be to design the reward for the meta-learning algorithms based on such individual class accuracies or the corresponding average. We report the validation accuracy per class (background and five defects) and their respective average for the CNN literature baselines, together with the multi-target best validation accuracy and the corresponding test accuracy in table 2. Note that we do this only for the sake of completeness as this thought could occur to other researchers and to show researchers the relationship between accuracy values. Initially, a multi-target accuracy of 65% might not look like a large value, but it practically translates to around 90% classification accuracy had each class been treated independently in our task. Apart from the above stated obvious argument of resemblance to real-world application, the table further indicates why the multi-target accuracy is a better metric to employ in meta-learning reward design. Although each of the baseline models learns to recognize individual defects in the image with high precision, they are not equally competent at recognizing all the defects together in the multi-target scenario. The individual class accuracies do not have clear trends as they fluctuate individually, are difficult to interpret from one model to the next and do not intuitively correlate with multi-target values. It is thus a bad idea to base evaluation and model comparison on single-target values and then later report multi-target accuracies as the former does not linearly correlate with the latter. We have noticed that rewards designed on the average per-class instead of multi-target accuracies lead to models that learn to predict only a subset of classes correctly and neglect overlaps as there is no reward for higher recognition rate of these overlaps.

C.2. Meta learned architecture definitions

We show the detailed configurations of the top three MetaQNN and ENAS neural architectures for which accuracies are shown in the main body.

Table 3 shows the definitions for the top three meta-learned models from MetaQNN on our task. Each convolutional layer is expressed through quadratic filter size and number of filters, followed by an optional specification of padding or stride. If a skip connection/convolution has been added it is added as an additional operation on the same level and we specify the layer to which it skips to. The SPP layer is characterized by the number of scales at which it pools its feature input. As an example, scales = 4 indicates

Architecture	Accuracy [%]								
	mt best val	mt bv-test	bv-bg	bv-cr	bv-sp	bv-ef	bv-eb	bv-cs	bv-avg
Alexnet	63.05	66.98	89.30	89.30	89.93	90.72	93.71	88.05	90.16
T-CNN	64.30	67.93	90.09	87.89	89.62	88.99	94.49	87.57	89.77
VGG-A	64.93	70.45	91.35	90.25	89.93	90.56	93.55	86.47	90.35
VGG-D	64.00	70.61	90.72	91.82	89.93	89.30	93.71	87.42	90.48
WRN-28-4	52.51	57.19	87.89	84.11	85.53	84.43	89.15	80.34	85.24
DenseNet-121	65.56	70.77	91.51	89.62	87.75	89.10	94.49	87.73	90.03

Table 2: Best validation model's accuracies for each individual class (bg - background, cr - crack, sp - spallation, ef - efflorescence, eb - exposed bars, cs - corrosion stain) and their average (avg) shown together with the multi-target accuracy (mt best val) and the corresponding multi-target test accuracy (mt bv-test).

Layer type	MetaQNN-1	MetaQNN-2	MetaQNN-3
conv 1	9×9 - 256, s = 2	5×5 - 128	3×3 - 128, p = 1; 1×1 - 128 (skip to 3)
conv 2	3×3 - 32, p = 1	7×7 - 32, s = 2	3×3 - 128, p = 1
conv 3	5×5 - 256	3×3 - 256, p = 1; 1×1 - 256 (skip to 5)	9×9 - 128, s = 2
conv 4	7×7 - 256, s = 2	3×3 - 256, p = 1	3×3 - 256, p = 1; 1×1 - 256 (skip to SPP)
conv 5		3×3 - 32	3×3 - 256, p = 1
conv 6		9×9 - 128, s = 2	
SPP	scales = 4	scales = 3	scales = 4
FC 1	128	128	64
classifier	linear - 6, sigmoid	linear - 6, sigmoid	linear - 6, sigmoid

Table 3: Top three neural architectures of MetaQNN for our task. Convolutional layers (conv) are parametrized by a quadratic filter size followed by the amount of filters. p and s represent padding and stride respectively. If no padding or stride is specified then $p = 0$ and $s = 1$. Skip connections are an additional operation at a layer, with the layer where the connection is attached to specified in brackets. A spatial pyramidal pooling (SPP) layer connects the convolutional feature extractor part to the classifier. Every convolutional and FC layer are followed by a batch-normalization and a ReLU and each model ends with a linear transformation with a Sigmoid function for multi-target classification.

four adaptive pooling operations such that the output width times height corresponds to $1 \times 1, 2 \times 2, \ldots 4 \times 4$. The fully-connected (FC) layer is defined by the number of feature outputs it produces. All convolutional and FC layers are followed by a batch-normalization and a ReLU layer.

Figure 7 shows graphical representations of the top three neural models of ENAS for our task. All of the ENAS architectures have seven convolutional layers followed by a linear transformation as defined prior to the search. We have chosen a visual representation instead of a table because the neural architectures (acyclic graphs) contain many skip connections that are easier to perceive this way. All convolutions have quadratic filter size and a base amount of 64 features that is doubled after the second and forth layer as defined by a DenseNet growth strategy with $k = 2$.

C.3. Transferring ImageNet and MINC features

We investigate transfer learning with features pre-trained on the ImageNet and MINC datasets for a variety of neu-

Transfer learning

Architecture	Source	Accuracy [%]	
		best val	bv-test
Alexnet	ImageNet	60.53	62.87
VGG-A	ImageNet	60.22	66.35
VGG-D	ImageNet	56.13	65.56
Densenet-121	ImageNet	54.71	57.66
Alexnet	MINC	60.06	66.50
VGG-D	MINC	61.47	67.14

Table 4: Multi-target best validation and best validation model's test accuracy for fine-tuned CNNs with convolutional feature transfer from models pre-trained on the MINC and ImageNet datasets.

ral architectures by using pre-trained weights provided by corresponding original authors. We fine-tune these models by keeping the convolutional features constant and only training the classification stage for 70 epochs with a cycled learning rate and other hyper-parameters as specified in the main body. Best multi-target validation and associated test values are reported in table 4. Although the pre-trained networks initially train much faster, we observe that transferring features from the unrelated ImageNet and MINC tasks does not help, it in fact hinders the multi-target defect classification task. We postulate that this could be due to a variety of factors like the task being too unrelated with respect to the combination of object and texture recognition demanded by our task. This observation matches previous work investigating transfer learning of object related features to texture recognition problems. In such a scenario, the authors of [3] find the need to evaluate feature importance and selectively integrate only a subset of relevant ImageNet object features to yield performance benefits for texture recognition and prevent performance degradation. We further hypothesize another possibility that the multi-target property of the task could require a different abstraction of features from those already present in the convolutional feature encoder of the pre-trained models. Further investigation of transfer learning should thus consider an approach that doesn't include all pre-trained features, selects a subset of pre-trained weights or employs different fine-tuning strategies.

C.4. Classification examples

In addition to the accuracy values reported in the main body, we show qualitative example multi-target classifications as predicted by our trained MetaQNN-1 model. We do this to give the reader a more comprehensive qualitative understanding of the complexity and challenges of our multi-target dataset. In order to better outline these challenges, we separate these examples into the following three categories:

1. Correct multi-target classification examples where all labels are predicted correctly.

2. False multi-target classification examples where at least one present defect class is recognized correctly, but one or more defect classes is missed or falsely predicted in addition.

3. False multi-target classification examples where none of the present defect classes is recognized correctly.

Corresponding images, together with ground-truth labels and the model's predictions are illustrated in the respective parts of figure 8. The few shown examples were picked to show the variety of different defect types and their combinations. Overall, the images show the challenging nature of the multi-target task. Whereas the majority of multi-target predictions are correct, the trained models face a number of different factors that make classification difficult. Particularly, partially visible defect classes, amount of overlap, variations in the surface, different exposure and illumination can lead to the model making false multi-target predictions, where only a subset of targets is predicted correctly.

C.5. Alternative dataset splits

Architecture	Multi-target accuracy [%]		Params [M]	Layers
	val	test		
Alexnet	63.50	62.94	57.02	8
T-CNN	63.87	63.00	58.60	8
VGG-A	65.33	61.93	128.79	11
VGG-D	63.76	62.50	134.28	16
WRN-28-4	59.75	55.56	5.84	28
Densenet-121	66.54	65.93	11.50	121
ENAS-1	67.71	66.31	3.41	8
ENAS-2	66.50	64.37	2.71	8
ENAS-3	65.66	65.81	1.70	8
MetaQNN-1	66.70	65.91	4.53	6
MetaQNN-2	65.25	64.82	1.22	8
MetaQNN-3	70.95	67.56	2.88	7

Table 5: Evaluation in analogy to table 2 of the main body, but on alternative dataset splits based on a per-bridge separation.

The final dataset presented in the main portion of the paper has been chosen to contain a random set of 150 unique defect examples per class for validation and test sets respectively. To avoid over-fitting we have further added the constraint that all crops stemming from bounding boxes from one image must be contained in only one of the dataset splits. The rationale behind this choice is to ensure a non-overlapping balanced test and validation set in order to avoid biased training that favors certain classes and report skewed loss and accuracy metrics.

A different alternative way of conducting such a validation and test split is to split the data based on unique bridges. Such an approach however features multiple challenges that make it infeasible to apply in practice. In particular, not every bridge has the same amount of defects and not every bridge has the same amount of defects per class. Typically also defects of varying severity and overlap are featured (e.g. some have more early-stage cracks than exposed bars). The main challenges thus are:

1. Only a certain combination of unique bridges can yield an even approximately balanced dataset split in terms of class presence.

2. Creation of class-balanced splits relies on either excluding some of the highly occurring defects or leaving

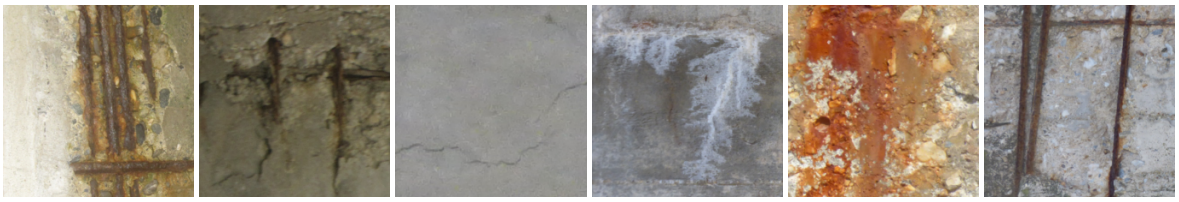
(a) ENAS-1

(b) ENAS-2

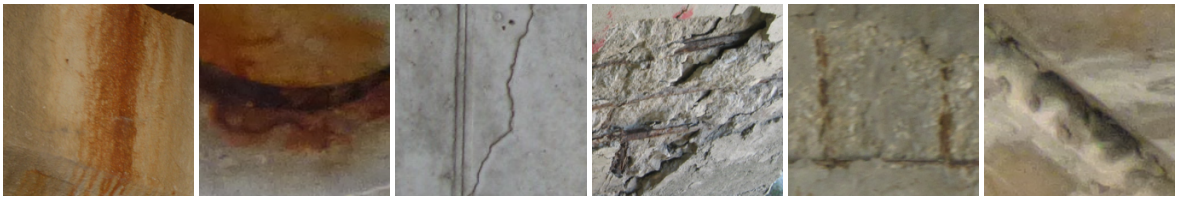
(c) ENAS-3

Figure 7: Top three neural architectures of ENAS for our task. Convolutions (conv) are denoted with quadratic filter size and a post-fix "S" for depth-wise separability. MaxPool and AvgPool are max and average pooling stages with 3×3 windows. ENAS uses a pre-determined amount of features per convolutional layer during the search and during final training uses a growth strategy of $k = 2$ similar to DenseNets. The amount of features per convolution is 64, doubled by the growth strategy after layers 2 and 4. The graph is acyclic and all connections between layers are indicated by directed arrows.

the dataset split only approximately balanced. The latter could result in training a model that favors a particular class and skewed average metrics being reported. The former can result in omitting particularly challenging or easy instances from the validation or test set and accordingly distorting the interpretation of the



(a) Correct multi-target classification examples from the validation set. From left to right: 1.) exposed bar, corrosion, spalling. 2.) spallation, exposed bars, corrosion and cracks. 3.) crack. 4.) efflorescence 5.) spallation and corrosion. 6.) spallation with exposed bars.



(b) False multi-target classification examples from the validation set where at least one present defect class is recognized correctly. From left to right: 1.) corrosion (predicted corrosion and efflorescence). 2.) corrosion (predicted corrosion, spallation and exposed bar). 3.) crack (predicted crack and efflorescence). 4.) spallation, exposed bar, corrosion (predicted spallation and corrosion). 5.) spallation, exposed bar, corrosion (predicted crack and corrosion). 6.) efflorescence (predicted efflorescence and crack).

(c) False multi-target classification examples from the validation set where none of the present defect categories is recognized correctly. From left to right: 1.) efflorescence (predicted background). 2.) crack (predicted background). 3.) exposed bar with corrosion (predicted background). 4.) efflorescence (predicted background). 5.) corrosion (predicted spallation). 6.) exposed bar (predicted crack).

Figure 8: Multi-target classification examples from the validation set using the trained MetaQNN-1 model.

model's accuracy.

3. Even when balancing the classes approximately by choosing complementary bridges, the severity of defects is not necessarily well sampled or balanced.

On the other hand, a bridge-based dataset split provides more insights with respect to over-fitting concrete properties such as surface roughness, color, context or, given that images at different bridges were acquired at different points in time with variations in global scene conditions. We therefore nevertheless investigate an alternative bridge-based dataset split that is based on three bridges for validation and test set respectively. The bridges have been chosen such that the resulting splits are approximately balanced in terms of class occurrence, albeit with the crack category being more present and the efflorescence class being under-sampled. The resulting accuracies should thus be considered with caution in direct comparison to the main paper.

Using this alternate dataset split we retrain all neural architectures presented in the main paper. We note that we have not repeated the previous hyper-parameter grid-search and simply use the previously obtained best set of hyper-parameters. In analogy, the meta-learning architecture search algorithms have not been used to sample new architectures specific to this dataset variant. The obtained final validation and test accuracies are reported in table 5. We re-iterate, that although we have coined the splits validation and test set, the sets can be used interchangeably here as no hyper-parameter tuning has been conducted on the validation set.

Obtained accuracies are similar to the experimental results presented in the paper's main body. We can observe that meta-learned architectures are not in exact previous order, e.g. MetaQNN3 outperforms MetaQNN1. However, meta-learned architectures still outperform the baselines and previous conclusions therefore hold. Due to the previously pre-

sented challenges in creation of an unbiased bridge-based dataset we therefore believe our dataset splits presented in the main body to be more meaningful to assess the models' generalization capabilities.

References

[1] Mark Everingham, S. M.Ali Ali Eslami, Luc Van Gool, Christopher K.I. I Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2014. 2

[2] Tzutalin. LabelImg. *https://github.com/tzutalin/labelImg*, 2015. 2

[3] Yan Zhang, Mete Ozay, Xing Liu, and Takayuki Okatani. Integrating deep features for material recognition. In *International Conference on Pattern Recognition (ICPR)*, 2016. 6

Chapter 2

Enabling Open Set Recognition and Continual Learning in Deep Neural Network Architectures

Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition

Martin Mundt, Sagnik Majumder, Iuliia Pliushch, Yong Won Hong, and Visvanathan Ramesh

Abstract—We introduce a probabilistic approach to unify open set recognition with the prevention of catastrophic forgetting in deep continual learning, based on variational Bayesian inference. Our single model combines a joint probabilistic encoder with a generative model and a linear classifier that get shared across sequentially arriving tasks. In order to successfully distinguish unseen unknown data from trained known tasks, we propose to bound the class specific approximate posterior by fitting regions of high density on the basis of correctly classified data points. These bounds are further used to significantly alleviate catastrophic forgetting by avoiding samples from low density areas in generative replay. Our approach requires neither storing of old, nor upfront knowledge of future data, and is empirically validated on visual and audio tasks in class incremental, as well as cross-dataset scenarios across modalities.

Index Terms—Continual Deep Learning, Catastrophic Forgetting, Open Set Recognition, Variational Inference, Deep Generative Models.

1 Introduction

Modern machine learning systems are typically trained in a closed world setting according to an isolated learning paradigm. They take on the assumption that data is available at all times and data inputs encountered during application of the learned model come from the same statistical population as the training data. However, the real world requires dealing with sequentially arriving tasks and data coming from potentially yet unknown sources. A neural network that is trained exclusively on such newly arriving data will overwrite its representations and thus forget knowledge of past tasks, an early identified phenomenon coined catastrophic forgetting [1]. Moreover, when confronting the learned model with unseen concepts, overconfident misclassification is inevitable [2].

Existing continual learning literature predominantly concentrates its efforts on finding mechanisms to alleviate catastrophic forgetting [3] and the term continual learning is not necessarily used in a wider sense. Specifically, the aforementioned crucial system component to distinguish seen from unseen unknown data, both as a guarantee for robust application and to avoid the requirement of explicit task labels for prediction, is generally missing. A naive conditioning on unseen unknown data through inclusion of a "background" class is infeasible as by definition we do not have access to it a priori. Commonly applied thresholding of prediction values is veritably insufficient as resulting large confidences cannot be prevented [2]. Arguably this also includes variational methods [4], [5], [6], [7] to gauge neural network uncertainty, since the closed world assumption also holds true for Bayesian methods [8]. Recently, Bendale et al. [9] have proposed extreme value theory (EVT) based meta-recognition to address open set detection on the basis of softmax predictions in conventional feed-forward deep neural networks. Inspired by this work, we propose a probabilistic approach to unify open set recognition and the prevention of catastrophic forgetting in continual learning of a single deep model. Our specific contributions are:

- We introduce a single model for continual learning that combines a joint probabilistic encoder with a generative model and a linear classifier. This architecture enables a natural formulation to address open set recognition on the basis of EVT bounds to the class conditional approximate posterior in variational Bayesian inference.
- Apart from using EVT for detection of unseen unknown data, we show that generated samples from areas of low probability density under the aggregate posterior can be excluded in generative replay for continual learning. This leads to significantly reduced catastrophic forgetting without storing real data.
- Empirically, we show that our model can incrementally learn the classes of two image and one audio dataset, as well as cross-dataset scenarios across modalities, while being able to successfully distinguish various unseen datasets from data belonging to known tasks.
- Finally, we demonstrate how our proposed framework can be extended and readily profits from recent advances in deep generative modelling, such as autoregression [10], [11], [12] and introspection [13], [14]. This is then empirically validated by scaling to high resolution color images in further experiments.

The remainder of the paper follows the structure of

- *M. Mundt, I. Pliushch and V. Ramesh are with the Department of Computer Science and Mathematics, Geothe University, Frankfurt am Main, Germany.*
  E-mail: {mmundt, vramesh}@em.uni-frankfurt.de
- *S. Majumder is with the Department of Computer Science, University of Texas at Austin, USA*
- *Y. W. Hong is with the Department of Computer Science, Yonsei University, Seoul, Republic of Korea.*

these listed contributions. We start section two by formally describing continual learning and open set recognition in the context of deep supervised learning, followed by a respective review of recent literature. Section three provides a step by step introduction of our probabilistic framework to unify open set recognition with the prevention of catastrophic forgetting in continual learning. Section four proceeds with an experimental evaluation and analysis, which is then revisited and extended in section five under the consideration of recent auxiliary generative modelling advances. The sixth and final section concludes the paper.

2 Background and Related Work

2.1 Continual Learning

In isolated supervised machine learning the core assumption is the presence of i.i.d. data at all times and training is conducted using a dataset $\boldsymbol{D} \equiv \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$, consisting of N pairs of data instances $\boldsymbol{x}^{(n)}$ and their corresponding labels $y^{(n)} \in \{1 \dots C\}$ for C classes. In contrast, in continual learning task data $\boldsymbol{D}_t \equiv \left\{\left(\boldsymbol{x}_t^{(n)}, y_t^{(n)}\right)\right\}_{n=1}^{N_t}$ with $t = 1, \dots, T$ arrives sequentially for T disjoint datasets, each with number of classes C_t. It is assumed that only the data of the current task is available. Different methods in the literature have been identified to prevent a model from forgetting past knowledge, either explicitly, through regularization or freezing of weights, or implicitly, through rehearsal of data by sampling retained subsets or sampling from a generative memory. A recent review of many continual learning methods is provided by Parisi et al. [3]. Here, we present a brief summary of particular related works.

Regularization and Weight Freezing: Regularization methods such as synaptic intelligence (SI) [15] or elastic weight consolidation (EWC) [16] explicitly constrain the weights during continual learning to avoid drifting too far away from previous tasks' solutions. In a related picture, learning without forgetting [17] uses knowledge distillation [18] to regularize the end-to-end functional. Further methods employ dynamically expandable neural networks [19] or progressive networks [20], that expand the capacity while freezing or regularizing existing representations.

Rehearsal: These methods store and rehearse data from distributions belonging to old tasks or generate samples in pseudo-rehearsal [21]. The central component of the former is thus the selection of significant instances. For methods such as iCarl [22] it is therefore common to resort to auxiliary techniques such as a nearest-mean classifier [23] or coresets [24]. Inspired by complementary learning systems theory [25], dual-model approaches sample data from a separate generative memory. In GeppNet [26] an additional long-short term memory [27] is used for storage, whereas generative replay [28] samples form a separately trained generative adversarial network (GAN) [29].

Bayesian Methods: As detailed in Variational Generative Replay (VGR) [6], Bayesian methods provide natural capability for continual learning by making use of the learned distribution. Existing works nevertheless fall into the above two categories: a prior-based approach using the former task's approximate posterior as the new task's prior [30] or estimating the likelihood of former data through generative replay or other forms of rehearsal [6], [7].

Evaluation Assumptions and Multiple Model Heads: The success of many of these techniques can be attributed mainly to the considered evaluation scenario. With the exception of VGR [6], all above techniques train a separate classifier per task and thus either require explicit storage of task labels, or assume the presence of a task oracle during evaluation. This multi-head classifier scenario prevents "cross-talk" between classifier units by not sharing them, which would otherwise rapidly decay the accuracy as newly introduced classes directly confuse existing concepts. While the latter is acceptable to assess catastrophic forgetting, it also signifies a major limitation in practical application. Even though VGR [6] uses a single classifier, they train a separate generative model per task to avoid catastrophic forgetting of the generator.

Our approach builds upon these previous works by proposing a single model with single classifier head with a natural mechanism for open set recognition and improved generative replay from a Bayesian perspective.

2.2 Out-of-distribution and Open Set Recognition

The above mentioned literature focuses their continual learning efforts predominantly on addressing catastrophic forgetting. Corresponding evaluation is thus conducted in a closed world setting, where instances that do not belong to the observed data distribution are not encountered. In reality, this is not guaranteed as users could provide arbitrary input or unknowingly present the system with novel inputs that deviate substantially from previously seen instances. Our models thus need the ability to identify unseen examples in an open world and categorize them as either belonging to the already known set of classes or as presently being unknown. We briefly recall the formal definition of open set recognition presented in Scheirer et al. [31] and corresponding follow-up literature [8], [9], [32], [33]: *For any recognition function f over an input space $\mathcal{X}$, the open space $\mathcal{O}$ is defined as $\mathcal{O} \subset= \mathcal{X} - \mathcal{S}_K$, where $\mathcal{S}_K$ is a union of balls of radius r_o including all of the training examples for known classes $x \in K$.* The goal in open set recognition is to learn this function f using the training data of known classes, i.e. minimizing the empirical risk R_ϵ (expected loss $\mathbb{E}[L(\dots)]$), while simultaneously limiting the open space risk $\mathcal{R}_{\mathcal{O}}(f) = \int_{\mathcal{O}} f_K(x)dx / \int_{\mathcal{S}_K} f_K(x)dx$. Minimizing the latter requires the ability to detect novelty with respect to the empirically observed distribution.

We provide a small overview of approaches that can be regarded as related to solving open set recognition in deep neural networks. A more comprehensive and general overview of recent methods is provided in the review by Boult et al. [8].

Bayesian Uncertainty and Deep Generative Models: Bayesian neural network models [34] could be believed

to intrinsically be able to reject statistical outliers through model uncertainty [6]. In inference with deep neural networks, it has been suggested that the use of stochastic forward passes with Monte-Carlo Dropout [35] provides a suitable approximation. However, repeating the argument of Boult et al. [8], this is generally insufficient as uncertain inputs are not necessarily unknown and unknowns do not necessarily have to appear as uncertain. In the context of deep generative models that are trained with various variational approximations, it is particularly well known that relying solely on deep uncertainty quantification to distinguish unseen data is unsatisfactory [36], [37].

Calibration: The aim of these works is to seperate a known and unknown input through prediction confidence, often by fine-tuning or re-training an already existing model. In ODIN [38] this is addressed through perturbations and temperature scaling, while Lee et al. [39] use a separately trained GAN to generate out-of-distribution samples from low probability densities and explicitly reduce their confidence through inclusion of an additional loss term. Similarly the objectosphere loss [40] defines an objective that explicitly aims to maximize entropy for upfront available unknown inputs.

Extreme Value Theory: One approach to open set recognition in deep neural networks is through extreme-value theory (EVT) based meta-recognition [9], [32], i.e. without re-training or modifying loss functions by assuming upfront presence of unknown data. The goal here is to bound the open space on the basis of already seen data instances. Scheirer et al. [32] have introduced the notion of a compact abating probability, a probabilistic model where the recognition function's probability decreases monotonically with increasing distance to known training points. They have identified the Weibull distribution as a suitable candidate to satisfy the latter when modelling the extreme prediction values. Bendale et al. [9] have extended this to the use with deep neural networks. They empirically observe that the penultimate layer of a deep neural network can be used as the underlying feature space for open set recognition. On the basis of extreme values to this layer's average activation values, the authors devise a procedure to revise a deep neural network's softmax prediction values. The proposed OpenMax algorithm thus aims to mitigate the issue of predicted scores summing to unity and unseen unknown data instances can in principle be assigned zero probability across all known classes.

Our work extends these approaches by moving away from potentially non-calibrated predictive values or empirically chosen deep neural network feature spaces. We instead propose to use EVT to bound the approximate posterior. In contrast to predictive values such as reconstruction losses, where differences in reconstructed images do not necessarily have to reflect the outcome with respect to our task's target, we thus directly operate on the underlying (lower-bound to the) data distribution, and the generative factors. This additionally allows us to constrain generative replay to distribution inliers, which further alleviates catastrophic forgetting in continual learning. While we can still leverage variational inference to gauge model uncertainty, the need to rely on classifier entropy or confidence, that are known to be overconfident and can never be calibrated for all unknown inputs, is circumvented.

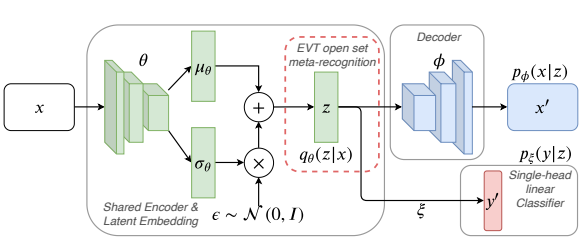


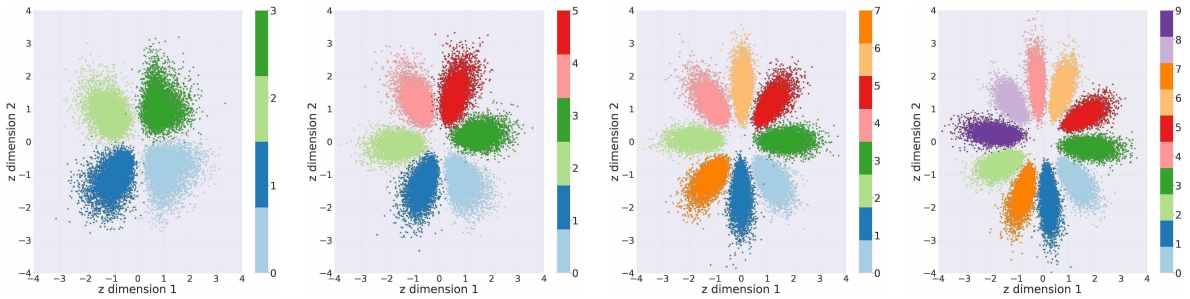
Fig. 1: Joint continual learning model consisting of a shared probabilistic encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, probabilistic decoder $p_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})$ and probabilistic classifier $p_{\boldsymbol{\xi}}(y, \boldsymbol{z})$. For open set recognition and generative replay with outlier rejection, EVT based bounds on the basis of the approximate posterior are established.

3 Unifying Open Set Recognition with the Prevention of Catastrophic Forgetting in Continual Learning

We consider the continual learning scenario with awareness of an open world from a perspective of variational inference in deep generative models [5]. Our model consists of a shared encoder with variational parameters $\boldsymbol{\theta}$, decoder and linear classifier with respective parameters $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$. The joint probabilistic encoder learns an encoding to a latent variable $\boldsymbol{z}$, over which a unit Gaussian prior is placed. Using variational inference, the encoder's purpose is to approximate the true posterior to both $p_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z})$ and $p_{\boldsymbol{\xi}}(y, \boldsymbol{z})$. The probabilistic decoder $p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$ and probabilistic linear classifier $p_{\boldsymbol{\xi}}(y|\boldsymbol{z})$ then return the conditional probability density of the input $\boldsymbol{x}$ and target y under the respective generative model given a sample $\boldsymbol{z}$ from the approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. This yields a generative model $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, for which we assume a factorization and generative process of the form $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})$. For variational inference with this model, the sum over all elements in the dataset n $\in$ D of the following loss thus needs to be optimized:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) &= -\beta KL(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)}) \,||\, p(\boldsymbol{z})) \\ &+ \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)})}\left[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}^{(n)}|\boldsymbol{z}) + \log p_{\boldsymbol{\xi}}(y^{(n)}|\boldsymbol{z})\right]\end{aligned} \tag{1}$$

This model can be seen as a variant of β-VAE [41], where in addition to approximating the data distribution the model learns to incorporate the class structure into the latent space. It forms the basis for continual learning with open set recognition and respective improvements to generative replay, which will be discussed in subsequent sections. An illustration of the model is shown in figure 1 and the corresponding full derivation of equation 1, the lower-bound to the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$ is supplied in the appendix.



(a) 4-classes (b) 6-classes (c) 8-classes (d) 10-classes

Fig. 2: (a) 2-D latent space visualization for continually learned MNIST.

3.1 Learning Continually through Generative Replay

Without further constraints, one could continually train above model by sequentially accumulating and optimizing equation 1 over all currently present tasks $t = 1, \ldots, T$:

$$\mathcal{L}_t^{\text{UB}}(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = \frac{1}{t} \sum_{\tau=1}^{t} \frac{1}{N_\tau} \sum_{n=1}^{N_\tau} \mathcal{L}\left(\boldsymbol{x}_\tau^{(n)}, \boldsymbol{y}_\tau^{(n)}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) \tag{2}$$

Being based on the accumulation of real data, this equation provides an upper-bound to achievable performance in continual learning. However, this form of continued training is generally infeasible if only the most recent task's data is assumed to be available. Making use of the generative nature of our model, we follow previous works [6], [7] and estimate the likelihood of former data through generative replay:

$$\begin{aligned} \mathcal{L}_t(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}) = &\frac{1}{2} \frac{1}{N_t} \sum_{n=1}^{N_t} \mathcal{L}\left(\boldsymbol{x}_t^{(n)}, \boldsymbol{y}_t^{(n)}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) \\ &+ \frac{1}{2} \frac{1}{N_t'} \sum_{n=1}^{N_t'} \mathcal{L}\left(\boldsymbol{x}_t'^{(n)}, \boldsymbol{y}_t'^{(n)}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) \end{aligned} \tag{3}$$

where,

$$\boldsymbol{x}_t' \sim p_{\boldsymbol{\phi}, t-1}(\boldsymbol{x}|\boldsymbol{z});\ y_t' \sim p_{\boldsymbol{\xi}, t-1}(y|\boldsymbol{z}) \text{ and } \boldsymbol{z} \sim p(\boldsymbol{z}) \tag{4}$$

Here, $\boldsymbol{x}_t'$ is a sample from the generative model with its corresponding label y_t' obtained from the classifier. N_t' is the number of total data instances of all previously seen tasks or alternatively a hyper-parameter. This way the expectation of the log-likelihood for all previously seen tasks is estimated and the dataset at any point in time $\tilde{\boldsymbol{D}}_t \equiv \left\{(\tilde{\boldsymbol{x}}_t^{(n)}, \tilde{y}_t^{(n)})\right\}_{n=1}^{\tilde{N}_t} = \{(\boldsymbol{x}_t \cup \boldsymbol{x}_t', y_t \cup y_t')\}$ is a combination of generations from seen past data distributions and the current task's real data.

3.2 Linear Classifier Expansion and the Role of β

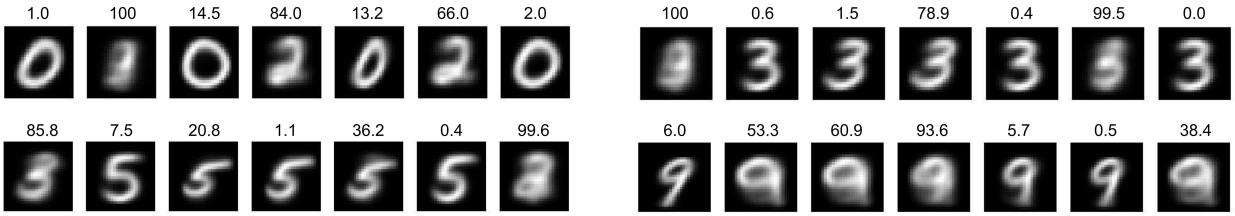
In contrast to prior works based on multiple models, our approach of using equation 3 to continually train a single model has two implications. With every encounter of an additional class: 1. a new classifier unit and corresponding weights need to be added. 2: the latent encoding needs to adjust to accommodate the additional class under the constraint of the classifier requirement of linear separability.

The first implication can be addressed by expanding the existing classifier weight tensor and only initializing the newly added weights. If the distribution from which the newly added weights are drawn is independent of the number of classes and only depends on the input dimensionality, such as the initialization scheme proposed by He et al. [42], the initialization scheme remains constant throughout training. While the addition itself will temporarily confuse existing units, this should make sure that newly added parameters are on the same scale as existing weights and thus trained in practice. Note that in principle, during the optimization of a task the weight distribution could shift significantly from its initial state. However, we do not encounter this potential issue in empirical experiments. Nevertheless, we point out that this currently under-explored topic requires separate future investigation in the context of model expansion.

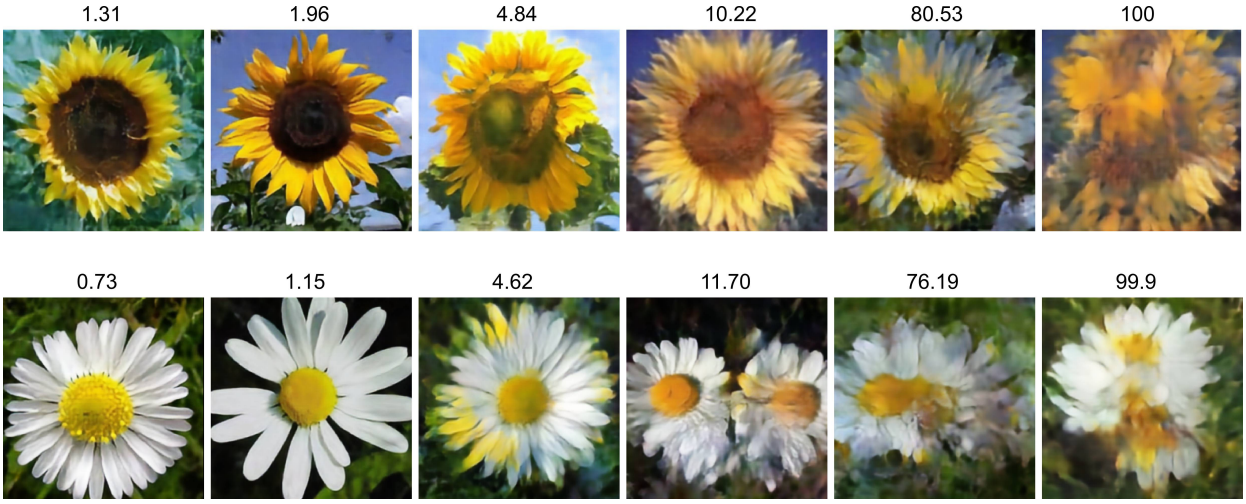
For the second implication, the β term of equation 1 is crucial. Here, the role of beta is to control the capacity of the information bottleneck and regulate the effective latent encoding overlap [43], which can best be summarized with a direct quote from the recent work of Mathieu et al. [44]: *"The overlap factor is perhaps best understood by considering extremes: too little, and the latents effectively become a lookup table; too much, and the data and latents do not convey information about each other. In either case, meaningfulness of the latent encodings is lost." (p. 4)*. This can be seen as under- or over-regularization by the prior of what is typically referred to as the aggregate posterior [45]:

$$q_{\boldsymbol{\theta}, t}(\boldsymbol{z}) = \mathbb{E}_{p_{\tilde{D}_t}(\tilde{\boldsymbol{x}})}\left[q_{\boldsymbol{\theta}, t}(\boldsymbol{z}|\tilde{\boldsymbol{x}})\right] \approx \frac{1}{\tilde{N}_t} \sum_{n=1}^{\tilde{N}_t} q_{\boldsymbol{\theta}, t}(\boldsymbol{z}|\tilde{\boldsymbol{x}}^{(n)}) \tag{5}$$

As an extension of this argument to our model, the necessity of linear class separation given $\boldsymbol{z}$ requires a suitable level of encoding overlap. This forms the basis for our open set recognition and respective improved generative replay for continual learning, which will be discussed in the following paragraphs. Example two-dimensional latent encodings for a continually trained MNIST [46] model with appropriate β are shown in figure 2. Here, we can see that the classes are cleanly separated in latent space, as enforced by the linear classification objective, and new classes can be accommodated continually. Further discussion on the choice of β can be found in the supplementary material.



(a) MNIST: 28×28 resolution classified as $c = 0$ (top left), $c = 3$ (top right), $c = 5$ (bottom left) and $c = 9$ (bottom right). Images were generated from the two-dimensional latent space visualized in figure 2.



(b) Flowers: 256×256 resolution classified as "sunflower" (top row) and "daisy" (bottom row). Images generated from a 60 dimensional latent space of deep wide residual models trained with introspection, as detailed in later experimental sections.

Fig. 3: Generated images $x \sim p_{\phi,t}(\boldsymbol{x}|\boldsymbol{z})$ with $\boldsymbol{z} \sim p(\boldsymbol{z})$ and their corresponding class c obtained from the classifier $p_{\boldsymbol{\xi},t}(y|\boldsymbol{z})$ together with their open set outlier percentage. Image quality degradation and class ambiguity can be observed with increasing outlier likelihood. Flower images have been compressed for side-by-side view.

3.3 Open Set Recognition and Generative Replay with Statistical Outlier Rejection

Trained naively in above fashion, our model would suffer from accumulated errors with each successive iteration of generative replay, similar to current literature approaches. The main challenge is that high density areas under the prior $p(\boldsymbol{z})$ are not necessarily reflected in the structure of the aggregate posterior $q_{\boldsymbol{\theta},t}(\boldsymbol{z})$ [47]. Thus, generated data from low density regions of the latter does not generally correspond to encountered data instances. Conversely, data instances that fall into high density regions under the prior should not generally be considered as statistical inliers with respect to the observed data distribution.

Ideally, this challenge would be solved by modifying equations 1 and 2 by replacing the Gaussian prior in the KL-divergence with $q_{\boldsymbol{\theta},t}(\boldsymbol{z})$ and respectively sampling $z \sim q_{\boldsymbol{\theta},t-1}(\boldsymbol{z})$ for generative replay in equations 3 and 4. Even though using the aggregate posterior as the prior is the objective in multiple recent works, it can be challenging in high dimensions, lead to over-fitting and often comes at the expense of additional hyper-parameters [47], [48], [49]. To avoid finding an explicit representation for the multi-modal $q_{\boldsymbol{\theta},t}(\boldsymbol{z})$, we leverage our model's class disentanglement and draw inspiration from the EVT based OpenMax approach [9] in deep neural networks. However, instead of using knowledge about extreme distance values in penultimate layer activations to modify a Softmax prediction's confidence, we propose to apply EVT on the basis of the class conditional aggregate posterior. In this view, any sample can be regarded as statistically outlying if its distance to the classes' latent means is extreme with respect to what has been observed for the majority of correctly predicted data instances, i.e. the sample falls into a region of low density under the aggregate posterior and is less likely to belong to $p_{\tilde{D}}(\tilde{\boldsymbol{x}})$.

For convenience, let us introduce the indices of all correctly classified data instances at the end of task t as $m = 1, \ldots, \tilde{M}_t$. To construct a statistical meta-recognition model, we first obtain each class' mean latent vector for all correctly predicted seen data instances:

$$\bar{\boldsymbol{z}}_{c,t} = \frac{1}{|\tilde{M}_{c,t}|} \sum_{m \in \tilde{M}_{c,t}} \mathbb{E}_{q_{\theta,t}(\boldsymbol{z}|\tilde{\boldsymbol{x}}_t^{(m)})} [\boldsymbol{z}] \tag{6}$$

and define the respective set of latent distances as:

$$\Delta_{c,t} \equiv \left\{ f_d \left(\bar{\boldsymbol{z}}_{c,t}, \mathbb{E}_{q_{\theta,t}(\boldsymbol{z}|\tilde{\boldsymbol{x}}_t^{(m)})} [\boldsymbol{z}] \right) \right\}_{m \in \tilde{M}_{c,t}} \tag{7}$$

Here, f_d signifies a choice of distance metric. We proceed to model this set of distances with a per class heavy-tail

Weibull distribution $\boldsymbol{\rho}_{c,t} = (\tau_{c,t}, \kappa_{c,t}, \lambda_{c,t})$ on $\Delta_{c,t}$ for a given tail-size η. As these distances are based on the class conditional approximate posterior, we can thus bound the latent space regions of high density. The tightness of the bounds is characterized through η, that can be seen as a prior belief with respect to the outlier quantity assumed to be inherently present in the data distribution. The choice of f_d determines the nature and dimensionality of the obtained distance distribution. For our experiments, we find that the cosine distance and thus a univariate Weibull distance distribution per class seems to be sufficient.

Using the cumulative distribution function of this Weibull model $\boldsymbol{\rho}_t$ we can now estimate any sample's outlier probability:

$$\omega_{\boldsymbol{\rho},t}(\boldsymbol{z}) = \texttt{min}\left(1 - \exp\left(-\frac{|f_d\left(\bar{\boldsymbol{z}}_t, \boldsymbol{z}\right) - \boldsymbol{\tau}_t|}{\boldsymbol{\lambda}_t}\right)^{\boldsymbol{\kappa}_t}\right) \quad (8)$$

where the minimum returns the smallest outlier probability across all classes. If this outlier probability is larger than a prior rejection probability Ω_t, the instance can be considered as unknown as it is far away from all known classes. For a novel data instance, the outlier probability can be based on computation of the probabilistic encoder $\boldsymbol{z} \sim q_{\boldsymbol{\theta},\boldsymbol{t}}(\boldsymbol{z}|\boldsymbol{x})$ and a false overconfident classifier prediction avoided. Analogously, for the generative model, equation 8 can be used with $\boldsymbol{z} \sim p(\boldsymbol{z})$ and the probabilistic decoder only calculated for samples that are considered to be statistically inlying. This way, we can constrain the naive generative replay of equation 4 to the aggregate posterior, while avoiding the need to sample $\boldsymbol{z} \sim q_{\boldsymbol{\theta},t}(\boldsymbol{z})$ directly. Although this may sound detrimental to our method, it comes with the advantage of scalability to high dimensions. We further argue that the computational overhead for generative replay, both from sampling from the prior $\boldsymbol{z} \sim p(\boldsymbol{z})$ in large parallelized batches and computation of equation 8, is negligible in contrast to the much more computationally heavy deep probabilistic decoder or even the linear classifier, as the latter only need to be calculated for accepted samples. To give a visual illustration, we show examples of generated MNIST [46] and larger resolution flower images [50] together with their outlier percentage in figure 3.

4 Experiments and Analysis

Similar to recent literature [3], [6], [15], [16], [28], we consider the incremental MNIST [46] dataset, where classes arrive in groups of two, and corresponding versions of the FashionMNIST [51] and AudioMNIST dataset [52]. For the latter we follow the authors' procedure of converting the audio recordings into spectrograms. In addition to this class incremental setting, we also evaluate cross-dataset scenarios, where datasets are sequentially added with all of their classes and the model has to learn across modalities.

For a common frame of reference, we base both encoder and decoder architectures on 14-layer wide residual networks with a latent dimensionality of 60 [11], [12], [53], [54]. For the generative replay with statistical outlier rejection, we use an aggressive rejection rate of $\Omega_t = 0.01$ and dynamically set tail-sizes to 5% of seen examples per class. To avoid over-fitting, we add noise sampled from $\mathcal{N}(0, 0.25)$ to each input. This is preferable to weight regularization as it doesn't entail unrecoverable units that are needed to encode later tasks. We thus refer to our proposed model as Open-set Classifying Denoising Variational Auto-Encoder (OCDVAE), for which we have found a value of $\beta = 0.1$ to consistently work well, see discussion in the appendix. An important practical aspect is that we include normalizing terms into our previously introduced loss functions in order to have a set-up that is agnostic to dataset properties such as image resolution or task complexity that manifests in minimum required latent dimensionality. Specifically, we normalize the reconstruction loss by the spatial data dimension, i.e. dividing it by the number of pixels, and the KL divergence by the latent dimensionality. This way, we do not need to find a different value for beta if the latent dimensionality is altered or alternatively scaling the reconstruction loss' magnitude if the input size is increased. We empirically compare the following methods:

Dual Model: separate generative and discriminative variational models in analogy to the deep generative replay of Shin et al. [28].
EWC: elastic weight consolidation [16] for a purely discriminative model.
OCDVAE (ours): our proposed joint model with posterior based open set recognition and resulting statistical outlier rejection in generative replay.
CDVAE: the naive approach of generating from the prior distribution in our joint model. We include these results to highlight the effect of aggregate posterior to prior mismatch.
ISO: isolated learning, where all data is always present.
UB: upper-bound on achievable model performance by sequentially accumulating all data, given by equation 2.
LB: lower-bound on model performance when only the current task's data is available. No additional mechanism is in place and full catastrophic forgetting occurs.

Our evaluation metrics are inspired by previously proposed continual learning measures [55], [56]. In addition to overall accuracy $\alpha_{t,all}$, these metrics monitor forgetting by computing a base accuracy $\alpha_{t,base}$ for the initial task at increment t, while also gauging the amount of new knowledge that can be encoded by monitoring the accuracy for the most recent increment $\alpha_{t,new}$. We evaluate the quality of the generative models through classification accuracy as it depends on generated replay and a direct evaluation of pixel-wise reconstruction losses is not necessarily coupled to classification accuracy or retention thereof. However, we provide a detailed analysis of reconstruction losses for all tasks, as well as KL divergences for all experiments in the supplementary material.

To provide a fair comparison of achievable accuracy, all above approaches are trained to converge on each task using the Adam optimizer [57]. We repeat all experiments five times to assess statistical consistency. The full hyper-parameter specification can be found in the supplementary material. There, we also provide the quantitative continual learning results for all experiments with a 2-hidden layer and 400 unit multi-layer perceptron [56], as the WRN architecture could be argued to be excessively large for simpler datasets such as MNIST, in particular with the parameters of the

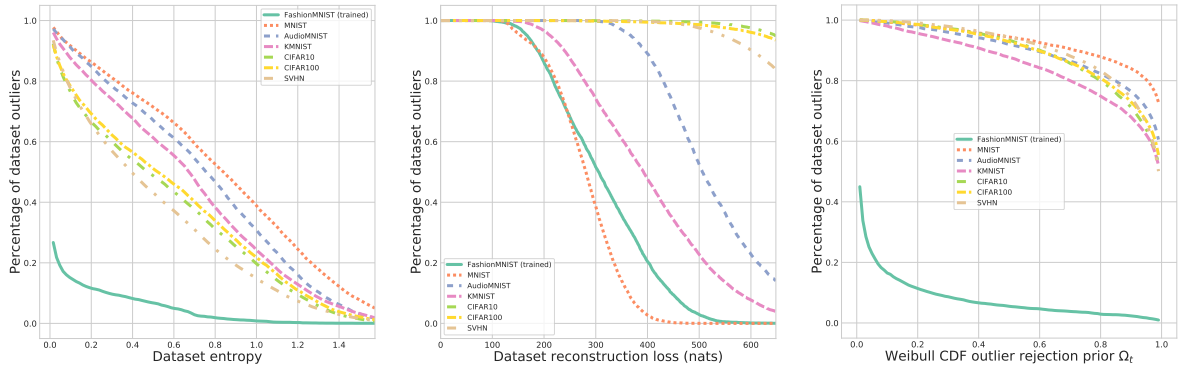


Fig. 4: Trained FashionMNIST OCDVAE evaluated on unknown datasets. All metrics are averaged over 100 approximate posterior samples per data point. (Left) Classifier entropy values are insufficient to separate most of unknown from the known task's test data. (Center) Reconstruction loss allows for a partial distinction. (Right) Our posterior based open set recognition considers the large majority of unknown data as statistical outliers across a wide range of rejection priors Ω_t.

TABLE 1: Results for continual learning across datasets averaged over 5 runs, baselines and the reference isolated learning scenario for FashionMNIST (F) $\rightarrow$ MNIST (M) $\rightarrow$ AudioMNIST (A) and the reverse order. α_T indicates the respective accuracy at the end of the last increment $T = 3$.

Cross-dataset		α_T(%) (T=3)		
		base	new	all
F-M-A	**CDVAE ISO**			94.95
	CDVAE UB	89.10	97.88	95.00
	CDVAE LB	00.00	98.12	22.70
	EWC	22.85 $\pm$ 0.294	93.31 $\pm$ 0.138	43.42 $\pm$ 0.063
	Dual Model	81.89 $\pm$ 0.104	96.78 $\pm$ 0.067	91.75 $\pm$ 0.064
	CDVAE	57.70 $\pm$ 4.480	96.73 $\pm$ 0.235	81.10 $\pm$ 1.769
	OCDVAE	80.11 $\pm$ 2.922	97.63 $\pm$ 0.042	91.13 $\pm$ 1.045
A-M-F	**CDVAE ISO**			94.95
	CDVAE UB	97.17	89.16	94.91
	CDVAE LB	00.00	89.72	34.51
	EWC	3.420 $\pm$ 0.026	87.54 $\pm$ 0.214	45.42 $\pm$ 0.731
	Dual Model	66.82 $\pm$ 0.337	89.15 $\pm$ 0.050	87.70 $\pm$ 0.102
	CDVAE	79.74 $\pm$ 2.431	88.50 $\pm$ 0.126	89.46 $\pm$ 0.600
	OCDVAE	94.53 $\pm$ 0.283	89.53 $\pm$ 0.367	94.06 $\pm$ 0.156

network being on a similar scale as the dataset itself. Note that all drawn conclusions remain the same independent of the architecture used and the main difference is a mild degradation in performance as an expected consequence of the less complex architecture. All models were trained on a single GTX 1080 GPU.

4.1 Learning Across Datasets in an Open World

Achieved accuracies for continual learning across datasets are summarized in table 1. In general the upper-bound values are almost identical to isolated learning. Similarly, the new task's metrics are negligibly close, as the WRN architecture ensures enough capacity to encode new knowledge. In contrast to EWC that is universally unable to maintain knowledge in a single-head classifier, as also previously observed by [3], [56], approaches based on generative replay are able to partially retain information. Yet they accumulate errors due to samples generated from low density regions. This is noticeable for both the dual model approach, with a separate VAE and discriminative model, and more heavily so for the naive CDVAE, where the structure of $q_{\boldsymbol{\theta},t}(\boldsymbol{z})$ is further affected by the discriminator. However, our proposed OCDVAE model overcomes this issue to a considerable degree, rivalling and improving upon the separately trained models.

Apart from these classification accuracies, we also quantitatively analyze the models' ability to distinguish unknown tasks' data from data belonging to known tasks. Here, the challenge is to consider all unseen test data of already trained tasks as inlying, while successfully identifying 100 % of unknown datasets as outliers. For this purpose, we evaluate models after training on one dataset on its respective test set, the remaining tasks' datasets and additionally the KMNIST [58], SVHN [59] and CIFAR [60] datasets.

We compare and contrast three criteria that could be used for open set recognition: classifier predictive entropy, reconstruction loss and our proposed latent based EVT approach. We approximate the expectation with 100 variational samples from the approximate posterior per data point, i.e. marginalising the latent variable $\boldsymbol{z}$ with Monte Carlo samples from $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. Figure 4 shows the three criteria and respective percentage of the total dataset being considered as outlying for the OCDVAE model trained on FashionMNIST. In consistence with [36], we can observe that use of reconstruction loss can sometimes distinguish between the known tasks' test data and unknown datasets, but results in failure for others. In the case of classifier predictive entropy, depending on the exact choice of entropy threshold, generally only a partial separation can be achieved. Furthermore, both of these criteria pose the additional challenge of results being highly dependent on the choice of the precise cut-off value. In contrast, the test data from the known tasks is regarded as inlying across a wide range of rejection priors Ω_t and the majority of other datasets is consistently regarded as outlying by our proposed open set mechanism.

We provide quantitative outlier detection accuracies in table 2. Here, a 5% validation split is used to determine the respective value at which 95% of the validation data is considered as inlying before using these priors to determine

TABLE 2: Test accuracies and outlier detection values of the joint OCDVAE and dual model (VAE and separate deep classifier, denoted as "CL + VAE") approaches when considering 95 % of known tasks' validation data is inlying. Percentage of detected outliers is reported based on classifier predictive entropy, reconstruction loss and our posterior based EVT approach, averaged over 100 $\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ samples per data-point respectively. Note that larger values are better, except for the test data of the trained dataset, where ideally 0% should be considered as outlying.

Outlier detection at 95% validation inliers (%)				MNIST	Fashion	Audio	KMNIST	CIFAR10	CIFAR100	SVHN
Trained	Model	Test acc.	Criterion							
FashionMNIST	Dual, CL + VAE	90.48	Class entropy	74.71	5.461	69.65	77.85	24.91	28.76	36.64
			Reconstruction	5.535	5.340	64.10	31.33	99.50	98.41	97.24
			Latent EVT	96.22	5.138	93.00	91.51	71.82	72.08	73.85
	Joint, OCDVAE	90.92	Class Entropy	66.91	5.145	61.86	56.14	43.98	46.59	37.85
			Reconstruction	0.601	5.483	63.00	28.69	99.67	98.91	98.56
			Latent EVT	96.23	5.216	94.76	96.07	96.15	95.94	96.84
MNIST	Dual, CL + VAE	99.40	Class entropy	4.160	90.43	97.53	95.29	98.54	98.63	95.51
			Reconstruction	5.522	99.98	99.97	99.98	99.99	99.96	99.98
			Latent EVT	4.362	99.41	99.80	99.86	99.95	99.97	99.52
	Joint, OCDVAE	99.53	Class entropy	3.948	95.15	98.55	95.49	99.47	99.34	97.98
			Reconstruction	5.083	99.50	99.98	99.91	99.97	99.99	99.98
			Latent EVT	4.361	99.78	99.67	99.73	99.96	99.93	99.70
AudioMNIST	Dual, CL + VAE	98.53	Class entropy	97.63	57.64	5.066	95.53	66.49	65.25	54.91
			Reconstruction	6.235	46.32	4.433	98.73	98.63	98.63	97.45
			Latent EVT	99.82	78.74	5.038	99.47	93.44	92.76	88.73
	Joint, OCDVAE	98.57	Class entropy	99.23	89.33	5.731	99.15	92.31	91.06	85.77
			Reconstruction	0.614	38.50	3.966	36.05	98.62	98.54	96.99
			Latent EVT	99.91	99.53	5.089	99.81	100.0	99.99	99.98

outlier counts for the known tasks' test set as well as other datasets. We provide this evaluation for both our joint model, as well as separate discriminative and generative models. While MNIST seems to be an easy to identify dataset for all approaches, we can make two major observations:

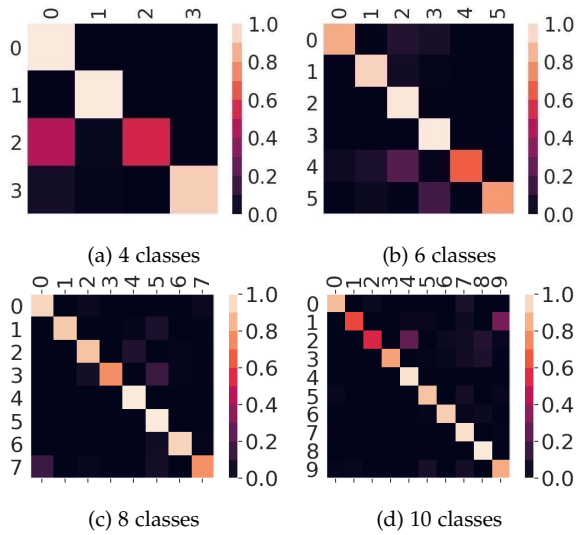
1) The latent based EVT approach outperforms the other criteria, particularly for the OCDVAE where a near perfect open set detection can be achieved.
2) Even though we can apply EVT to purely discriminative models, the joint OCDVAE model consistently exhibits more accurate outlier detection. We hypothesize that this is due to the joint model also optimizing a variational lower bound to the data distribution $p(\boldsymbol{x})$ in addition to taking into account labels.

We provide figures similar to figure 4 for all models reported in table 2 in the supplementary material.

Naively one might at this point be tempted to argue that the trained weights of the individual deep neural network encoder layers are still deterministic and the failure of predictive entropy as a measure for unseen unknown data could thus primarily be attributed to uncertainty not being expressed adequately. Placing a distribution on the weights would then be expected to resolve this issue. Although it has previously been argued that this is not the case [8], we further repeat all of our quantitative open set experiments by treating the model weights as the random variable being marginalised through the use of MC-Dropout [35]. Whereas some improvements upon the presented results of this section are noticeable, they are overall negligible with respect to observed patterns, the two major observations formulated in above list, and drawn conclusions. The corresponding table containing the quantitative Monte-Carlo Dropout results have accordingly been moved to the supplementary material.

TABLE 3: Results for class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for the three datasets. α_T indicates the respective accuracy at the end of the last increment $T = 5$.

Class-incremental		α_T(%) (T=5)		
		base	new	all
FashionMNIST	**CDVAE ISO**			89.54
	CDVAE UB	92.20	97.50	89.24
	CDVAE LB	00.00	99.80	19.97
	EWC	00.17 $\pm$ 0.076	99.60 $\pm$ 0.023	20.06 $\pm$ 0.059
	Dual Model	94.26 $\pm$ 0.192	93.55 $\pm$ 0.708	63.21 $\pm$ 1.957
	CDVAE	39.51 $\pm$ 7.173	96.92 $\pm$ 0.774	58.82 $\pm$ 2.521
	OCDVAE	60.63 $\pm$ 12.16	96.51 $\pm$ 0.707	69.88 $\pm$ 1.712
MNIST	**CDVAE ISO**			99.45
	CDVAE UB	99.57	99.10	99.29
	CDVAE LB	00.00	99.85	20.16
	EWC	00.45 $\pm$ 0.059	99.58 $\pm$ 0.052	20.26 $\pm$ 0.027
	Dual Model	97.31 $\pm$ 0.489	98.59 $\pm$ 0.106	96.64 $\pm$ 0.079
	CDVAE	19.86 $\pm$ 7.396	99.00 $\pm$ 0.100	64.34 $\pm$ 4.903
	OCDVAE	92.35 $\pm$ 4.485	99.06 $\pm$ 0.171	93.24 $\pm$ 3.742
AudioMNIST	**CDVAE ISO**			97.75
	CDVAE UB	98.42	98.67	97.87
	CDVAE LB	00.00	100.0	20.02
	EWC	00.11 $\pm$ 0.007	99.41 $\pm$ 0.207	19.98 $\pm$ 0.032
	Dual Model	61.58 $\pm$ 0.747	89.41 $\pm$ 0.691	47.42 $\pm$ 1.447
	CDVAE	59.36 $\pm$ 7.147	84.93 $\pm$ 6.297	81.49 $\pm$ 1.944
	OCDVAE	79.73 $\pm$ 4.070	89.52 $\pm$ 6.586	87.72 $\pm$ 1.594



(a) 4 classes (b) 6 classes

(c) 8 classes (d) 10 classes

Fig. 5: AudioMNIST confusion matrices for the incrementally learned OCDVAE model. When adding classes two and three the model experiences difficulty in classification, however is able to overcome this challenge by exhibiting backward transfer when later learning classes four and five. Similarly, classes four and five are then retrospectively improved through the addition of classes six and seven. It is also observable how forgetting of the initial classes is limited.

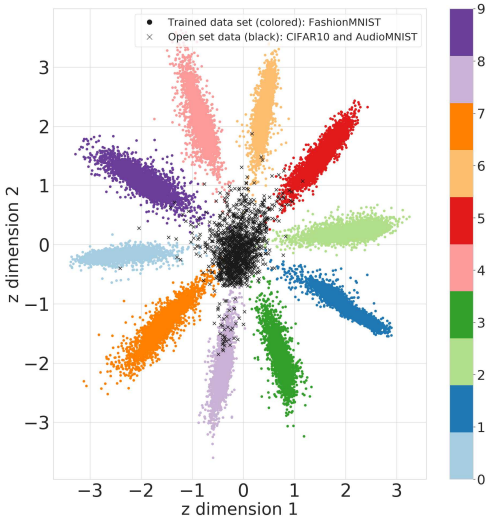


Fig. 6: Latent space visualization for OCDVAE with a two-dimensional latent space trained on FashionMNIST. In addition to the learned classes, embedded data points for unseen unknown classes belonging to AudioMNIST and CIFAR10 are shown. The latter can be observed to be separable by their distance to trained concepts.

4.2 Learning Classes Incrementally

We show results in analogy to table 1 for the class incremental scenario in table 3. With the exception of MNIST, where the dual model approach fares well, a similar pattern as before can be observed and our proposed OCDVAE approach significantly outperforms all other methods. Interestingly, as a result of using a single model across tasks, we observe backward transfer in some experiments. We dedicate the next subsection to this desirable phenomenon and tie its forthcoming discussion to potential limitations of regularization based continual learning methods.

4.3 Backward Transfer and the Limits of Regularization

The existing tasks' representations are typically exploited in the acquisition of a new task's information in continual learning, transfer learning and all other scenarios that formulate some kind of incremental learning problem. However, the concept of backward transfer is generally less deliberated. It describes the reverse phenomenon where introduction of a new task leads to learning of representations that retrospectively improve former tasks. We observe this behavior in multiple of our experiments, whose detailed numerical account together with examples of all generated images for all increments $t = 1, \ldots, 5$ can be found in the supplementary material. For the purpose of the following discussion, it is sufficient to single out one particularly noteworthy example of backward transfer. Figure 5 highlights the occurrence of retrospective improvement for class-incremental learning with our OCDVAE model on the AudioMNIST dataset, as quantitatively presented in the tables of the supplementary material. Here, the addition of two new classes (four and five) at the end of the second increment leads to an improvement in the classification performance on class two, as illustrated by the confusion matrices. Analogously, subsequent inclusion of the additional classes six and seven enhances the classes of the second task increment, even if none of the former tasks' real data is present any longer. We point out that this is continual learning desideratum can only emerge from having a single model with a single classification head and alleviated catastrophic forgetting through mechanisms different from heavy regularization. By definition, obtaining retrospective improvement through regularization is unlikely, if not entirely unachievable. This is because continual learning through regularization encourages the model to reproduce solutions for previous tasks by maintaining the parameters or upholding a specific prediction, e.g. through knowledge distillation [17], [61], [62], [63]. The focus therefore lies on avoiding model deterioration, without the possibility of surpassing previously reached performance. Although cases were backward transfer may not always be necessary are conceivable, e.g. if a task's performance requirements are already met from the start, this inability for retroactive correction can be one major drawback of regularization methods.

At this point, we emphasize that the goal of this question is not to altogether question the merit of prior works that have made use of regularization techniques. Instead, we would simply like to raise awareness that there exist contexts in which regularization techniques might be helpful and on the contrary, settings, where use of regularization may be in direct opposition to the desired goals. Apart from task sequences where backward transfer can be of essence, another context in which current continual learning

regularization methods may be antagonistic is the objective of open set recognition. In particular, we posit that commonly employed regularization techniques and the ability to recognize the open set are interdependent. To specify this statement, figure 6 shows another visualization of a trained model's two-dimensional latent embedding for FashionMNIST, similar to the MNIST visualization of figure 2. However, here we have also included the probabilistic encoder's mapping of previously unseen unknown classes from the AudioMNIST and CIFAR10 datasets. On the one hand, it is observable how the corresponding latent values have large distance to the clusters belonging to the learned classes, painting an intuitive two-dimensional picture for the earlier demonstrated success of our framework in open set recognition in high-dimensional latent spaces. On the other hand, the large majority of the unseen unknown data points fall into a central cluster. If we now desire to incorporate these currently unseen unknown classes by including them into the next continual learning task, this single cluster of unseen unknowns will need to be divided in order for the individual classes to be discriminable. In analogy to the visualized rearrangement of the latent space over time in figure 2, the aggregate posterior thus need to be given the flexibility to experience ample change. If despite of this requirement a regularization approach regularizes the current aggregate posterior $q_{\boldsymbol{\theta},t}(\boldsymbol{z})$, e.g. by replacing the Gaussian prior with the former tasks' aggregate posterior $KL(q_{\boldsymbol{\theta},t}(\boldsymbol{z}|\boldsymbol{x}^{(n)}) \mid\mid q_{\boldsymbol{\theta},t-1}(\boldsymbol{z}))$ in equation 1 such as proposed in the variational continual learning (VCL) [30], this may not be possible. A similar argument provides the rationale behind the earlier empirically demonstrated failure of EWC, where restrictions on updates to the probabilistic encoder's parameters hinder the disambiguation of new classes or conversely discount the solution for previous tasks.

Before we continue to showcase ways in which our proposed framework can naturally be scaled to high-resolution color images, we would like to give credit to related works that have purposely not been included in our experimental comparison for an entirely different reason. These approaches are naturally synergistic reporting them separately in a standalone quantitative comparison could mislead the reader. They primarily belong to the category of *exemplar/core set rehearsal*. Prominent examples are iCarl [22], gradient episodic memory [55], FearNet [64], Variational Continual Learning [30] or CLEAR [65]. The retention and rehearsal of real data can always be a valid strategy to address the challenge of continual learning, if memory is of little concern. The problem is then re-framed to the discovery of suitable data subset selection schemes. The latter can naturally be integrated into our proposed framework by devising mechanisms to select data subsets which best approximate the aggregate posterior of the entire dataset.

5 Improving the Generative Model: Scaling through Autoregression and Introspection

At the time of their initial introduction, it was notorious that variational autoencoders produce blurry examples and were associated with an inability to scale to more complex high-resolution color images. This is in contrast to their prominent generative counterparts, the generative adversarial network

TABLE 4: PixelVAE based continual learning approaches averaged over 5 runs in analogy to tables 1 and 3.

Class-incremental		α_T(%) (T=5)		
		base	new	all
Fashion	**Dual Pix Model**	60.04 ± 5.151	98.85 ± 0.141	72.41 ± 2.941
	PixCDVAE	47.83 ± 13.41	97.91 ± 0.596	63.05 ± 1.826
	PixOCDVAE	74.45 ± 2.889	98.63 ± 0.176	80.85 ± 0.721
MNIST	**Dual Pix Model**	98.04 ± 1.397	97.31 ± 0.575	96.52 ± 0.658
	PixCDVAE	56.53 ± 4.032	96.77 ± 0.337	83.61 ± 0.927
	PixOCDVAE	97.44 ± 0.785	98.63 ± 0.430	96.84 ± 0.346
Audio	**Dual Pix Model**	64.60 ± 8.739	98.18 ± 0.885	75.50 ± 3.032
	PixCDVAE	29.94 ± 18.47	97.00 ± 0.520	63.44 ± 5.252
	PixOCDVAE	75.25 ± 10.18	99.43 ± 0.495	90.23 ± 1.139
Cross-dataset		α_T(%) (T=3)		
		base	new	all
F-M-A	**Dual Pix Model**	82.88 ± 0.116	97.23 ± 0.212	92.16 ± 0.061
	PixCDVAE	56.44 ± 1.831	97.50 ± 0.184	80.76 ± 0.842
	PixOCDVAE	81.84 ± 0.212	97.75 ± 0.169	91.76 ± 0.212
A-M-F	**Dual Pix Model**	71.58 ± 2.536	88.76 ± 0.255	88.61 ± 0.547
	PixCDVAE	49.38 ± 2.256	88.54 ± 0.042	82.18 ± 0.672
	PixOCDVAE	91.90 ± 0.282	89.91 ± 0.177	93.82 ± 0.354

[29]. Although this stigma perhaps still holds until today, there has been many successful recent efforts to address this challenge. In what is supposed to constitute a final outlook for our work, we thus empirically investigate the choice of generative model and optionally improve the probabilistic decoding with the help of two promising research directions: autoregression [10], [11], [12] and introspection [13], [14]. The commonality between these approaches is their aim to overcome the limitations of independent pixel-wise reconstructions. We will briefly summarize these generative extensions, empirically show their advantage by revisiting our previous experiments, before continuing to demonstrate our framework's efficacy on high resolution color images.

5.1 Improvements through Autoregressive Decoding

In essence, autoregressive models improve the probabilistic decoder by introducing a spatial conditioning of each scalar output value on the previous ones, in addition to conditioning on the latent variable:

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_i p(x_i|x_1, \ldots, x_{i-1}, \boldsymbol{z}) \tag{9}$$

In an image, generation thus needs to proceed pixel by pixel and is commonly referred to as PixelVAE [11]. This conditioning is generally achieved by providing the input to the decoder during training, i.e. including a skip path that bypasses the probabilistic encoding. A concurrent introduction of autoregressive VAEs has thus coined this model "lossy" [12]. This is because local information can now be modelled without access to the latent variable and only the global information will be encoded in $\boldsymbol{z}$.

We repeat our previously shown continual learning experiments with three additional appended autoregressive decoder layers, each with a kernel size of 7×7 and 60 channels, following the experimental set-up of the original PixelVAE. We also follow the authors' recommendation to

train the decoder using a 256-way Softmax and treating the reconstruction as classification in practice. Results corresponding to tables 1 and 3 for these pixel models are shown in table 4. While we can observe that the introduction of the autoregressive decoder generally further alleviates catastrophic forgetting, it does significantly more so for our proposed approach.

5.2 Introspection and Adversarial Training

Although the earlier shown accuracies of generative replay with autoregression are assuring, autoregressive sampling comes with a major caveat. When attempting to operate on larger data, the computational complexity of the pixel by pixel data creation procedure grows in direct proportion to the input dimensionality. With increasing input size, the repeated calculation of the autoregressive decoder layers can thus rapidly render the generation required for optimization of equation 3 practically infeasible. A promising alternative perspective towards autoencoding beyond pixel similarities is to leverage the insights obtained from GANs. To this matter, Larsen et al. [66] have proposed a hybrid model called VAEGAN. Here, the crucial idea is to append a GAN style adversarial discriminator to the variational autoencoder. This yields a model that promises to overcome a conventional GAN's mode collapse issues, as the VAE is responsible for the rich encoding, while letting the added discriminator judge the decoder's output based on perceptual criteria rather than individual pixel values. The more recent IntroVAE [13] and adversarial encoder generator networks [14] have subsequently come to the realization that this doesn't necessarily require the auxiliary real-fake discriminator, as the VAE itself already provides strong means for discrimination, namely its probabilistic encoder. We leverage this idea of introspection for our framework, as it doesn't require any architectural or structural changes beyond an additional term in the loss function.

For sake of brevity we denote the probabilistic encoder through their parameters $\boldsymbol{\phi}$ and decoder $\boldsymbol{\theta}$ in the following equations. Training our model with introspection is then equivalent to adding the following two terms to our previously formulated loss function:

$$\begin{aligned} &\mathcal{L}_{IntroCDVAE\_Enc} = \\ &\mathcal{L}_{CDVAE} - \beta \left[m - KL(\boldsymbol{\theta}(\boldsymbol{\phi}(\boldsymbol{z})) \mid\mid p(\boldsymbol{z})) \right]^{+} \end{aligned} \tag{10}$$

and

$$\mathcal{L}_{IntroCDVAE\_Dec} = \mathcal{L}_{Rec} - \beta KL(\boldsymbol{\theta}(\boldsymbol{\phi}(\boldsymbol{z})) \mid\mid p(\boldsymbol{z})) \tag{11}$$

Here, $\mathcal{L}_{CDVAE}$ corresponds to the full loss of equation 1 and $\mathcal{L}_{Rec}$ corresponds to the reconstruction loss portion: $\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)})}[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}^{(n)}|\boldsymbol{z})]$. In above equations, we have followed the original authors proposal to include a positive margin m, with $[\cdot]$ denoting $max(0, \cdot)$. This hinge loss formulation serves the purpose of empirically limiting the encoder's reward to avoid a too massive gap in a min-max game of above competing KL terms. Aside from the regular loss that encourages the encoder to match the approximate posterior to the prior for real data, the encoder is now further driven to maximize the deviation from the posterior to the prior for generated images. Conversely, the decoder is encouraged to "fool" the encoder into producing a posterior distribution that matches the prior for these generated images. The optimization is conducted jointly. In comparison with a traditional VAE, this can thus be seen as training in an adversarial like manner, without necessitating additional discriminative models. As such, introspection fits naturally into our proposed framework and no further changes are required.

Before we proceed with demonstrating the empirical value, we note that the original authors of IntroVAE have introduced additional weighting terms in front of the reconstruction loss, in order to drastically lower its magnitude, and the added KL divergence. We have observed that the former is simply due to lack of normalization with respect to input width and height and hence the reconstruction loss growing proportionally with the spatial input size, whereas the KL divergence typically does not reflect this behavior for a fixed-size latent space. Given that we have included this normalization in our practical experimentation, we have found this additional hyper-parameter to be unnecessary. The other hyper-parameter to weight the added adversarial KL divergence term is essentially equivalent to the already introduced beta, alas without our motivation in earlier sections, but simply as a heuristic to not overpower the reconstruction loss.

5.3 Incrementally Learning High Resolution Flowers

In this section, we empirically demonstrate the efficacy of generative modelling advances for high resolution natural data and respective improvements by using our proposed open set aware approach. For this purpose, we continual learn five types of flowers, in analogy to the experiment conducted in the recent Lifelong GAN [63]. Whereas the latter makes use of lower resolutions, we let the resolution remain at 256×256 pixels to demonstrate application of our approach to high-resolution. Apart from the high resolution, this scenario is interesting for two further reasons: the dataset contains less than 100 images per class and the classes are introduced one by one in continual training. This introduction of a single class makes a multi-head approach unrealisable, and thus a large portion of previously proposed approaches based on task labels and regularization, infeasible. The small-sample scenario also underlines that deep generative models can be trained without massive amounts of data. Due to the larger resolution we employ a deeper variant of our previously used 14-layer WRN architecture. In addition to the three convolution blocks that comprise a total of 12 layers, three further blocks are added, resulting in a 26 layer architecture that down-samples the input by an extra factor of eight across the added stages. This way, the encoded spatial dimensionality that precedes the 60 dimensional latent space is the same for WRN-14 experiments on 32×32 resolution and this section's experiments based on a WRN-26 and 256×256 resolution. We use a batch size of 32 and let all other hyper-parameters remain the same as described for previous experiments. The only exception is the amount of epochs, which we increase to 2000 per task in order to reach a significant amount of update iterations as a result of the small dataset size. In analogy to previous sections we report results as the average over multiple runs, with the exception of the autoregressive models. As the latter

had to be trained on multiple NVIDIA V100 GPUs over the course of three weeks for a single experiment, we report a single run. We did not repeat this experiment as below results are believed to sufficiently demonstrate limitations and are notably surpassed by the computationally favorable introspection model.

5.3.1 Denoising and the choice of perturbation

In the previous continual learning experiments, the introduced denoising acted as one way to avoid over-fitting, akin to the use of data augmentation. However, the choice of noise distribution can have an additional, very different purpose. Recall that in a wider sense, "denoising" refers to the concept of introducing an arbitrarily sampled perturbation that is added to the input, but needs to be discounted in reconstruction with respect to the original unperturbed data instance. This perturbation doesn't necessarily need to take on the earlier introduced pixel-wise noise from a Gaussian or Uniform distribution to alter each pixel independently. If our primarily interest lies in maintaining the discriminative performance of our model and less so on the visual quality of the generated data, we can take advantage of the perturbation distribution as means to encode our prior knowledge of common generative pitfalls. For example, in our specific context, it is well known that a traditional VAE without further advances commonly fails to generate non-blurry, crisp images. However, we can include and work around this belief by letting the denoising assume the form of deblurring, e.g. by stochastically adding a variety of Gaussian blurs to subsets of inputs. Even though the decoder is ultimately still encouraged to remove this blur and reconstruct the original clean image, the encoder is now inherently required to learn how to manage blurry input. It is encouraged to build up a natural invariance to our choice of perturbation. In the context of maintaining a classifier with generative replay, to an extend it should then no longer be a strict requirement to replay locally detailed crisp images, as long as the information required for discrimination is present.

5.3.2 Results and discussion

Figure 7 shows the quantitative accuracy of the evaluated methods and the continual learning upper and lower-bound. As usual, the lower-bound corresponds to predicting just the present class correctly and full catastrophic forgetting occurring for all previously seen concepts. The upper-bound shows an expected gradual decay with increased amount of classes. For all introduced model variants, we can observe significant improvement over baseline versions (dashed lines) with the introduction of our open set method (solid lines). As expected, the plain OCDVAE model is further substantially outperformed by the introspective model. The latter closely mirrors the upper-bound performance and starts to deviate only after multiple repetitions of generative replay. Although the open set aware generation also generally enhances the autoregressive baseline, the autoregressive PixOCDVAE, perhaps to the readers surprise, fares comparably much worse than the OCDVAE or IntroOCDVAE counterparts. We can observe the empirical rationale for this in a qualitative illustration of select generated samples for each continual learning step, provided in figure 8. For the PixOCDVAE

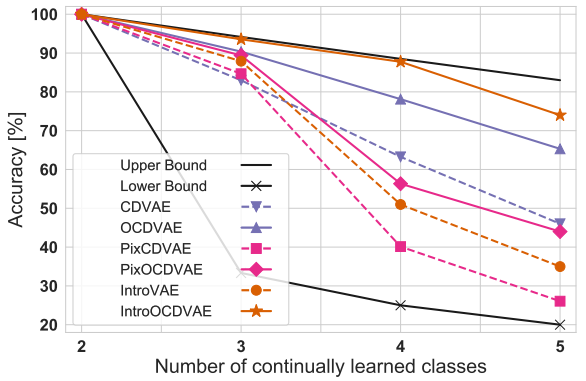


Fig. 7: Continual learning accuracies for flowers at 256×256 resolution to demonstrate how generative modelling advances enable scaling of our framework. Pairs of colored lines show respective improvements of our proposed aggregate posterior constrained generative replay (solid lines) over the open set unaware baselines (dashed lines). Whereas every model surpasses the lower-bound and thus to an extent alleviates catastrophic forgetting, our proposed framework in conjunction with introspection clearly beats the other contestants and approaches the upper-bound achievable accuracy. An accompanying qualitative illustration of generated images is provided in figure 8.

we can see that the initial task's generative replay is locally consistent, which is reflected in the quantitative accuracy values for the first tasks being almost indistinguishable from the other models. However, starting from the second cycle of generative replay, the conditioning of equation 9 seems to lose long-term correlations and an increasing amount of the image is filled with noise with each further step. In a visual comparison between IntroVAE and IntroOCDVAE, we again observe that ambiguous interpolations rapidly take over without constraining generative replay to aggregate posterior inliers, recall figure 4.

For OCDVAE, we observe that all images are blurry from the start. Even though the classes are distinguishable, this blur is amplified over time. The respectively very high accuracy of fig 7 can be attributed to the deblurring objective, where the encoder's hypothesized blur invariance largely compensates the model's inability to generate detailed examples. As a result, the accuracy gap between OCDVAE and IntroOCDVAE is rather small, despite what we as humans would perhaps initially expect from the visually less pleasing images. Correspondingly, when the deblurring is removed, we observe major drops in the final OCDVAE accuracy of up to 15%, with negligible degradation of reported values for the highly detailed introspection images. As a final remark, we note that Lifelong GAN [63] and MeRGAN [62] have conducted similar experiments on the flower dataset. We did not explicitly include results for the latter for two reasons. First, at this stage, it should be clear that an "either or" comparison is deceptive, as VAEs and GANs can go hand in hand to benefit each other. Second, these works have conducted experiments at a lower resolution and we were simply unable to reach accuracies at higher resolution

Fig. 8: Generated 256×256 flower images for various continually trained models. Images have been selected to provide a qualitative intuition behind the quantiative results of figure 7. The unmodified OCDVAE appears to suffer from the limitations of a traditional VAE and generates blurry images, although performs remarkably well in terms of quantitative classification. Its open set unaware counterpart CDVAE deteriorates similarly to earlier experiments due to the generation of ambiguous samples from low density areas outside the aggregate posterior. PixOCDVAE is initially competitive but rapidly loses long-range correlations of the autoregressive conditioning, resulting in increasingly noisy images. Introspection significantly increases the image detail, albeit still degrades considerable due to ambiguous interpolations. This is again resolved by combining introspection with our proposed posterior based EVT approach, where image quality is retained across multiple generative replay steps. Images have been compressed for a side-by-side view.

that would do the method justice, without resorting to substantial hyperparameter and architecture tuning. We have thus decided in favor of showing higher resolution experiments in contrast to a comparison on more heavily down-sampled images. Depending on the precise setup, MeRGAN and Lifelong GAN have been reported to result in final accuracies between 60% and 85% respectively [63], values that are generally similar to the ones reported in 7. However, note that the former achieves this accuracy by keeping a complete model copy at all times, whereas the latter makes use of auxiliary data and augmentation. This is in addition to both of these works requiring a separately trained deep discriminative model to solve the classification task. Neither of them considers the challenge of open set recognition, where the uniqueness of our work lies, and treats the problem in a closed world. With this in mind, we encourage future replay based continual learning to further explore generative modelling advances and their hybrid combinations, while keeping in mind that continual learning goes beyond subjective visual generation quality and measuring catastrophic forgetting.

6 Conclusion

We have proposed a probabilistic approach to unify the prevention of catastrophic forgetting with open set recognition based on variational inference in continual learning. Using a single model that combines a shared probabilistic encoder with a generative model and an expanding linear classifier, we have introduced EVT based bounds to the approximate posterior. The derived open set recognition and corresponding generative replay with statistical outlier rejection have been shown to achieve compelling results in both task incremental as well as cross-dataset continual learning across image and audio modalities, while being able to distinguish seen from unseen data. Our approach readily benefits from recent generative modelling techniques, which has been empirically demonstrated in the context of high resolution flower images. We expect future work to explore more natural synergies with further generative modelling advances and investigate a range of practical applications.

References

[1] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks : The Sequential Learning Problem," *Psychology of Learning and Motivation - Advances in Research and Theory*, vol. 24, no. C, pp. 109–165, 1989.

[2] O. Matan, R. Kiang, C. E. Stenard, and B. E. Boser, "Handwritten Character Recognition Using Neural Network Architectures," *4th USPS Advanced Technology Conference*, vol. 2, no. 5, pp. 1003–1011, 1990.

[3] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual Lifelong Learning with Neural Networks: A Review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[4] A. Graves, "Practical variational inference for neural networks," *Neural Information Processing Systems (NeurIPS)*, 2011.

[5] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *International Conference on Learning Representations (ICLR)*, 2013.

[6] S. Farquhar and Y. Gal, "A Unifying Bayesian View of Continual Learning," *Neural Information Processing Systems (NeurIPS) Bayesian Deep Learning Workshop*, 2018.

[7] A. Achille, T. Eccles, L. Matthey, C. P. Burgess, N. Watters, A. Lerchner, and I. Higgins, "Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies," *Neural Information Processing Systems (NeurIPS)*, 2018.

[8] T. E. Boult, S. Cruz, A. Dhamija, M. Gunther, J. Henrydoss, and W. Scheirer, "Learning and the Unknown : Surveying Steps Toward Open World Recognition," *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[9] A. Bendale and T. E. Boult, "Towards Open Set Deep Networks," *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," *International Conference on Machine Learning (ICML)*, vol. 48, pp. 1747–1756, 2016.

[11] I. Gulrajani, K. Kumar, A. Faruk, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, "PixelVAE: a Latent Variable Model for Natural Images," *International Conference on Learning Representations (ICLR)*, 2017.

[12] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational Lossy Autoencoder," *International Conference on Learning Representations (ICLR)*, 2017.

[13] H. Huang, Z. Li, R. He, Z. Sun, and T. Tan, "Introvae: Introspective variational autoencoders for photographic image synthesis," *Neural Information Processing Systems (NeurIPS)*, 2018.

[14] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "It Takes (Only) Two : Adversarial Generator-Encoder Networks," *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[15] F. Zenke, B. Poole, and S. Ganguli, "Continual Learning Through Synaptic Intelligence," *International Conference on Machine Learning (ICML)*, vol. 70, pp. 3987–3995, 2017.

[16] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 114, no. 13, pp. 3521–3526, 2017.

[17] Z. Li and D. Hoiem, "Learning without forgetting," *European Conference on Computer Vision (ECCV)*, 2016.

[18] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *NeurIPS Deep Learning Workshop*, 2014.

[19] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong Learning with Dynamically Expandable Networks," *International Conference on Learning Representations (ICLR)*, 2018.

[20] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive Neural Networks," *arXiv preprint arXiv:1606.04671*, 2016.

[21] A. Robins, "Catastrophic Forgetting, Rehearsal and Pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[22] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] T. Mensink, J. Verbeek, F. Perronnin, G. Csurka, T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric Learning for Large Scale Image Classification : Generalizing to New Classes at Near-Zero Cost," *European Conference on Computer Vision (ECCV)*, 2012.

[24] O. Bachem, M. Lucic, and A. Krause, "Coresets for Nonparametric Estimation - the Case of DP-Means," *International Conference on Machine Learning (ICML)*, vol. 37, pp. 209–217, 2015.

[25] R. C. O'Reilly and K. A. Norman, "Hippocampal and neocortical contributions to memory: Advances in the complementary learning systems framework," *Trends in Cognitive Sciences*, vol. 6, no. 12, pp. 505–510, 2003.

[26] A. Gepperth and C. Karaoguz, "A Bio-Inspired Incremental Learning Architecture for Applied Perceptual Problems," *Cognitive Computation*, vol. 8, no. 5, pp. 924–934, 2016.

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] H. Shin, J. K. Lee, and J. J. Kim, "Continual Learning with Deep Generative Replay," *Neural Information Processing Systems (NeurIPS)*, 2017.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Neural Information Processing Systems (NeurIPS)*, 2014.

[30] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational Continual Learning," *International Conference on Learning Representations (ICLR)*, 2018.

[31] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards Open Set Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 7, pp. 1757–1772, 2013.

[32] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability Models For Open Set Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[33] A. Bendale and T. Boult, "Towards Open World Recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] D. J. C. MacKay, "A Practical Bayesian Framework," *Neural Computation*, vol. 472, no. 1, pp. 448–472, 1992.

[35] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation : Representing Model Uncertainty in Deep Learning," *International Conference on Machine Learning (ICML)*, vol. 48, 2015.

[36] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do Deep Generative Models Know What They Don't Know?" *International Conference on Learning Representations (ICLR)*, 2019.

[37] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, "Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift," *Neural Information Processing Systems (NeurIPS)*, 2019.

[38] S. Liang, Y. Li, and R. Srikant, "Enhancing the Reliability of Out-of-distribution Image Detection in Neural Networks," *International Conference on Learning Representations (ICLR)*, 2018.

[39] K. Lee, H. Lee, K. Lee, and J. Shin, "Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples," *International Conference on Learning Representations (ICLR)*, 2018.

[40] A. R. Dhamija, M. Günther, and T. E. Boult, "Reducing Network Agnostophobia," *Neural Information Processing Systems (NeurIPS)*, 2018.

[41] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *International Conference on Learning Representations (ICLR)*, 2017.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *International Conference on Computer Vision (ICCV)*, 2015.

[43] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta-VAE," *Neural Information Processing Systems (NeurIPS), Workshop on Learning Disentangled Representations*, 2017.

[44] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh, "Disentangling disentanglement in variational autoencoders," *International Conference on Machine Learning (ICML)*, pp. 7744–7754, 2019.

[45] M. D. Hoffman and M. J. Johnson, "ELBO surgery: yet another way to carve up the variational evidence lower bound," *Neural Information Processing Systems (NeurIPS), Advances in Approximate Bayesian Inference Workshop*, 2016.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[47] J. M. Tomczak and M. Welling, "VAE with a vampprior," *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 84, 2018.

[48] M. Bauer and A. Mnih, "Resampled Priors for Variational Autoencoders," *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 89, 2019.

[49] H. Takahashi, T. Iwata, Y. Yamanaka, M. Yamada, and S. Yagi, "Variational Autoencoder with Implicit Optimal Priors," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5066–5073, 2019.

[50] M. E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[51] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," *arXiv preprint arXiv: 1708.07747*, 2017.

[52] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek, "Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals," *arXiv preprint arXiv: 1807.03418*, 2018.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[54] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," *British Machine Vision Conference (BMVC)*, 2016.

[55] D. Lopez-Paz and M. A. Ranzato, "Gradient Episodic Memory for Continual Learning," *Neural Information Processing Systems (NeurIPS)*, 2017.

[56] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring Catastrophic Forgetting in Neural Networks," *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[57] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[58] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep Learning for Classical Japanese Literature," *Neural Information Processing Systems (NeurIPS), Workshop on Machine Learning for Creativity and Design*, 2018.

[59] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning," *Neural Information Processing Systems (NeurIPS), Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[60] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Toronto, Tech. Rep., 2009.

[61] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder Based Lifelong Learning," *International Conference on Computer Vision (ICCV)*, 2017.

[62] C. Wu, L. Herranz, X. Liu, Y. Wang, J. van de Weijer, and B. Raducanu, "Memory Replay GANs: learning to generate images from new categories without forgetting," *Neural Information Processing Systems (NeurIPS)*, 2018.

[63] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong GAN: Continual Learning for Conditional Image Generation," *International Conference on Computer Vision (ICCV)*, 2019.

[64] R. Kemker and C. Kanan, "FearNet: Brain-inspired model for incremental learning," *International Conference on Learning Representations (ICLR)*, 2018.

[65] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large Scale Incremental Learning," *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[66] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *International Conference On Machine Learning (ICML)*, 2016.

Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition - Supplementary Material

Martin Mundt, Sagnik Majumder, Iuliia Pliushch, Yong Won Hong, and Visvanathan Ramesh

APPENDIX

The supplementary material provides further details for the material presented in the main body. Specifically, the structure is as follows:

- **A.** Derivation of our model and loss in equation 1 of the main body.
- **B.** Extended discussion, qualitative and quantitative examples for the role of β.
- **C.** Full specification of training procedure and hyper-parameters, including exact architecture definitions.
- **D.** Additional visualization of open set detection for all quantitatively evaluated models considered in table 2 of the main body.
- **E.** Continual learning results with a multi-layer perceptron (MLP).
- **F.** Full continual learning results for all task increments of the MNIST, FashionMNIST and AudioMNIST main body experiments, including all reconstruction losses and KL divergences.
- **G.** Visualization of generative replay examples for MNIST, FashionMNIST and AudioMNIST.

A. LOSS DERIVATION

As mentioned in the main body of the paper, in supervised continual learning we are confronted with a dataset $\boldsymbol{D} \equiv \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$, consisting of N pairs of data instances $\boldsymbol{x}^{(n)}$ and their corresponding labels $y^{(n)} \in \{1 \dots C\}$ for C classes. We consider a problem scenario similar to the one introduced in "Auto-Encoding Variational Bayes" [1], i.e. we assume that there exists a data generation process responsible for the creation of the labelled data given some random latent variable $\boldsymbol{z}$. For simplicity, we follow the authors' derivation for our model with the additional inclusion of data labels, but without the β term that is present in the main body.

Ideally we would like to maximize $p(\boldsymbol{x}, \boldsymbol{y}) = \int p(\boldsymbol{z}) p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{z}) d\boldsymbol{z}$, where the integral and the true posterior density

$$p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}) = \frac{p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x}, \boldsymbol{y})} \tag{1}$$

are intractable. We thus follow the standard practice of using variational Bayesian inference and introducing an approximation to the posterior $q(\boldsymbol{z})$, for which we will specify the exact form later. Making use of the properties of logarithms and applying above Bayes rule, we can now write:

$$\begin{aligned}\log p(\boldsymbol{x}, \boldsymbol{y}) = \int q(\boldsymbol{z}) [&\log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{z}) + \log p(\boldsymbol{z}) \\ &- \log p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}) + \log q(\boldsymbol{z}) - \log q(\boldsymbol{z})] d\boldsymbol{z},\end{aligned} \tag{2}$$

as the left-hand side is independent of $\boldsymbol{z}$ and $\int q(\boldsymbol{z}) d\boldsymbol{z} = 1$. Using the definition of the Kullback-Leibler divergence (KLD) $KL(q \,||\, p) = -\int q(x) \log(p(x)/q(x))$ we can rewrite this as:

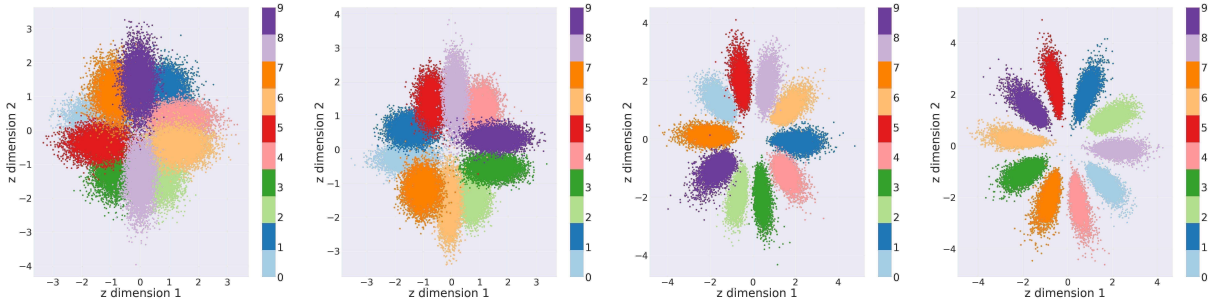
$$\begin{aligned}&\log p(\boldsymbol{x}, \boldsymbol{y}) - KL(q(\boldsymbol{z}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})) = \\ &\mathbb{E}_{q(\boldsymbol{z})} \left[\log p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{z})\right] - KL(q(\boldsymbol{z}) \,||\, p(\boldsymbol{z}))\end{aligned} \tag{3}$$

Here, the right hand side forms a variational lower-bound to the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$ as the KLD between approximate and true posterior on the left hand side is strictly positive.

At this point we make two choices that deviate from prior works that made use of labelled data in the context of generative models for semi-supervised learning [2]. We assume a factorization of the generative process of the form $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})$ and introduce a dependency of $q(\boldsymbol{z})$ on $\boldsymbol{x}$, but not explicitly on $\boldsymbol{y}$, i.e. $q(\boldsymbol{z}|\boldsymbol{x})$. In contrast to class-conditional generation, this dependency essentially assumes that all information about the label can be captured by the latent $\boldsymbol{z}$ and there is thus no additional benefit in explicitly providing the label when estimating the data likelihood $p(\boldsymbol{x}|\boldsymbol{z})$. This is crucial as our probabilistic encoder should be able to predict labels without requiring it as input to our model, i.e. $q(\boldsymbol{z}|\boldsymbol{x})$ instead of the intuitive choice of $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$. However, we would like the label to nevertheless be directly inferable from the latent $\boldsymbol{z}$. In order for the latter to be achievable, we require the corresponding classifier that learns to predict $p(\boldsymbol{y}|\boldsymbol{z})$ to be linear in nature. This guarantees linear separability of the classes in latent space, which can in turn then be used to for open set recognition and generation of specific classes as shown in the main body.

B. FURTHER DISCUSSION ON THE ROLE OF β

In the main body the role of the β term [3] in our model's loss function is pointed out. Here, we delve into further detail with qualitative and quantitative examples to support the



(a) β 1.0 (b) β 0.5 (c) β 0.1 (d) β 0.05

Fig. 1: 2-D MNIST latent space visualization with different β values for the used WRN architecture.

arguments. To facilitate the discussion, we repeat equation 1 of the main body:

$$\mathcal{L}\left(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) = -\beta KL(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)}) \,||\, p(\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)})}\left[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}^{(n)}|\boldsymbol{z}) + \log p_{\boldsymbol{\xi}}(y^{(n)}|\boldsymbol{z})\right] \tag{4}$$

The β term weights the strength of the regularization by the prior through the KL divergence. Selection of this strength is necessary to control the information bottleneck of the latent space and regulate the effective latent encoding overlap. To repeat the main body, and previous arguments by [4] and [5]: too large β values (typically $>> 1$) will result in a collapse of any structure present in the aggregate posterior. Too small β values (typically $<< 1$) lead to the latent space being a lookup table. In either case, there is no meaningful information between the latents. This is particularly relevant to our objective of linear class separability, that requires formation of an aggregate latent encoding that is disentangled with respect to the different classes. To visualize this, we have trained multiple models with different β values on the MNIST dataset, in an isolated fashion with all data present at all times to focus on the effect of β. The corresponding aggregate encodings at the end of training are shown in figure 1. Here, we can empirically observe above points. With a beta of one and larger, the aggregate posterior's structure starts to collapse and the aggregate encoding converges to a Normal distribution. While this minimizes the distributional mismatch with respect to the prior, the separability of classes is also lost and an accurate classification cannot be achieved. On the other hand, if the beta value gets ever smaller there is insufficient regularization present and the aggregate posterior no longer follows a Normal distribution. The latter does not only render sampling for generative replay difficult, it also challenges the assumption of distances to each class' latent mean being Weibull distributed, as the latter can essentially be seen as a skewed Normal.

As pointed out in the main body, it is important to note that the losses are normalized with respect to spatial image and latent dimensionality. The value of β should thus also be seen as a normalized quantity. While the relative effect of increasing or decreasing beta stays the same, the absolute value of β can be subject to any normalization.

We provide corresponding quantitative examples for the models trained with different β with 2-D latent spaces and 60-D latent spaces in tables 1 and 2 respectively. In both cases,

TABLE 1: Losses obtained for different β values for MNIST using the WRN architecture with 2-D latent space. Training conducted in isolated fashion to quantitatively showcase the role of β. Un-normalized values in nats are reported in brackets for reference purposes.

		In nats per dimension (nats in brackets)			
2-D latent	Beta	KLD	Recon loss	Class Loss	Accuracy [%]
train	1.0	1.039 (2.078)	0.237 (185.8)	0.539 (5.39)	79.87
test		1.030 (2.060)	0.235 (184.3)	0.596 (5.96)	78.30
train	0.5	1.406 (2.812)	0.230 (180.4)	0.221 (2.21)	93.88
test		1.382 (2.764)	0.228 (178.8)	0.305 (3.05)	92.07
train	0.1	2.055 (4.110)	0.214 (167.8)	0.042 (0.42)	99.68
test		2.071 (4.142)	0.212 (166.3)	0.116 (1.16)	98.73
train	0.05	2.395 (4.790)	0.208 (163.1)	0.025 (0.25)	99.83
test		2.382 (4.764)	0.206 (161.6)	0.159 (1.59)	98.79

TABLE 2: Losses obtained for different β values for MNIST using the WRN architecture with 60-D latent space. Training conducted in isolated fashion to quantitatively showcase the role of β. Un-normalized values in nats are reported in brackets for reference purposes.

		In nats per dimension (nats in brackets)			
60-D latent	Beta	KLD	Recon loss	Class Loss	Accuracy [%]
train	1.0	0.108 (6.480)	0.184 (144.3)	0.0110 (0.110)	99.71
test		0.110 (6.600)	0.181 (142.0)	0.0457 (0.457)	99.03
train	0.5	0.151 (9.060)	0.162 (127.1)	0.0052 (0.052)	99.87
test		0.156 (9.360)	0.159 (124.7)	0.0451 (0.451)	99.14
train	0.1	0.346 (20.76)	0.124 (97.22)	0.0022 (0.022)	99.95
test		0.342 (20.52)	0.126 (98.79)	0.0286 (0.286)	99.38
train	0.05	0.476 (28.56)	0.115 (90.16)	0.0018 (0.018)	99.95
test		0.471 (28.26)	0.118 (92.53)	0.0311 (0.311)	99.34

we observe that decreasing the value of beta below one is necessary to improve classification accuracy, as well as the overall variational lower bound. Taking the 60 dimensional case as a specific example, we can also observe that reducing the beta value too far and decreasing it from e.g. 0.1 to 0.05 leads to deterioration of the variational lower bound, from 119.596 to 121.101 natural units, while the classification accuracy by itself does not improve further.

C. Training Hyper-parameters and Architecture Definitions

We provide a full specification of hyper-parameters, model architectures and the training procedure used in the main body. We base our encoder and decoder architecture on 14-layer wide residual networks [6], [7] with a latent dimensionality

of 60 to demonstrate scalability to high-dimensions and as used in lossy auto-encoders [8], [9]. These architectures are shown in detail in tables 3 and 4. Hidden layers include batch-normalization [10] with a value of 10^{-5} and use ReLU activations. For a common frame of reference, all methods' share the same underlying WRN architecture, including the separate classifiers and generative models of the dual model approaches. Experiments with a simpler MLP architecture can be found in section E of the supplementary material. For the higher resolution 256×256 flower images, we have used a deeper 26 layer WRN version, in analogy to previous works [8], [9]. Here, the last encoder, and first decoder blocks are repeated an extra three times, resulting in an additional three stages of down- and up-sampling by factor two. The encoder's spatial output dimensionality is thus equivalent to the 14-layer architecture applied to the eight times lower resolution images of the simpler datasets. For the autoregressive addition to our joint model, we set the number of output channels of the decoder to 60 and append three additional pixel decoder layers, each with a kernel size of 7×7 and 60 channels. Whereas we report reconstruction log-likelihoods in nats, these models are practically formulated as a classification problem with a 256-way softmax. The corresponding loss is in bits per dimension. We have converted these values to have a better comparison, but in order to do so we need to sample from the pixel decoder's multinomial distribution to calculate a binary cross-entropy on reconstructed images. We further note that all losses are normalized with respect to spatial and latent dimensions, as mentioned in the main body.

We use hyper-parameters consistent with the literature [8], [9]. Accordingly, all models are optimized using stochastic gradient descent with a mini-batch size of 128 and Adam [11] with a learning rate of 0.001 and first and second momenta equal to 0.9 and 0.999. For MNIST, FashionMNIST and AudioMNIST no data augmentation or preprocessing is applied. For the flower experiments, images are stochastically flipped horizontally with a 50 % chance and the batch size is reduced to 32. We initialize all weights according to [12].

All class incremental models are trained for 120 epochs per task on MNIST and FashionMNIST and 150 epochs on AudioMNIST. Complementary incremental cross-dataset models are trained for 200 epochs per task on data resized to 32×32. While our proposed model exhibits forward transfer due to weight sharing and need not necessarily be trained for the entire amount of epochs for each subsequent task, this guarantees convergence and a fair comparison of results with respect to achievable accuracy of other methods. Isolated models are trained for 200 and 300 epochs until convergence respectively. Due to the much smaller dataset size, architectures are trained for 2000 epochs on the flower images, in order to obtain a similar amount of update steps. For the generative replay with statistical outlier rejection, we use an aggressive rejection rate of $\Omega_t = 0.01$ (with analogous results with 0.05) and dynamically set tail-sizes to 5% of seen examples per class. As mentioned in the main body, the used open set distance measure is the cosine distance.

For EWC, the number of Fisher samples is fixed to the total number of data points from all the previously seen tasks. A suitable Fisher multiplier value λ has been determined by conducting a grid search over a set of five values: 50, 100, 500, 1000 and 5000 on held-out validation data for the first two tasks in sequence. We observe exploding gradients if λ is too high. However, a very small λ leads to excessive drift in the weight distribution across subsequent tasks that further results in catastrophic interference. Empirically, $\lambda = 500$ in the class-incremental scenario and $\lambda = 1000$ in the cross-dataset setting seem to provide the best balance.

TABLE 3: 14-layer WRN encoder with a widen factor of 10. Convolutional layers (conv) are parametrized by a quadratic filter size followed by the amount of filters. p and s represent zero padding and stride respectively. If no padding or stride is specified then p = 0 and s = 1. Skip connections are an additional operation at a layer, with the layer to be skipped specified in brackets. Every convolutional layer is followed by batch-normalization and a ReLU activation function. The probabilistic encoder ends on fully-connected layers for μ and σ that depend on the chosen latent space dimensionality and the data's spatial size.

Layer type	WRN encoder	
Layer 1	conv 3 × 3 - 48, p = 1	
Block 1	conv 3 × 3 - 160, p = 1; conv 3 × 3 - 160, p = 1 conv 3 × 3 - 160, p = 1; conv 3 × 3 - 160, p = 1	conv 1 × 1 - 160 (skip next layer) shortcut (skip next layer)
Block 2	conv 3 × 3 - 320, s = 2, p = 1; conv 3 × 3 - 320, p = 1 conv 3 × 3 - 320, p = 1; conv 3 × 3 - 320, p = 1	conv 1 × 1 - 320, s = 2 (skip next layer) shortcut (skip next layer)
Block 3	conv 3 × 3 - 640, s = 2, p = 1; conv 3 × 3 - 640, p = 1 conv 3 × 3 - 640, p = 1; conv 3 × 3 - 640, p = 1	conv 1 × 1 - 640, s = 2 (skip next layer) shortcut (skip next layer)

TABLE 4: 14-layer WRN decoder with a widen factor of 10. P_w and P_h refer to the input's spatial dimension. Convolutional (conv) and transposed convolutional (conv_t) layers are parametrized by a quadratic filter size followed by the amount of filters. p and s represent zero padding and stride respectively. If no padding or stride is specified then p = 0 and s = 1. Skip connections are an additional operation at a layer, with the layer to be skipped specified in brackets. Every convolutional and fully-connected (FC) layer are followed by batch-normalization and a ReLU activation function. The model ends on a Sigmoid function.

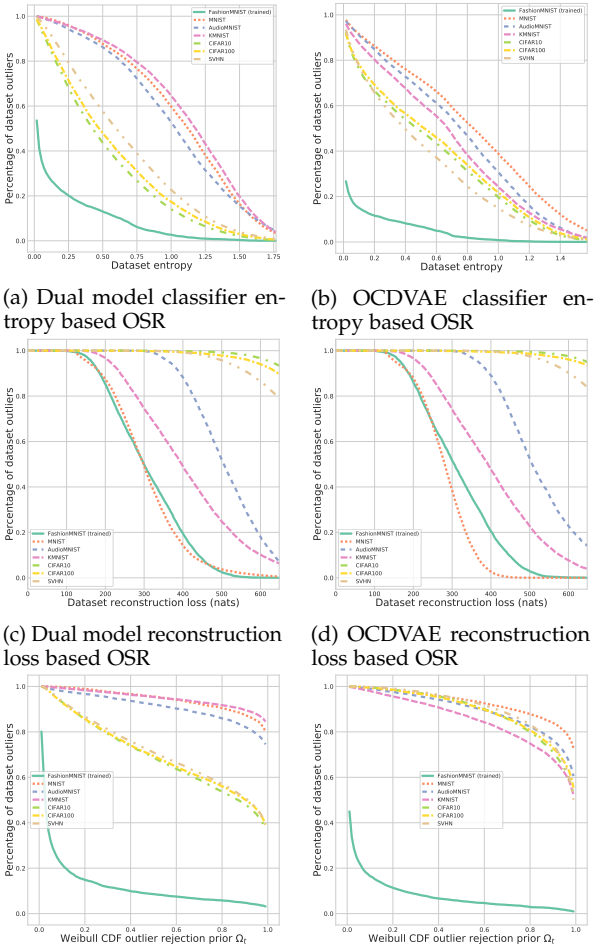
Layer type	WRN decoder	
Layer 1	FC $640 \times \lfloor P_w/4 \rfloor \times \lfloor P_h/4 \rfloor$	
Block 1	conv_t 3 × 3 - 320, p = 1; conv 3 × 3 - 320, p = 1 conv 3 × 3 - 320, p = 1; conv 3 × 3 - 320, p = 1 upsample × 2	conv_t 1 × 1 - 320 (skip next layer) shortcut (skip next layer)
Block 2	conv_t 3 × 3 - 160, p = 1; conv 3 × 3 - 160, p = 1 conv 3 × 3 - 160, p = 1; conv 3 × 3 - 160, p = 1 upsample × 2	conv_t 1 × 1 - 160 (skip next layer) shortcut (skip next layer)
Block 3	conv_t 3 × 3 - 48, p = 1; conv 3 × 3 - 48, p = 1 conv 3 × 3 - 48, p = 1; conv 3 × 3 - 48, p = 1	conv_t 1 × 1 - 48 (skip next layer) shortcut (skip next layer)
Layer 2	conv 3 × 3 - 3, p = 1	

D. Additional Open Set Recognition Visualization

As we point out in section 4 of the main paper, our posterior based open set recognition considers almost all of the unknown datasets as statistical outliers, while at the same time regarding unseen test data from the originally trained tasks as distribution inliers across a wide range of rejection priors. In addition to the outlier rejection curves for FashionMNIST and the quantitative results presented in the main body, we also show the full outlier rejection curves for the remaining datasets, as well as all dual model approaches in figures 2, 3 and 4. These figures visually support the quantitative findings described in the main body and respective conclusions. In summary, the joint OCDVAE performs better at open set recognition in direct comparison to the dual model setting, particularly when using the EVT based criterion. Apart from the MNIST dataset, where reconstruction loss can be a sufficient metric for open set detection, the latent based approach also exhibits less dependency on the outlier rejection prior and consistently improves the ability to discern unknown data.

Monte Carlo Dropout

In this subsection we provide additional quantitative results for open set recognition with Monte-Carlo Dropout (MCD) in order to assess the effectiveness of approximating a distribution on the weights to estimate uncertainty, in addition to the experiments of the main body where the latent variable is marginalised. We have therefore re-trained all of the models reported in table **??** with a Dropout probability of 0.2 in each layer. We then conduct 50 stochastic forward passes through the entire model for prediction. The obtained open set recognition results are reported in 5. Although MCD boosts the outlier detection accuracy, particularly for criteria such as predictive entropy, the insights of the main body still hold. In summary, the joint model generally outperforms a purely discriminative model in terms of open set recognition, independently of the used metric, and our proposed aggregate posterior based EVT approach of the OCDVAE yields an almost perfect separation of known and unseen unknown data. Interestingly, this has already been achieved in the experiments of the main body. Resorting to the repeated model calculation of MCD thus seems to come without enough of an advantage to warrant the added computational complexity in the context of posterior based open set recognition.



(a) Dual model classifier entropy based OSR

(b) OCDVAE classifier entropy based OSR

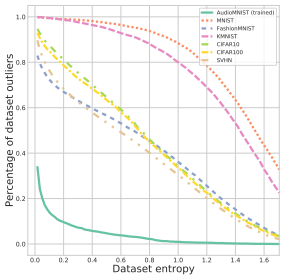
(c) Dual model reconstruction loss based OSR

(d) OCDVAE reconstruction loss based OSR

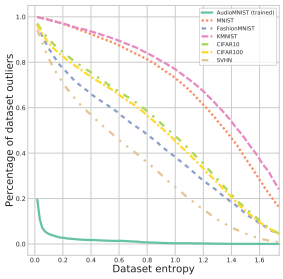
(e) Dual model posterior EVT based OSR

(f) OCDVAE posterior EVT based OSR

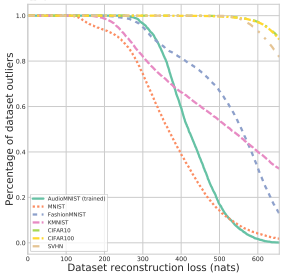
Fig. 2: Dual model and OCDVAE trained on FashionMNIST evaluated on unseen datasets. Pairs of panels show the contrast between the approaches. Left panels correspond to the dual model, right panels show the joint OCDVAE model. (a+b) The classifier entropy values by itself are insufficient to separate most of unknown from the known task's test data. (c+d) Reconstruction loss allows for a partial distinction. (e+f) Our posterior-based open set recognition considers the large majority of unknown data as statistical outliers across a wide range of rejection priors Ω_t, significantly more so in the OCDVAE model. All metrics are reported as the mean over 100 approximate posterior samples per data point.



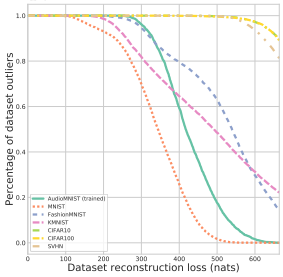
(a) Dual model classifier entropy based OSR



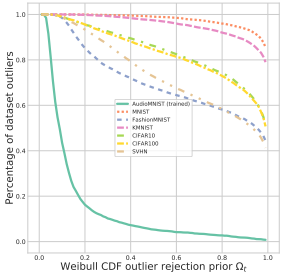
(b) OCDVAE classifier entropy based OSR



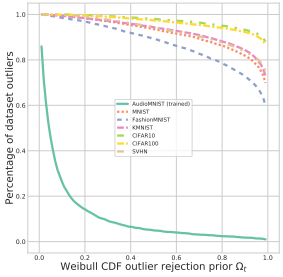
(c) Dual model reconstruction loss based OSR



(d) OCDVAE reconstruction loss based OSR

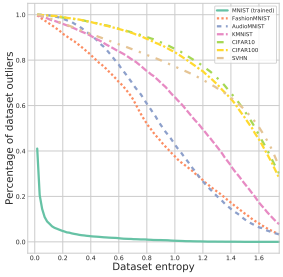


(e) Dual model posterior EVT based OSR

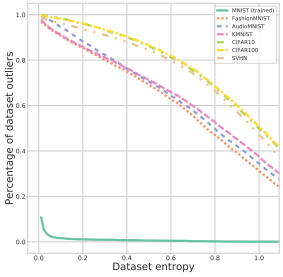


(f) OCDVAE posterior EVT based OSR

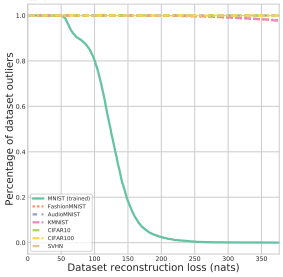
Fig. 3: Dual model and OCDVAE trained on AudioMNIST evaluated on unseen datasets. Pairs of panels show the contrast between the approaches. Left panels correspond to the dual model, right panels show the joint OCDVAE model. (a+b) The classifier entropy values by itself are insufficient to separate most of unknown from the known task's test data. (c+d) Reconstruction loss allows for a partial distinction. (e+f) Our posterior-based open set recognition considers the large majority of unknown data as statistical outliers across a wide range of rejection priors Ω_t, significantly more so in the OCDVAE model. All metrics are reported as the mean over 100 approximate posterior samples per data point.



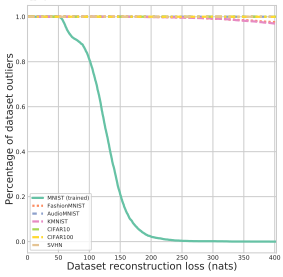
(a) Dual model classifier entropy based OSR



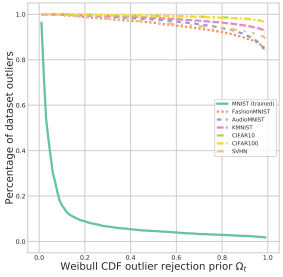
(b) OCDVAE classifier entropy based OSR



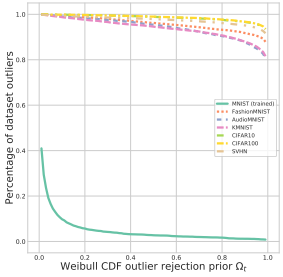
(c) Dual model reconstruction loss based OSR



(d) OCDVAE reconstruction loss based OSR



(e) Dual model posterior EVT based OSR



(f) OCDVAE posterior EVT based OSR

Fig. 4: Dual model and OCDVAE trained on MNIST evaluated on unseen datasets. Pairs of panels show the contrast between the approaches. Left panels correspond to the dual model, right panels show the joint OCDVAE model. (a+b) The classifier entropy values by itself can achieve a partial separation between unknown and the known task's test data. (c+d) Reconstruction loss allows for distinction if the cut-off is chosen correctly. (e+f) Our posterior-based open set recognition considers the large majority of unknown data as statistical outliers across a wide range of rejection priors Ω_t. All metrics are reported as the mean over 100 approximate posterior samples per data point. While the OCDVAE shows improvement upon the dual model approach, particularly if using classifier entropies for OSR, both models trained on MNIST perform well in OSR. In direct contrast with models trained on Fashion- or AudioMNIST and respective figures 2 and 3, this shows that evaluation on MNIST alone is generally insufficient.

TABLE 5: Test accuracies and outlier detection values of the joint OCDVAE and dual model (VAE and separate deep classifier) approaches when considering 95 % of known tasks' validation data is inlying. Percentage of detected outliers is reported based on classifier predictive entropy, reconstruction loss and our posterior based EVT approach, averaged over 50 Monte Carlo dropout samples, with $p_{dropout} = 0.2$ for each layer, per data-point respectively. Note that larger values are better, except for the test data of the trained dataset, where ideally 0% should be considered as outlying.

Outlier detection at 95% validation inliers (%)				MNIST	Fashion	Audio	KMNIST	CIFAR10	CIFAR100	SVHN
Trained	Model	Test acc.	Criterion							
FashionMNIST	Dual, CL + VAE	90.58	Class entropy	75.50	5.366	70.78	74.41	49.42	49.17	38.84
			Reconstruction	55.45	5.048	59.99	99.83	99.35	99.35	99.62
			Latent EVT	77.03	4.920	55.48	70.23	58.73	57.06	44.54
	Joint, OCDVAE	91.50	Class Entropy	85.05	4.740	67.90	78.04	63.89	66.11	59.42
			Reconstruction	1.227	5.422	85.85	39.76	99.94	99.72	99.99
			Latent EVT	95.83	4.516	94.56	96.04	96.81	96.66	96.28
MNIST	Dual, CL + VAE	99.41	Class entropy	4.276	91.88	96.50	96.65	95.84	97.37	98.58
			Reconstruction	4.829	99.99	100.0	99.90	100.0	100.0	100.0
			Latent EVT	4.088	87.84	98.06	95.79	97.34	98.30	95.74
	Joint, OCDVAE	99.54	Class entropy	4.801	97.63	99.38	98.01	99.16	99.39	98.90
			Reconstruction	5.264	99.98	100.0	100.0	100.0	100.0	100.0
			Latent EVT	4.978	99.99	100.0	99.94	99.96	99.95	99.68
AudioMNIST	Dual, CL + VAE	98.76	Class entropy	99.97	61.26	4.996	96.77	63.78	65.76	59.38
			Reconstruction	7.334	52.37	5.100	98.19	99.97	99.90	99.96
			Latent EVT	92.74	67.18	5.073	90.41	90.56	90.97	89.58
	Joint, OCDVAE	98.85	Class entropy	99.39	89.50	5.333	99.16	94.66	95.12	97.13
			Reconstruction	15.81	53.83	4.837	41.89	99.90	99.82	99.95
			Latent EVT	99.50	99.27	5.136	99.75	99.71	99.59	99.91

TABLE 6: Results for continual learning across datasets averaged over 5 runs, baselines and the reference isolated learning scenario for FashionMNIST (F) $\rightarrow$ MNIST (M) $\rightarrow$ AudioMNIST (A) and the reverse order. α_T indicates the respective accuracy at the end of the last increment $T = 3$.

Cross-dataset		α_T(%) (T=3)					
		base		new		all	
		MLP	WRN	MLP	WRN	MLP	WRN
F-M-A	**CDVAE ISO**					93.86	94.95
	CDVAE UB	89.75	89.10	97.28	97.88	93.94	95.00
	CDVAE LB	00.00	00.00	97.38	98.12	22.51	22.70
	EWC	42.10 $\pm$ 1.880	22.85 $\pm$ 0.294	31.33 $\pm$ 2.037	93.31 $\pm$ 0.138	46.04 $\pm$ 1.195	43.42 $\pm$ 0.063
	Dual Model	81.12 $\pm$ 0.341	81.89 $\pm$ 0.104	97.15 $\pm$ 0.320	96.78 $\pm$ 0.067	91.03 $\pm$ 0.096	91.75 $\pm$ 0.064
	CDVAE	74.23 $\pm$ 0.587	57.70 $\pm$ 4.480	97.04 $\pm$ 0.105	96.73 $\pm$ 0.235	85.55 $\pm$ 0.234	81.10 $\pm$ 1.769
	OCDVAE	79.01 $\pm$ 0.591	80.11 $\pm$ 2.922	97.34 $\pm$ 0.152	97.63 $\pm$ 0.042	89.87 $\pm$ 0.262	91.13 $\pm$ 1.045
A-M-F	**CDVAE ISO**					93.67	94.95
	CDVAE UB	96.97	97.17	89.34	89.16	93.75	94.91
	CDVAE LB	00.00	00.00	89.81	89.72	34.55	34.51
	EWC	7.178 $\pm$ 2.432	3.420 $\pm$ 0.026	73.83 $\pm$ 2.873	87.54 $\pm$ 0.214	46.37 $\pm$ 1.908	45.42 $\pm$ 0.731
	Dual Model	51.70 $\pm$ 2.611	66.82 $\pm$ 0.337	89.53 $\pm$ 0.093	89.15 $\pm$ 0.050	83.95 $\pm$ 0.644	87.70 $\pm$ 0.102
	CDVAE	65.38 $\pm$ 2.501	79.74 $\pm$ 2.431	89.30 $\pm$ 0.116	88.50 $\pm$ 0.126	86.19 $\pm$ 0.584	89.46 $\pm$ 0.600
	OCDVAE	81.65 $\pm$ 1.414	94.53 $\pm$ 0.283	89.31 $\pm$ 0.109	89.53 $\pm$ 0.367	90.08 $\pm$ 0.471	94.06 $\pm$ 0.156

TABLE 7: Results for class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for the three datasets. α_T indicates the respective accuracy at the end of the last increment $T = 5$.

Class-incremental		α_T(%) (T=5)					
		base		new		all	
		MLP	WRN	MLP	WRN	MLP	WRN
FashionMNIST	**CDVAE ISO**					87.68	89.54
	CDVAE UB	91.10	92.20	96.75	97.50	87.35	89.24
	CDVAE LB	00.00	00.00	99.75	99.80	19.95	19.97
	EWC	21.79 ± 2.610	00.17 ± 0.076	96.80 ± 0.873	99.60 ± 0.023	24.48 ± 2.862	20.06 ± 0.059
	Dual Model	91.64 ± 1.233	94.26 ± 0.192	97.18 ± 0.171	93.55 ± 0.708	68.49 ± 2.110	63.21 ± 1.957
	CDVAE	49.71 ± 1.363	39.51 ± 7.173	97.84 ± 0.375	96.92 ± 0.774	62.72 ± 1.379	58.82 ± 2.521
	OCDVAE	56.67 ± 2.279	60.63 ± 12.16	97.89 ± 0.332	96.51 ± 0.707	66.14 ± 0.497	69.88 ± 1.712
MNIST	**CDVAE ISO**					98.87	99.45
	CDVAE UB	99.57	99.57	98.04	99.10	98.84	99.29
	CDVAE LB	00.00	00.00	99.75	99.85	19.92	20.16
	EWC	24.08 ± 0.487	00.45 ± 0.059	96.70 ± 2.039	99.58 ± 0.052	26.46 ± 2.351	20.26 ± 0.027
	Dual Model	92.63 ± 1.609	97.31 ± 0.489	98.48 ± 0.145	98.59 ± 0.106	89.74 ± 0.726	96.64 ± 0.079
	CDVAE	34.48 ± 9.512	19.86 ± 7.396	98.84 ± 0.228	99.00 ± 0.100	60.88 ± 3.308	64.34 ± 4.903
	OCDVAE	82.54 ± 2.26	92.35 ± 4.485	98.89 ± 0.151	99.06 ± 0.171	87.31 ± 1.224	93.24 ± 3.742
AudioMNIST	**CDVAE ISO**					96.33	97.75
	CDVAE UB	99.08	98.42	98.25	98.67	96.43	97.87
	CDVAE LB	00.00	00.00	99.92	100.0	20.03	20.02
	EWC	17.51 ± 3.380	00.11 ± 0.007	85.25 ± 4.209	99.41 ± 0.207	20.48 ± 1.727	19.98 ± 0.032
	Dual Model	53.60 ± 0.586	61.58 ± 0.747	97.22 ± 0.559	89.41 ± 0.691	48.42 ± 2.808	47.42 ± 1.447
	CDVAE	20.76 ± 5.521	59.36 ± 7.147	89.21 ± 0.402	84.93 ± 6.297	69.76 ± 1.369	81.49 ± 1.944
	OCDVAE	56.68 ± 5.059	79.73 ± 4.070	89.35 ± 0.244	89.52 ± 6.586	81.84 ± 1.438	87.72 ± 1.594

E. MLP based Continual Learning

For comparably simple datasets such as MNIST, it could be argued that optimizing a deep WRN decoder for generative replay is more expensive than simply storing the entire original MNIST dataset for continued classifier training. In the main body we have used this WRN architecture to provide a common frame of reference across all experiments. To nevertheless demonstrate that such a complex network is not essential for continual learning of simple datasets, we repeat all MNIST, FashionMNIST and AudioMNIST with a shallow MLP architecture of limited representational capacity. To allow for a direct comparison with the WRN based results, we use the same latent dimensionality of 60 and similarly let all the other hyper-parameters remain the same. However, we replace the deep encoder and decoder with two fully-connected hidden layers of 400 units [13]. The corresponding quantitative results for cross-dataset and class incremental learning are reported in tables 6 and 7 respectively. The assuring main observation is that the MLP models fare only marginally worse, with the biggest difference to the WRN being perceivable on the audio dataset. However, the relative ranking of individual methods remains the same in almost all cases and the general insight and conclusions of the main body prevail. The only exception is the use of EWC in conjunction with the shallow MLP. With a lambda value of 500, we find EWC in an MLP to work significantly better than in application to the deep counterpart, in particular in initial task increments. Although the approach still faces difficulty with a growing single-head classifier, see the discussion in section 4 of the main body, and is still by far the worst in a global comparison, it no longer directly mirrors the lower bound accuracy. We hypothesize that this is due to a more informative and accurate estimate of important parameters in the presence of only two layers with significantly less units.

F. Detailed Results for the MNIST, Fashion-MNIST and AudioMNIST Experiments

In the main body we have reported three metrics for our continual learning experiments based on classification accuracy: the base task's accuracy over time $\alpha_{t,base}$, the new task's accuracy $\alpha_{t,new}$ and the overall accuracy at any point in time $\alpha_{t,all}$. This is an appropriate measure to evaluate the quality of the generative model over time given that the employed mechanism to avoid catastrophic interference in continual learning is generative replay. On the one hand, if catastrophic interference occurs in the decoder the sampled data will no longer resemble the instances of the observed data distribution. This will in turn degrade the encoder during continued training and thus the classification accuracy. On the other hand, this proxy measure for the generation quality avoids the common pitfalls of pixel-wise reconstruction metrics. The information necessary to maintain respective knowledge of the data distribution through the variational approximation in the probabilistic encoder does not necessarily rely on correctly reconstructing data's local information. To take an example, if a model were to reconstruct all images perfectly but with some degree of spatial translation or rotation, then the negative log likelihood (NLL) would arguable be worse than that of a model which reconstructs local details correctly on a pixel level for a fraction of the image. As this could be details in e.g. the background or other class unspecific areas, training on corresponding generations does not have to prevent loss of encoder knowledge with respect to the classification task.

As such, a similar argument can be conjured for the KL divergence. On the one hand, monitoring the KL divergence as a regularization term by itself over the course of continual learning is meaningless without regarding the data's NLL. On the other hand, for our OCDVAE model the exact value of the KL divergence does not immediately reflect the quality of the generated data. This is because we do not sample merely from the prior, but as explained in the main body employ a rejection mechanism to draw samples that belong to the aggregate posterior.

Nevertheless, for the purpose of completeness and in addition to the results provided in the experimental section of the main body, we provide the reconstruction losses and KL divergences for all applicable models in this supplementary material section. Analogous to the three metrics for classification accuracy of base, new and all tasks, we define the respective reconstruction losses $\gamma_{t,base}, \gamma_{t,new}$ and $\gamma_{t,all}$. The KL divergence KL_t always measures the deviation from the prior $p(z)$ at any point in time, as the prior remains the same throughout continual training. Following the above discussion, we argue that these values should be regarded with caution and should not be interpreted separately.

Full Cross Dataset Results

We show the full cross dataset results in table 8 in extension to table 1 in the main body. An analogous table for the presented autoregressive models can be found in table 9. Similar to the accuracy values, we can observe that the mismatch between aggregate posterior and prior as expressed through the KL divergence is greater in a naive joint model (naive CDVAE) in comparison to a dual model approach with separate generative and discriminative models. Our proposed OCDVAE model, with respective rejection sampling scheme that takes into account the structure of the aggregate posterior, alleviates this to a large degree. The reconstruction losses of both the dual model and the joint OCDVAE approach show only negligible deviation with respect to the achievable upper bound and only limited catastrophic interference of the decoder occurs. However, we can also observe that by itself these quantities are not indicative of maintaining encoder knowledge with respect to representations required for classification. This is particularly visible in the tables' second experiment, where we first train Audio data and then proceed with the two image datasets. Here, the KL divergence and reconstruction loss are both better for the dual model, whereas a much higher accuracy over time is maintained in the OCDVAE model. Naturally, this is because a significant mismatch between aggregate posterior and prior is also present in a purely unsupervised generative model and naively sampling from the prior will result in generated instances that do not resemble those present in the observed data distribution. While weaker in effect, this is similar to the naive CDVAE approach. Without the presence of the linear discriminator on the latents in the purely unsupervised generative model, there is however no

straightforward mechanism to disentangle the latent space according to classes. Our proposed open set approach and the resulting constraint to samples from the aggregate posterior as presented in the OCDVAE is thus not trivially applicable.

Full Class Incremental Results

In addition to reconstruction losses and KL divergences, we also report the detailed full set of intermediate results for the five task steps of the class incremental scenario. We thus extend table 3 in the main body with results for all task increments $t = 1, \ldots, 5$ and a complete list of losses in tables 10, 11 and 12 for the three datasets respectively. The corresponding results for autoregressive models are presented in tables 13, 14 and 15.

Once more, we can observe the increased effect of error accumulation due to unconstrained generative sampling from the prior in comparison to the open set counterpart that limits sampling to the aggregate posterior. The statistical deviations across experiment repetitions in the base and the overall classification accuracies are higher and are generally decreased by the open set models. For example, in table 10 the MNIST base and overall accuracy deviations of a naive CDVAE are higher than the respective values for OCDVAE starting already from the second task increment. Correspondingly, the accuracy values themselves experience larger decline for CDVAE than for OCDVAE with progressive increments. This difference is not as pronounced at the end of the first task increment because the models haven't been trained on any of their own generated data yet. Successful literature approaches such as the variational generative replay proposed by [14] thus avoid repeated learning based on previous generated examples and simply store and retain a separate generative model for each task. The strength of our model is that, instead of storing a trained model for each task increment, we are able to continually keep training our joint model with data generated for all previously seen tasks by filtering out ambiguous samples from low density areas of the posterior. Similar trends can also be observed for the respective pixel models.

We also see that regularization approaches such as EWC already fail at the first increment. In contrast to the success that has been reported in prior literature [13], [15], this is due to the use of a single classification head. This is intuitive because introduction of new units, as described in the main body, directly confuses the existing classification. Regularization approaches by definition are challenged in this scenario because the weights are not allowed to drift too far away from previous values. For emphasis we repeat that however this scenario is much more practical and realistic than a multi-head scenario with a separate classifier per task. While regularization approaches are largely successful in the latter setting, it is not only restricted to the closed world, but further requires an oracle at prediction stage to chose the correct classification head. In contrast, our proposed approach requires no knowledge of task labels for prediction and is robust in an open world.

With respect to KL divergences and reconstruction losses we can make two observations. First, the arguments of the previous section hold and by itself the small relative improvements between models should be interpreted with caution as they do not directly translate to maintaining continual learning accuracy. Second, we can also observe that reconstruction losses at every increment for all $\gamma_{t,all}$ and respective negative log likelihoods for only the new task $\gamma_{t,new}$ are harder to interpret than the accuracy counterpart. While the latter is normalized between zero and unity, the reconstruction loss of different tasks is expected to fluctuate largely according to the task's images' reconstruction complexity. To give a concrete example, it is rather straightforward to come to the conclusion that a model suffers from limited capacity or lack of complexity if a single newly arriving class cannot be classified well. In the case of reconstruction it is common to observe either a large decrease in negative log likelihood for the newly arriving class, or a big increase depending on the specific introduced class. As such, these values are naturally comparable between models, but are challenging to interpret across time steps without also analyzing the underlying nature of the introduced class. The exception is formed by the base task's reconstruction loss $\gamma_{t,base}$. In analogy to base classification accuracy, this quantity still measures the amount of catastrophic forgetting across time. However, in all tables we can observe that catastrophic forgetting of the decoder as measured by the base reconstruction loss is almost imperceivable. As this is not at all reflected in the respective accuracy over time, it further underlines our previous arguments that reconstruction loss is not necessarily the best metric to monitor in the presented continual learning scenario.

G. Generative Replay Examples with CDVAE and OCDVAE

In this section we provide visualization of data instances that are produced during generative replay at the end of each task increment. In particular, we qualitative illustrate the effect of constraining sampling to the aggregate posterior in contrast to naively sampling from the prior without statistical outlier rejection for low density regions. Figures 5, 6 and 7 illustrate generated images for MNIST, FashionMNIST and AudioMNIST respectively. For both a naive CDVAE as well as the autoregressive PixCDVAE we observe significant confusion with respect to classes. As the generative model needs to learn how to replay old tasks' data based on its own former generations, ambiguity and blurry interpolations accumulate and are rapidly amplified. This is not the case for OCDVAE and PixOCDVAE, where the generative model is capable of maintaining higher visual fidelity throughout continual training and misclassification is scarce.

TABLE 8: Results for incremental cross-dataset continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for FashionMNIST (F) $\rightarrow$ MNIST (M) $\rightarrow$ AudioMNIST (A) and the reverse order. Extension of table 1 in the main body. Here, in addition to the accuracy α_T, γ_T and KL_T also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of the last increment $T = 3$.

Cross-dataset		α_T(%)			γ_T(nats)			KL_T(nats)
		base	new	all	base	new	all	all
F-M-A	**CDVAE ISO**			94.95			269.6	24.97
	CDVAE UB	89.10	97.88	95.00	311.2	434.3	269.7	25.20
	CDVAE LB	00.00	98.12	22.70	689.7	341.0	511.7	98.74
	EWC	22.85 $\pm$ 0.294	93.31 $\pm$ 0.138	43.42 $\pm$ 0.063				
	Dual Model	81.89 $\pm$ 0.104	96.78 $\pm$ 0.067	91.75 $\pm$ 0.064	320.0 $\pm$ 1.275	431.1 $\pm$ 1.474	273.7 $\pm$ 1.174	12.80 $\pm$ 0.060
	CDVAE	57.70 $\pm$ 4.480	96.73 $\pm$ 0.235	81.10 $\pm$ 1.769	360.9 $\pm$ 20.15	432.1 $\pm$ 0.231	296.4 $\pm$ 7.966	44.29 $\pm$ 4.047
	OCDVAE	80.11 $\pm$ 2.922	97.63 $\pm$ 0.042	91.13 $\pm$ 1.045	345.1 $\pm$ 7.446	430.7 $\pm$ 0.600	280.2 $\pm$ 1.069	25.42 $\pm$ 1.876
A-M-F	**CDVAE ISO**			94.95			269.6	24.97
	CDVAE UB	97.17	89.16	94.91	428.8	311.9	268.2	23.91
	CDVAE LB	00.00	89.72	34.51	506.6	311.0	351.1	34.13
	EWC	3.420 $\pm$ 0.026	87.54 $\pm$ 0.214	45.42 $\pm$ 0.731				
	Dual Model	66.82 $\pm$ 0.337	89.15 $\pm$ 0.050	87.70 $\pm$ 0.102	447.3 $\pm$ 6.700	308.5 $\pm$ 0.599	270.9 $\pm$ 1.299	12.89 $\pm$ 0.109
	CDVAE	79.74 $\pm$ 2.431	88.50 $\pm$ 0.126	89.46 $\pm$ 0.600	448.6 $\pm$ 5.187	315.1 $\pm$ 1.305	281.6 $\pm$ 3.205	33.38 $\pm$ 0.898
	OCDVAE	94.53 $\pm$ 0.283	89.53 $\pm$ 0.367	94.06 $\pm$ 0.156	433.4 $\pm$ 0.424	311.6 $\pm$ 0.353	271.2 $\pm$ 0.424	23.16 $\pm$ 0.121

TABLE 9: Results for PixelVAE based cross-dataset continual learning approaches averaged over 5 runs in analogy to table 8. Extension of table 4 in the main body. Here, in addition to the accuracy α_T, γ_T and KL_T also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of the last increment $T = 3$.

Cross-dataset		α_T(%)			γ_T(nats)			KL_T(nats)
		base	new	all	base	new	all	all
F-M-A	**Dual Pix Model**	82.88 $\pm$ 0.116	97.23 $\pm$ 0.212	92.16 $\pm$ 0.061	288.5 $\pm$ 0.723	437.7 $\pm$ 0.404	251.6 $\pm$ 0.231	9.025 $\pm$ 1.378
	PixCDVAE	56.44 $\pm$ 1.831	97.50 $\pm$ 0.184	80.76 $\pm$ 0.842	289.8 $\pm$ 1.283	438.1 $\pm$ 0.990	252.6 $\pm$ 1.424	29.99 $\pm$ 0.629
	PixOCDVAE	81.84 $\pm$ 0.212	97.75 $\pm$ 0.169	91.76 $\pm$ 0.212	288.8 $\pm$ 0.141	437.1 $\pm$ 0.725	251.8 $\pm$ 0.636	21.07 $\pm$ 0.248
A-M-F	**Dual Pix Model**	71.58 $\pm$ 2.536	88.76 $\pm$ 0.255	88.61 $\pm$ 0.547	445.8 $\pm$ 1.601	290.4 $\pm$ 0.603	255.0 $\pm$ 0.533	9.164 $\pm$ 1.312
	PixCDVAE	49.38 $\pm$ 2.256	88.54 $\pm$ 0.042	82.18 $\pm$ 0.672	441.4 $\pm$ 0.495	287.0 $\pm$ 0.212	252.5 $\pm$ 0.201	30.60 $\pm$ 1.556
	PixOCDVAE	91.90 $\pm$ 0.282	89.91 $\pm$ 0.177	93.82 $\pm$ 0.354	438.5 $\pm$ 1.626	289.4 $\pm$ 0.356	251.3 $\pm$ 0.354	20.35 $\pm$ 0.424

TABLE 10: Results for class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for MNIST at the end of every task increment. Extension of table 3 in the main body. Here, in addition to the accuracy α_t, γ_t and KL_t also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of every task increment t.

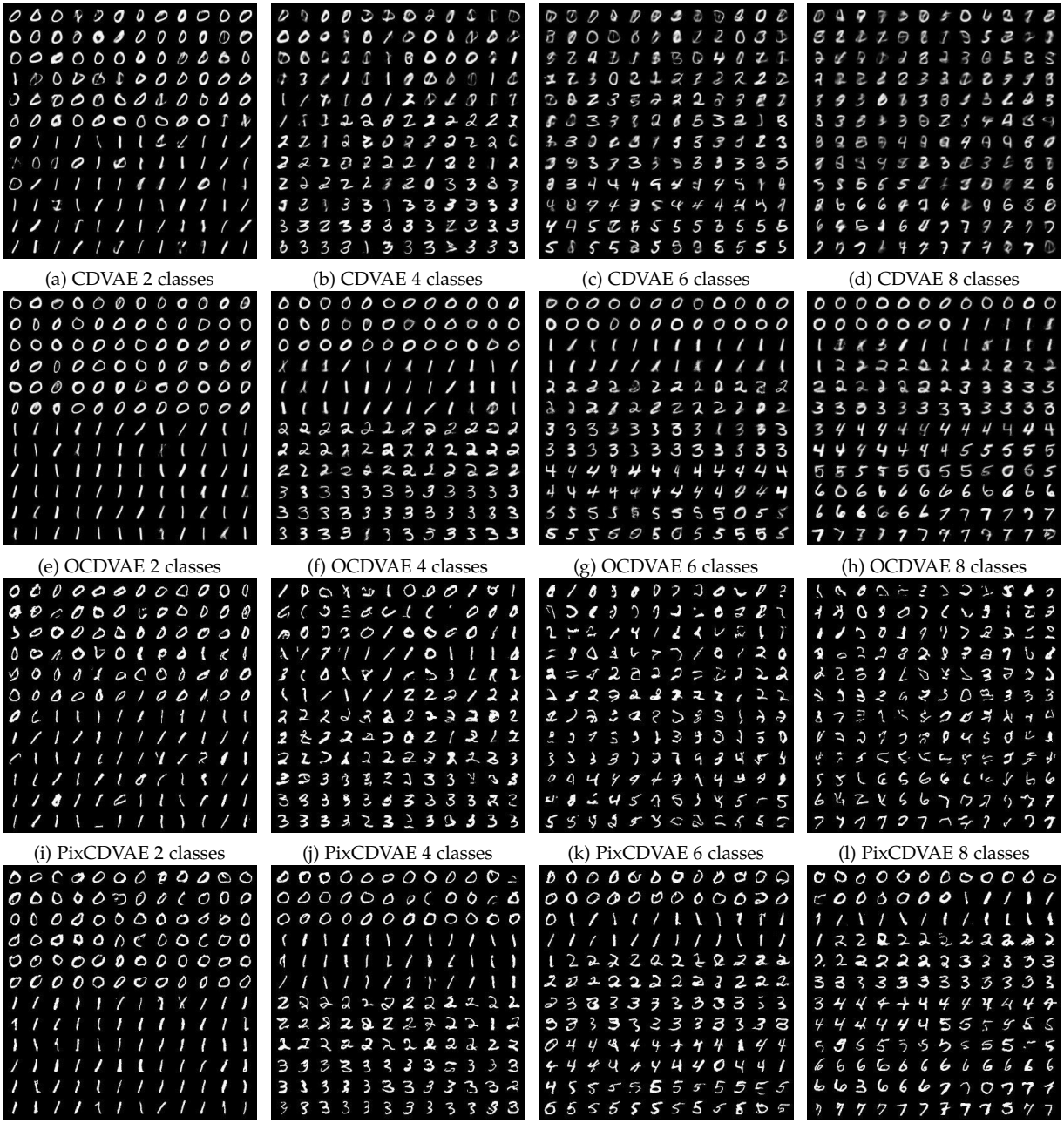
MNIST	t	CDVAE ISO	CDVAE UB	CDVAE LB	EWC	Dual Model	CDVAE	OCDVAE
$\boldsymbol{\alpha_{base,t}}$ (%)	1		100.0	100.0	99.88 ± 0.010	99.98 ± 0.023	99.97 ± 0.029	99.98 ± 0.018
	2		99.82	00.00	00.61 ± 0.057	99.77 ± 0.032	97.28 ± 3.184	99.30 ± 0.100
	3		99.80	00.00	00.17 ± 0.045	99.51 ± 0.094	87.66 ± 8.765	96.69 ± 2.173
	4		99.85	00.00	00.49 ± 0.017	98.90 ± 0.207	54.70 ± 22.84	94.71 ± 1.792
	5		99.57	00.00	00.45 ± 0.059	97.31 ± 0.489	19.86 ± 7.396	92.53 ± 4.485
$\boldsymbol{\alpha_{new,t}}$ (%)	1		100.0	100.0	99.88 ± 0.010	99.98 ± 0.023	99.97 ± 0.029	99.98 ± 0.018
	2		99.80	99.85	99.70 ± 0.013	99.81 ± 0.062	99.75 ± 0.127	99.80 ± 0.126
	3		99.67	99.94	99.94 ± 0.002	99.48 ± 0.294	99.63 ± 0.172	99.61 ± 0.055
	4		99.49	100.0	99.87 ± 0.015	99.46 ± 0.315	99.05 ± 0.470	99.15 ± 0.032
	5		99.10	99.86	99.58 ± 0.052	98.59 ± 0.106	99.00 ± 0.100	99.06 ± 0.171
$\boldsymbol{\alpha_{all,t}}$ (%)	1		100.0	100.0	99.88 ± 0.010	99.98 ± 0.023	99.97 ± 0.029	99.98 ± 0.018
	2		99.81	49.92	50.16 ± 0.029	99.79 ± 0.049	98.54 ± 1.638	99.55 ± 0.036
	3		99.72	31.35	33.42 ± 0.027	99.32 ± 0.057	95.01 ± 3.162	98.46 ± 0.903
	4		99.50	24.82	25.36 ± 0.025	98.56 ± 0.021	81.50 ± 9.369	97.06 ± 1.069
	5	99.45	99.29	20.16	20.26 ± 0.027	96.64 ± 0.079	64.34 ± 4.903	93.24 ± 3.742
$\boldsymbol{\gamma_{base,t}}$ (nats)	1		63.18	62.08		62.17 ± 0.979	64.34 ± 2.054	62.53 ± 1.166
	2		62.85	126.8		63.69 ± 0.576	74.41 ± 10.89	65.68 ± 1.166
	3		63.36	160.4		67.34 ± 0.445	81.89 ± 10.09	69.29 ± 1.541
	4		64.25	126.9		70.41 ± 0.436	90.62 ± 10.08	71.69 ± 1.379
	5		64.99	123.2		75.08 ± 0.623	101.6 ± 8.347	77.16 ± 1.104
$\boldsymbol{\gamma_{new,t}}$ (nats)	1		63.18	62.08		62.17 ± 0.979	64.34 ± 2.054	62.53 ± 1.166
	2		88.75	87.93		88.03 ± 0.664	89.91 ± 0.107	89.64 ± 3.709
	3		82.53	87.22		83.46 ± 0.992	87.65 ± 0.530	85.37 ± 1.725
	4		72.68	74.61		73.23 ± 0.280	79.49 ± 0.489	74.75 ± 0.777
	5		85.88	92.00		89.32 ± 0.626	93.55 ± 0.391	89.68 ± 0.618
$\boldsymbol{\gamma_{all,t}}$ (nats)	1		63.18	62.08		62.17 ± 0.979	64.34 ± 2.054	62.53 ± 1.166
	2		75.97	107.3		75.64 ± 0.600	82.02 ± 5.488	76.62 ± 1.695
	3		79.58	172.3		81.24 ± 0.262	89.88 ± 3.172	82.95 ± 1.878
	4		79.72	203.1		82.92 ± 0.489	95.83 ± 2.747	85.30 ± 1.524
	5	78.12	81.97	163.7		88.29 ± 0.363	107.6 ± 1.724	92.92 ± 2.283
$\boldsymbol{KL_{all,t}}$ (nats)	1		12.55	13.08		11.81 ± 0.123	13.00 ± 0.897	13.68 ± 0.785
	2		18.50	25.84		16.15 ± 0.149	20.20 ± 1.188	18.01 ± 0.154
	3		20.16	24.28		16.46 ± 0.122	24.24 ± 1.974	20.02 ± 0.161
	4		20.48	26.32		16.09 ± 0.177	27.01 ± 1.851	20.26 ± 0.186
	5	22.12	21.02	24.87		16.13 ± 0.225	30.61 ± 1.240	21.02 ± 0.717

TABLE 11: Results for class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for FashionMNIST at the end of every task increment. Extension of table 3 in the main body. Here, in addition to the accuracy α_t, γ_t and KL_t also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of every task increment t.

Fashion	t	CDVAE ISO	CDVAE UB	CDVAE LB	EWC	Dual Model	CDVAE	OCDVAE
$\boldsymbol{\alpha_{base,t}}$ (%)	1		99.65	99.60	99.17 ± 0.037	99.58 ± 0.062	99.55 ± 0.035	99.59 ± 0.082
	2		96.70	00.00	02.40 ± 0.122	94.50 ± 0.389	92.02 ± 1.175	92.36 ± 2.092
	3		95.95	00.00	01.63 ± 0.032	94.88 ± 0.432	79.26 ± 4.170	83.90 ± 2.310
	4		91.35	00.00	00.33 ± 0.097	82.25 ± 4.782	50.16 ± 6.658	64.70 ± 2.580
	5		92.20	00.00	00.17 ± 0.076	94.26 ± 0.192	39.51 ± 7.173	60.63 ± 12.16
$\boldsymbol{\alpha_{new,t}}$ (%)	1		99.65	99.60	99.17 ± 0.037	99.58 ± 0.062	99.55 ± 0.035	99.59 ± 0.082
	2		95.55	97.95	96.09 ± 0.260	89.31 ± 0.311	90.98 ± 0.626	92.64 ± 2.302
	3		93.35	99.95	99.92 ± 0.012	86.06 ± 2.801	90.26 ± 1.435	83.40 ± 3.089
	4		84.75	99.90	99.95 ± 0.060	73.63 ± 3.861	85.65 ± 2.127	84.18 ± 2.715
	5		97.50	99.80	99.60 ± 0.023	93.55 ± 0.708	96.92 ± 0.774	96.51 ± 0.707
$\boldsymbol{\alpha_{all,t}}$ (%)	1		99.65	99.60	99.17 ± 0.037	99.58 ± 0.062	99.55 ± 0.035	99.59 ± 0.082
	2		95.75	48.97	49.28 ± 0.242	91.91 ± 0.043	91.83 ± 0.730	92.31 ± 1.163
	3		93.02	33.33	34.34 ± 0.009	79.98 ± 0.634	83.35 ± 1.597	86.93 ± 0.870
	4		87.51	25.00	25.21 ± 0.100	64.37 ± 0.707	64.66 ± 3.204	76.05 ± 1.391
	5	89.54	89.24	19.97	20.06 ± 0.059	63.21 ± 1.957	58.82 ± 2.521	69.88 ± 1.712
$\boldsymbol{\gamma_{base,t}}$ (nats)	1		209.7	209.8		207.7 ± 1.558	208.9 ± 1.213	209.7 ± 3.655
	2		207.4	240.7		209.0 ± 0.731	212.7 ± 0.579	212.1 ± 0.937
	3		207.6	258.7		213.0 ± 1.854	219.5 ± 1.376	216.9 ± 1.208
	4		207.7	243.6		213.6 ± 0.509	223.8 ± 0.837	217.1 ± 0.979
	5		208.4	306.5		217.7 ± 1.510	232.8 ± 5.048	222.8 ± 1.632
$\boldsymbol{\gamma_{new,t}}$ (nats)	1		209.7	209.8		207.7 ± 1.558	208.9 ± 1.213	209.7 ± 3.655
	2		241.1	240.2		238.7 ± 0.081	241.8 ± 0.502	241.9 ± 0.960
	3		213.6	211.8		211.6 ± 0.543	215.4 ± 0.501	213.0 ± 0.635
	4		220.5	219.7		219.5 ± 0.216	223.6 ± 0.381	220.9 ± 0.522
	5		246.2	242.0		242.8 ± 0.898	248.8 ± 0.398	244.0 ± 0.646
$\boldsymbol{\gamma_{all,t}}$ (nats)	1		209.7	209.8		207.7 ± 1.558	208.9 ± 1.213	209.7 ± 3.655
	2		224.2	240.4		223.8 ± 0.402	226.6 ± 2.31	226.9 ± 0.918
	3		220.7	246.1		221.9 ± 0.648	227.2 ± 0.606	224.9 ± 0.642
	4		220.4	238.7		225.1 ± 3.629	230.4 ± 0.524	226.1 ± 0.560
	5	224.8	226.2	275.1		230.5 ± 1.543	242.2 ± 0.754	234.6 ± 0.823
$\boldsymbol{KL_{all,t}}$ (nats)	1		12.17	12.20		9.710 ± 0.345	13.21 ± 0.635	13.28 ± 0.644
	2		16.54	17.47		10.65 ± 0.101	17.60 ± 0.755	15.56 ± 0.696
	3		18.84	19.34		11.34 ± 0.057	21.25 ± 0.872	17.35 ± 0.307
	4		20.06	17.31		10.96 ± 0.106	25.21 ± 0.929	19.81 ± 0.462
	5	23.27	20.27	21.61		11.45 ± 0.228	26.68 ± 0.859	20.47 ± 0.742

TABLE 12: Results for class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for AudioMNIST at the end of every task increment. Extension of table 3 in the main body. Here, in addition to the accuracy α_t, γ_t and KL_t also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of every task increment t.

Audio	t	CDVAE ISO	CDVAE UB	CDVAE LB	EWC	Dual Model	CDVAE	OCDVAE
	1		99.99	100.0	100.0 ± 0.000	100.0 ± 0.000	99.21 ± 0.568	99.95 ± 0.035
	2		99.92	00.00	00.16 ± 0.040	93.08 ± 5.854	98.98 ± 0.766	98.61 ± 0.490
$\boldsymbol{\alpha_{base,t}}$	3		100.0	00.00	00.29 ± 0.029	83.25 ± 6.844	92.44 ± 1.306	95.12 ± 2.248
(%)	4		99.92	00.00	00.31 ± 0.015	72.02 ± 0.677	76.43 ± 4.715	86.37 ± 5.63
	5		98.42	00.00	00.11 ± 0.007	61.57 ± 0.747	59.36 ± 7.147	79.73 ± 4.070
	1		99.99	100.0	100.0 ± 0.000	100.0 ± 0.000	99.21 ± 0.568	99.95 ± 0.035
	2		99.75	100.0	99.78 ± 0.019	86.25 ± 8.956	91.82 ± 4.577	89.23 ± 7.384
$\boldsymbol{\alpha_{new,t}}$	3		98.92	99.58	99.25 ± 0.054	95.16 ± 1.490	95.20 ± 1.495	94.43 ± 3.030
(%)	4		97.33	98.67	97.03 ± 0.019	62.52 ± 4.022	53.02 ± 6.132	72.22 ± 8.493
	5		98.67	100.0	99.41 ± 0.207	89.41 ± 0.691	84.93 ± 6.297	89.52 ± 6.586
	1		99.99	100.0	100.0 ± 0.000	100.0 ± 0.000	99.21 ± 0.568	99.95 ± 0.035
	2		99.83	50.00	50.16 ± 0.119	89.67 ± 1.763	93.84 ± 2.558	93.93 ± 3.756
$\boldsymbol{\alpha_{all,t}}$	3		99.56	33.19	33.28 ± 0.022	78.24 ± 3.315	94.26 ± 1.669	95.70 ± 1.524
(%)	4		98.60	24.58	24.50 ± 0.017	60.43 ± 4.209	77.90 ± 4.210	85.59 ± 3.930
	5	97.75	97.87	20.02	19.98 ± 0.032	47.42 ± 1.447	81.49 ± 1.944	87.72 ± 1.594
	1		433.7	423.2		422.3 ± 0.573	435.2 ± 15.69	424.2 ± 2.511
	2		422.5	439.4		426.6 ± 2.840	423.9 ± 0.517	425.2 ± 1.402
$\boldsymbol{\gamma_{base,t}}$	3		420.7	429.2		425.0 ± 0.339	422.7 ± 0.690	423.8 ± 1.148
(nats)	4		419.9	428.5		425.4 ± 0.081	422.8 ± 0.367	423.5 ± 0.937
	5		418.4	432.9		425.2 ± 0.244	422.7 ± 0.182	423.5 ± 0.586
	1		433.7	423.2		422.3 ± 0.573	435.2 ± 15.69	424.2 ± 2.511
	2		381.2	384.1		381.3 ± 2.039	382.5 ± 1.355	385.3 ± 12.56
$\boldsymbol{\gamma_{new,t}}$	3		435.9	436.7		436.8 ± 0.188	436.3 ± 0.639	436.9 ± 0.688
(nats)	4		485.9	487.1		486.5 ± 0.432	486.7 ± 0.385	486.5 ± 0.701
	5		421.3	425.2		422.4 ± 0.784	423.9 ± 0.681	422.9 ± 0.537
	1		433.7	423.2		422.3 ± 0.573	435.2 ± 15.69	424.2 ± 2.511
	2		401.9	411.8		404.0 ± 2.407	403.2 ± 0.831	403.5 ± 1.274
$\boldsymbol{\gamma_{all,t}}$	3		412.1	418.9		414.4 ± 0.385	413.6 ± 0.410	413.8 ± 0.573
(nats)	4		430.3	438.4		433.9 ± 0.374	432.4 ± 0.436	432.6 ± 0.862
	5	429.7	427.2	440.4		432.7 ± 0.385	431.4 ± 0.255	430.9 ± 0.541
	1		11.65	11.20		4.639 ± 0.107	11.78 ± 1.478	11.16 ± 0.713
	2		11.78	13.61		5.135 ± 0.127	15.13 ± 1.128	14.06 ± 1.140
$\boldsymbol{KL_{all,t}}$	3		13.40	17.09		5.427 ± 0.105	18.18 ± 1.140	13.61 ± 0.901
(nats)	4		13.61	14.41		5.243 ± 0.135	22.93 ± 1.134	17.58 ± 1.102
	5	17.89	15.15	14.52		5.470 ± 0.055	22.96 ± 0.912	18.52 ± 1.131



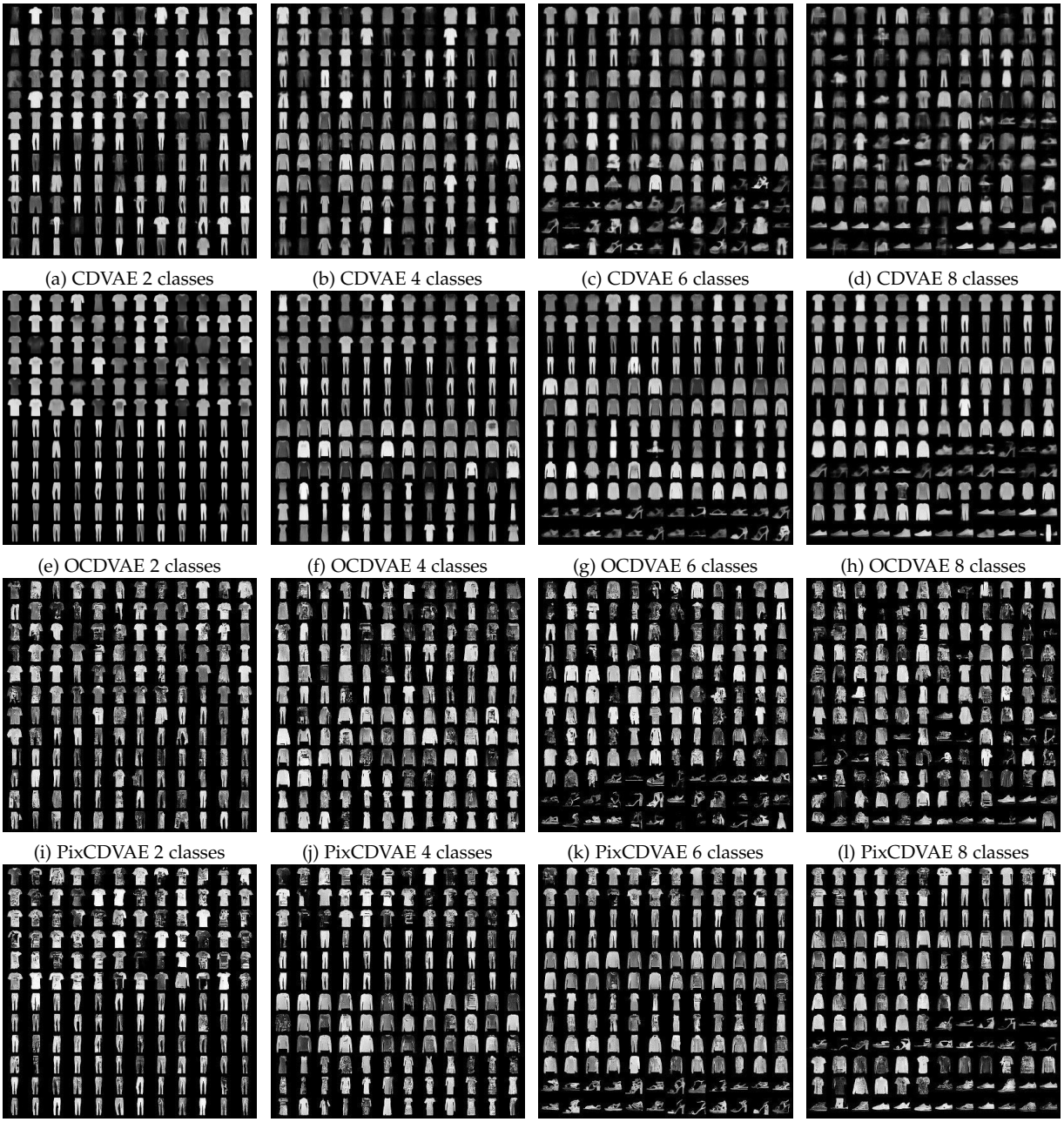
(a) CDVAE 2 classes (b) CDVAE 4 classes (c) CDVAE 6 classes (d) CDVAE 8 classes

(e) OCDVAE 2 classes (f) OCDVAE 4 classes (g) OCDVAE 6 classes (h) OCDVAE 8 classes

(i) PixCDVAE 2 classes (j) PixCDVAE 4 classes (k) PixCDVAE 6 classes (l) PixCDVAE 8 classes

(m) PixOCDVAE 2 classes (n) PixOCDVAE 4 classes (o) PixOCDVAE 6 classes (p) PixOCDVAE 8 classes

Fig. 5: Generated images for continually learned incremental MNIST at the end of task increments for CDVAE (a-d), OCDVAE (e-h), PixCDVAE (i-l) and PixOCDVAE (m-p). Each individual grid is sorted according to the class label that is predicted by the classifier.



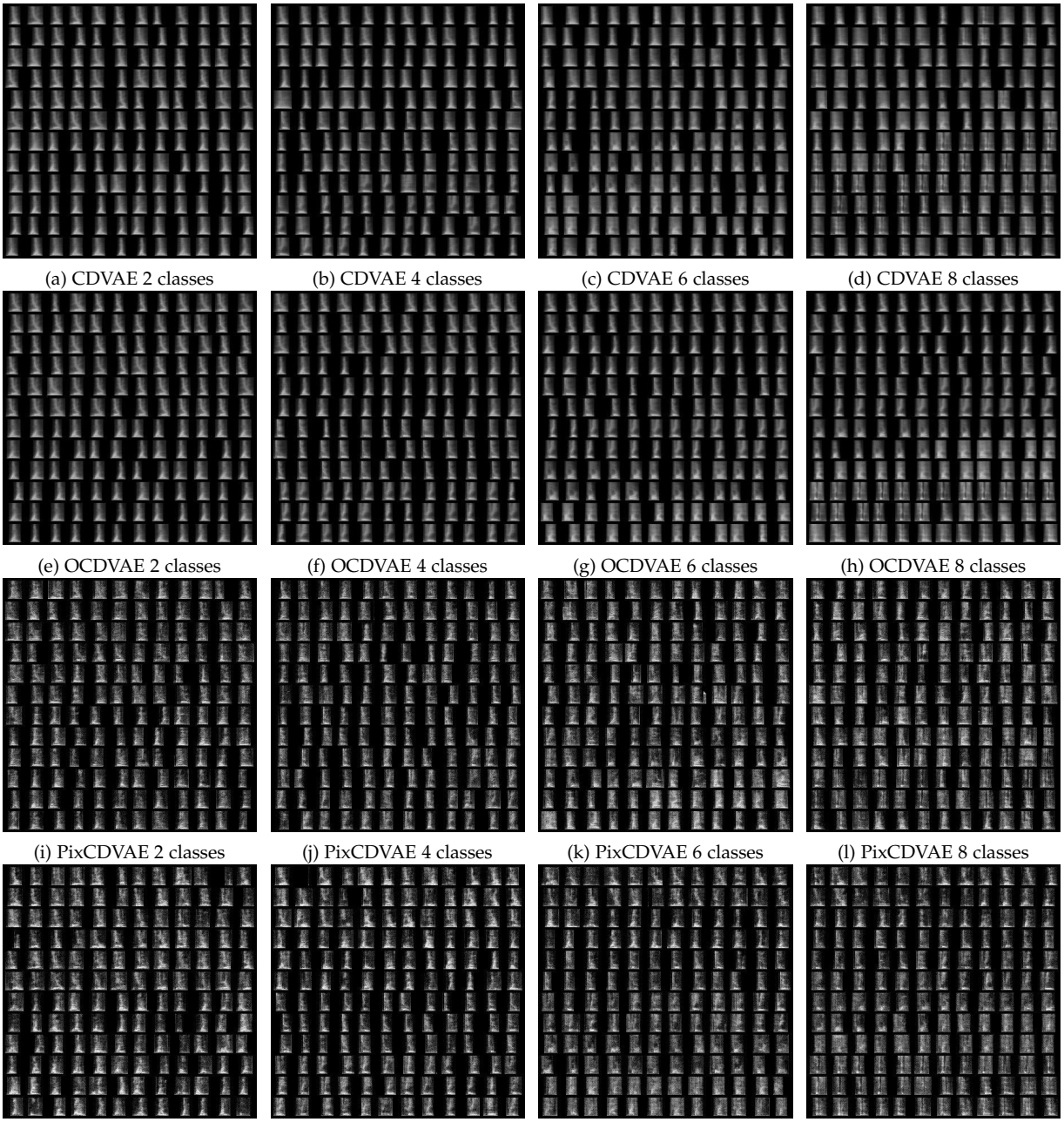
(a) CDVAE 2 classes (b) CDVAE 4 classes (c) CDVAE 6 classes (d) CDVAE 8 classes

(e) OCDVAE 2 classes (f) OCDVAE 4 classes (g) OCDVAE 6 classes (h) OCDVAE 8 classes

(i) PixCDVAE 2 classes (j) PixCDVAE 4 classes (k) PixCDVAE 6 classes (l) PixCDVAE 8 classes

(m) PixOCDVAE 2 classes (n) PixOCDVAE 4 classes (o) PixOCDVAE 6 classes (p) PixOCDVAE 8 classes

Fig. 6: Generated images for continually learned incremental FashionMNIST at the end of task increments for CDVAE (a-d), OCDVAE (e-h), PixCDVAE (i-l) and PixOCDVAE (m-p). Each individual grid is sorted according to the class label that is predicted by the classifier.



(a) CDVAE 2 classes (b) CDVAE 4 classes (c) CDVAE 6 classes (d) CDVAE 8 classes

(e) OCDVAE 2 classes (f) OCDVAE 4 classes (g) OCDVAE 6 classes (h) OCDVAE 8 classes

(i) PixCDVAE 2 classes (j) PixCDVAE 4 classes (k) PixCDVAE 6 classes (l) PixCDVAE 8 classes

(m) PixOCDVAE 2 classes (n) PixOCDVAE 4 classes (o) PixOCDVAE 6 classes (p) PixOCDVAE 8 classes

Fig. 7: Generated images for continually learned incremental AudioMNIST at the end of task increments for CDVAE (a-d), OCDVAE (e-h), PixCDVAE (i-l) and PixOCDVAE (m-p). Each individual grid is sorted according to the class label that is predicted by the classifier.

TABLE 13: Results for PixelVAE based class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for MNIST at the end of every task increment in analogy to table 10. Extension of table 4 in the main body. Here, in addition to the accuracy α_t, γ_t and KL_t also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of every task increment t.

MNIST	t	Dual Pix Model	PixCDVAE	PixOCDVAE
$\boldsymbol{\alpha_{base,t}}$ (%)	1	99.97 ± 0.002	99.97 ± 0.026	99.86 ± 0.084
	2	99.54 ± 0.285	96.90 ± 2.907	99.64 ± 0.095
	3	99.16 ± 0.611	90.12 ± 5.846	98.88 ± 0.491
	4	98.33 ± 1.119	76.84 ± 9.095	98.11 ± 0.797
	5	98.04 ± 1.397	56.53 ± 4.032	97.44 ± 0.785
$\boldsymbol{\alpha_{new,t}}$ (%)	1	99.97 ± 0.002	99.97 ± 0.026	99.86 ± 0.084
	2	99.71 ± 0.122	99.74 ± 0.052	99.82 ± 0.027
	3	99.41 ± 0.084	99.22 ± 0.082	99.56 ± 0.092
	4	98.61 ± 0.312	97.84 ± 0.180	98.80 ± 0.292
	5	97.31 ± 0.575	96.77 ± 0.337	98.63 ± 0.430
$\boldsymbol{\alpha_{all,t}}$ (%)	1	99.97 ± 0.002	99.97 ± 0.026	99.86 ± 0.084
	2	99.60 ± 0.142	98.37 ± 1.448	99.69 ± 0.051
	3	98.93 ± 0.291	96.14 ± 1.836	99.20 ± 0.057
	4	98.22 ± 0.560	91.25 ± 0.992	98.13 ± 0.281
	5	96.52 ± 0.658	83.61 ± 0.927	96.84 ± 0.346
$\boldsymbol{\gamma_{base,t}}$ (nats)	1	90.52 ± 0.263	100.0 ± 1.572	99.77 ± 2.768
	2	91.27 ± 0.789	100.4 ± 1.964	101.2 ± 3.601
	3	91.92 ± 0.991	100.3 ± 4.562	101.1 ± 4.014
	4	91.75 ± 1.136	102.7 ± 7.134	101.0 ± 4.573
	5	92.05 ± 1.212	102.4 ± 6.195	100.5 ± 4.942
$\boldsymbol{\gamma_{new,t}}$ (nats)	1	90.52 ± 0.263	100.0 ± 1.572	99.77 ± 2.768
	2	115.8 ± 0.805	125.7 ± 2.413	124.6 ± 3.822
	3	107.7 ± 0.600	118.3 ± 3.523	116.5 ± 2.219
	4	100.9 ± 0.659	107.1 ± 5.316	102.3 ± 1.844
	5	113.4 ± 0.820	118.2 ± 1.572	113.3 ± 0.755
$\boldsymbol{\gamma_{all,t}}$ (nats)	1	90.52 ± 0.263	100.0 ± 1.572	99.77 ± 2.768
	2	102.9 ± 0.408	111.9 ± 2.627	112.7 ± 3.300
	3	104.8 ± 1.114	114.9 ± 4.590	114.6 ± 4.788
	4	103.9 ± 0.759	114.3 ± 3.963	112.1 ± 2.150
	5	106.1 ± 0.868	118.7 ± 5.320	111.9 ± 2.663
$\boldsymbol{KL_{all,t}}$ (nats)	1	1.410 ± 0.181	5.629 ± 3.749	5.635 ± 3.739
	2	3.177 ± 0.702	9.238 ± 0.674	7.495 ± 0.738
	3	4.923 ± 1.085	12.13 ± 0.977	10.17 ± 1.528
	4	5.603 ± 1.250	14.32 ± 1.040	11.66 ± 1.004
	5	9.296 ± 1.346	16.37 ± 0.970	12.49 ± 0.551

TABLE 14: Results for PixelVAE based class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for FashionMNIST at the end of every task increment in analogy to table 11. Extension of table 4 in the main body. Here, in addition to the accuracy α_t, γ_t and KL_t also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of every task increment t.

Fashion	t	Dual Pix Model	PixCDVAE	PixOCDVAE
$\boldsymbol{\alpha_{base,t}}$ (%)	1	99.57 ± 0.091	99.58 ± 0.076	99.54 ± 0.079
	2	82.40 ± 6.688	90.06 ± 1.782	88.60 ± 1.998
	3	78.55 ± 3.964	83.70 ± 3.571	87.66 ± 0.375
	4	54.69 ± 3.853	50.23 ± 7.004	68.31 ± 3.308
	5	60.04 ± 5.151	47.83 ± 13.41	74.45 ± 2.889
$\boldsymbol{\alpha_{new,t}}$ (%)	1	99.57 ± 0.091	99.58 ± 0.076	99.54 ± 0.079
	2	97.73 ± 1.113	96.47 ± 0.596	97.31 ± 0.475
	3	99.09 ± 0.367	97.33 ± 0.725	96.88 ± 1.156
	4	97.55 ± 0.588	96.12 ± 0.675	95.47 ± 1.332
	5	98.85 ± 0.141	97.91 ± 0.596	98.63 ± 0.176
$\boldsymbol{\alpha_{all,t}}$ (%)	1	99.57 ± 0.091	99.58 ± 0.076	99.54 ± 0.079
	2	86.22 ± 3.704	92.93 ± 0.160	92.17 ± 1.425
	3	76.77 ± 4.378	84.07 ± 1.069	87.30 ± 0.322
	4	62.93 ± 3.738	64.42 ± 1.837	76.36 ± 1.267
	5	72.41 ± 2.941	63.05 ± 1.826	80.85 ± 0.721
$\boldsymbol{\gamma_{base,t}}$ (nats)	1	267.8 ± 1.246	230.8 ± 3.024	232.0 ± 2.159
	2	273.6 ± 0.631	232.5 ± 1.582	231.8 ± 0.416
	3	274.0 ± 0.552	235.6 ± 2.784	231.6 ± 0.832
	4	273.7 ± 0.504	236.4 ± 3.157	231.4 ± 2.550
	5	274.1 ± 0.349	241.1 ± 1.747	234.1 ± 1.498
$\boldsymbol{\gamma_{new,t}}$ (nats)	1	267.8 ± 1.246	230.8 ± 3.024	232.0 ± 2.159
	2	313.4 ± 1.006	275.8 ± 1.888	275.3 ± 1.473
	3	269.1 ± 0.616	268.3 ± 3.852	262.9 ± 1.893
	4	282.4 ± 0.321	259.1 ± 1.305	259.6 ± 2.050
	5	305.8 ± 0.286	283.2 ± 2.150	283.5 ± 2.458
$\boldsymbol{\gamma_{all,t}}$ (nats)	1	267.8 ± 1.246	230.8 ± 3.024	232.0 ± 2.159
	2	293.8 ± 0.349	254.3 ± 1.513	255.8 ± 0.436
	3	285.7 ± 0.510	261.5 ± 2.970	259.1 ± 0.929
	4	284.9 ± 0.703	263.2 ± 2.259	259.5 ± 3.218
	5	289.5 ± 0.396	271.7 ± 2.117	267.2 ± 0.586
$\boldsymbol{KL_{all,t}}$ (nats)	1	3.610 ± 0.856	7.164 ± 0.759	7.809 ± 1.255
	2	6.247 ± 0.710	13.79 ± 0.282	12.23 ± 0.287
	3	7.811 ± 0.799	18.26 ± 0.818	15.36 ± 0.530
	4	8.982 ± 0.812	21.75 ± 0.561	18.31 ± 0.333
	5	9.781 ± 1.068	22.14 ± 0.377	17.93 ± 0.360

TABLE 15: Results for PixelVAE based class incremental continual learning approaches averaged over 5 runs, baselines and the reference isolated learning scenario for AudioMNIST at the end of every task increment in analogy to table 12. Extension of table 4 in the main body. Here, in addition to the accuracy α_t, γ_t and KL_t also indicate the respective NLL reconstruction metrics and corresponding KL divergences at the end of every task increment t.

Audio	**t**	**Dual Pix Model**	**PixCDVAE**	**PixOCDVAE**
$\boldsymbol{\alpha_{base,t}}$ (%)	1	100.0 ± 0.000	99.71 ± 0.218	99.27 ± 0.410
	2	99.52 ± 0.273	97.86 ± 0.799	97.88 ± 2.478
	3	93.15 ± 3.062	81.38 ± 5.433	95.82 ± 3.602
	4	81.55 ± 8.468	50.58 ± 14.60	91.56 ± 5.640
	5	64.60 ± 8.739	29.94 ± 18.47	75.25 ± 10.18
$\boldsymbol{\alpha_{new,t}}$ (%)	1	100.0 ± 0.000	99.71 ± 0.218	99.27 ± 0.410
	2	99.71 ± 0.043	99.78 ± 0.128	99.81 ± 0.189
	3	98.23 ± 1.092	98.41 ± 0.507	99.30 ± 0.550
	4	95.31 ± 0.868	94.30 ± 0.914	97.87 ± 0.293
	5	98.18 ± 0.885	97.00 ± 0.520	99.43 ± 0.495
$\boldsymbol{\alpha_{all,t}}$ (%)	1	100.0 ± 0.000	99.71 ± 0.218	99.27 ± 0.410
	2	99.50 ± 0.157	98.64 ± 0.875	99.67 ± 0.033
	3	95.37 ± 1.750	90.10 ± 1.431	97.77 ± 1.017
	4	86.97 ± 2.797	75.55 ± 3.891	95.41 ± 1.345
	5	75.50 ± 3.032	63.44 ± 5.252	90.23 ± 1.139
$\boldsymbol{\gamma_{base,t}}$ (nats)	1	434.2 ± 1.068	432.6 ± 0.321	433.8 ± 0.370
	2	434.4 ± 1.082	432.5 ± 0.551	433.5 ± 1.464
	3	434.6 ± 0.785	432.9 ± 0.723	433.1 ± 1.269
	4	434.2 ± 1.209	433.0 ± 0.781	433.0 ± 1.283
	5	435.1 ± 1.915	431.4 ± 0.666	432.3 ± 0.189
$\boldsymbol{\gamma_{new,t}}$ (nats)	1	434.2 ± 1.068	432.6 ± 0.321	433.8 ± 0.370
	2	390.4 ± 0.694	389.4 ± 0.208	389.4 ± 1.304
	3	444.7 ± 0.545	442.7 ± 0.513	442.4 ± 0.275
	4	497.4 ± 0.740	494.4 ± 0.700	494.8 ± 0.386
	5	431.9 ± 1.032	428.0 ± 0.851	429.7 ± 1.223
$\boldsymbol{\gamma_{all,t}}$ (nats)	1	435.2 ± 15.69	432.6 ± 0.321	433.8 ± 0.370
	2	412.4 ± 0.871	410.9 ± 0.351	411.5 ± 1.406
	3	423.3 ± 0.618	421.0 ± 1.026	421.9 ± 0.661
	4	441.6 ± 0.420	439.8 ± 0.833	439.8 ± 0.718
	5	440.3 ± 1.297	436.9 ± 0.751	437.7 ± 0.432
$\boldsymbol{KL_{all,t}}$ (nats)	1	4.361 ± 0.671	9.293 ± 0.943	11.87 ± 1.504
	2	5.130 ± 0.636	14.00 ± 0.748	12.40 ± 0.719
	3	5.399 ± 0.724	20.28 ± 0.774	14.41 ± 0.461
	4	5.817 ± 1.038	24.91 ± 0.845	16.00 ± 0.505
	5	6.031 ± 0.832	27.14 ± 1.139	17.45 ± 0.835

REFERENCES

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *International Conference on Learning Representations (ICLR)*, 2013.

[2] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-Supervised Learning with Deep Generative Models," *Neural Information Processing Systems (NeurIPS)*, 2014.

[3] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *International Conference on Learning Representations (ICLR)*, 2017.

[4] M. D. Hoffman and M. J. Johnson, "ELBO surgery: yet another way to carve up the variational evidence lower bound," *Neural Information Processing Systems (NeurIPS), Advances in Approximate Bayesian Inference Workshop*, 2016.

[5] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta-VAE," *Neural Information Processing Systems (NeurIPS), Workshop on Learning Disentangled Representations*, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," *British Machine Vision Conference (BMVC)*, 2016.

[8] I. Gulrajani, K. Kumar, A. Faruk, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville, "PixelVAE: a Latent Variable Model for Natural Images," *International Conference on Learning Representations (ICLR)*, 2017.

[9] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational Lossy Autoencoder," *International Conference on Learning Representations (ICLR)*, 2017.

[10] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *International Conference on Machine Learning (ICML)*, 2015.

[11] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *International Conference on Computer Vision (ICCV)*, 2015.

[13] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring Catastrophic Forgetting in Neural Networks," *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[14] S. Farquhar and Y. Gal, "A Unifying Bayesian View of Continual Learning," *Neural Information Processing Systems (NeurIPS) Bayesian Deep Learning Workshop*, 2018.

[15] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 114, no. 13, pp. 3521–3526, 2017.

Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers?

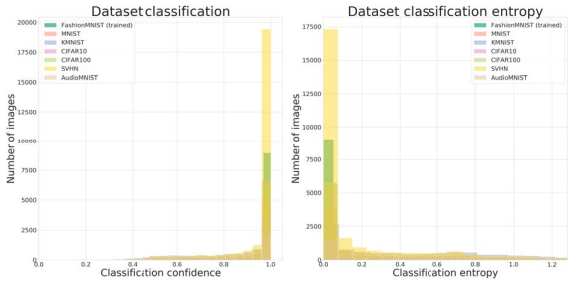
Martin Mundt, Iuliia Pliushch, Sagnik Majumder and Visvanathan Ramesh
Goethe University, Frankfurt, Germany
{mmundt, pliushch, vramesh}@em.uni-frankfurt.de majumder@ccc.cs.uni-frankfurt.de

Abstract

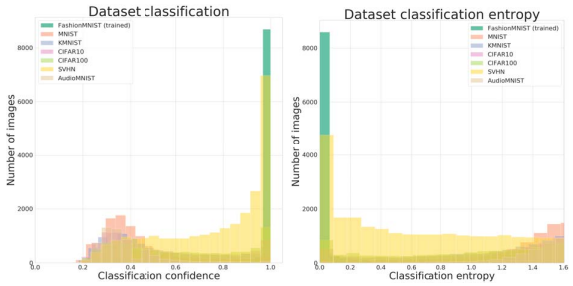
We present an analysis of predictive uncertainty based out-of-distribution detection for different approaches to estimate various models' epistemic uncertainty and contrast it with extreme value theory based open set recognition. While the former alone does not seem to be enough to overcome this challenge, we demonstrate that uncertainty goes hand in hand with the latter method. This seems to be particularly reflected in a generative model approach, where we show that posterior based open set recognition outperforms discriminative models and predictive uncertainty based outlier rejection, raising the question of whether classifiers need to be generative in order to know what they have not seen.

1. Introduction

A particular challenge of modern deep learning based computer vision systems is a neural network's tendency to produce outputs with high confidence when presented with task unrelated data. Early works have identified this issue and have shown that methods employing forms of thresholding a neural network's softmax confidence are generally not enough for rejection of unknown inputs [15]. Recently, deep learning methods for approximate Bayesian inference [12, 5, 10, 5], such as deep latent variable models [12] or Monte Carlo dropout (MCD) [5], have opened the pathway to capturing neural network uncertainty. Access to these uncertainties comes with the promise of allowing to separate what a model is truly confident about through output variability. However, misclassification is not prevented and in a Bayesian approach uncertain inputs are not necesssarily unknown and vice versa unknowns do not necessarily appear as uncertain [3]. This has recently been observed on a large empirical scale [19] and figure 1 illustrates this challenge. Here we show the prediction confidence and entropy of two deep residual neural networks [7, 23] trained on FashionMNIST [22] as obtained through a standard feed-forward pass and variational inference using 50 MCD samples. Neither



(a) Standard deep neural network classifier



(b) Approximate variational inference with average over 50 Monte Carlo dropout stochastic forward passes

Figure 1: Classification confidence and entropy for deep neural network classifiers with and without approximate variational inference. Models have been trained on FashionMNIST and are evaluated on out-of-distribution datasets.

of the approaches is able to avoid over-confident predictions on previously unseen datasets, even if MCD fares much better in separating the distributions.

A different thread for open-set recognition in deep neural networks is through extreme-value theory (EVT) based meta-recognition [21, 2]. When applied to a neural network's penultimate feature representation, it has originally been shown to improve out-of-distribution (OOD) detection in contrast to simply relying on a neural network's output values. We have recently extended this approach by adapting EVT to each class' approximate posterior in a latent

variable model for continual learning [16]. However, EVT based open set recognition and capturing epistemic uncertainty need not be seen as separate approaches. In this work we thus empirically demonstrate that:

1. combining the benefit of capturing a model's uncertainty with EVT based open set recognition outperforms out-of-distribution detection using prediction uncertainty on a variety of classification tasks.

2. moving to a generative model, which in addition to the label distribution $p(\boldsymbol{y})$ also approximates the data distribution $p(\boldsymbol{x})$, results in similar prediction entropy but further improves the latent based EVT approach.

2. Variational open set neural networks

We consider three different models for which we investigate open set detection based on both prediction uncertainty as well as the EVT based approach. The simplest model is a standard deep neural network classifier. Such a model however doesn't capture epistemic uncertainty. We thus consider variational Bayesian inference with neural networks consisting of an encoder with variational parameters $\boldsymbol{\theta}$ and a linear classifier $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$ that gives the probability density of target y given a sample $\boldsymbol{z}$ from the approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. We optionally also consider the addition of a probabilistic decoder $p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z})$ that returns the probability density of x under the generative model. With the added decoder we thus learn a joint generative model $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$. These models are trained by optimizing the following variational evidence lower-bound:

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) = \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}|\boldsymbol{z}) + \log p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})\right] - \beta KL(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}))] \tag{1}$$

Here β is an additional parameter that weighs the contribution of the Kullback-Leibler divergence between approximate posterior $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and prior $p(\boldsymbol{z})$ as suggested by the authors of β-Variational Autoencoder [8]. We can summarize the considered models as follows:

1. Standard discriminative neural network classifier that maximizes $\log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$ (not described by equation 1).

2. Variational discriminative classifier with graph $\boldsymbol{x} \rightarrow \boldsymbol{z} \rightarrow \boldsymbol{y}$. Maximizes the lower-bound to $p(\boldsymbol{y})$ as given by equation 1 without the $\boldsymbol{\phi}$ dependent (blue) term.

3. Variational generative model as described by equation 1 with generative process $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$. In addition to $p(\boldsymbol{y})$, also jointly maximizes the variational lower-bound to $p(\boldsymbol{x})$.

Following a variational formulation, the second and third model have natural means to capture epistemic uncertainty, i.e. uncertainty that could be lowered by training on more data. Drawing multiple samples $\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ from the approximate posterior yields a distribution over the models' outputs as specified by the expectation in 1. For all above approaches we can additionally place a prior distribution over the models' weights to find a distribution $q_{\boldsymbol{\theta}}(\boldsymbol{W})$ for the weights posterior. This can be achieved by performing a dropout operation [20] at every weight layer and conducting approximate variational inference through multiple stochastic forward passes during evaluation. We do not consider variational autoencoders [12] that only maximize the variational lower-bound to $p(\boldsymbol{x})$ (i.e. equation 1 without the blue term), as these models have been shown to be incapable of separating seen from unseen data in previous literature [17].

Algorithm 1 Open set recognition calibration for deep variational neural networks. A Weibull model fit of tail-size η is conducted to bound the per class approximate posterior. Per class c Weibull models $\boldsymbol{\rho}_c$ with their respective shift τ_c, shape κ_c and scale λ_c parameters are returned.

Require: Trained encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and classifier $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$
Require: Classifier probabilities $p_{\boldsymbol{\xi}}(\boldsymbol{y}|\boldsymbol{z})$ and samples from the approximate posterior $\boldsymbol{z}(\boldsymbol{x}^{(i)}) \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ for each training dataset example $\boldsymbol{x}^{(i)}$
Require: For each class c, let $\boldsymbol{S}_c^{(i)} = \boldsymbol{z}(\boldsymbol{x}_c'^{(i)})$ for each correctly classified training example $\boldsymbol{x}_c'^{(i)}$
for $c = 1 \ldots C$ **do**
  Get per class latent mean $\bar{\boldsymbol{S}}_c = mean(\boldsymbol{S}_c^{(i)})$
  Weibull model $\boldsymbol{\rho}_c =$ Fit Weibull $\left(||\boldsymbol{S}_c - \bar{\boldsymbol{S}}_c||, \eta\right)$
Return means $\bar{\boldsymbol{S}}$ and Weibull models $\boldsymbol{\rho}$

Algorithm 2 Open set probability estimation for unknown inputs. Data points are considered statistical outliers if a Weibull model's cumulative distribution function's (CDF) probability value exceeds a task specific prior Ω_t.

Require: Trained encoder $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$
Require: Per class latent mean $\bar{\boldsymbol{S}}_c$ and Weibull model $\boldsymbol{\rho}_c$, each with parameters $(\tau_c, \kappa_c, \lambda_c)$
For a novel input example $\hat{\boldsymbol{x}}$ **sample** $\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\hat{\boldsymbol{x}})$
Compute distances to $\bar{\boldsymbol{S}}_c$**:** $d_c = ||\bar{\boldsymbol{S}}_c - \boldsymbol{z}||$
for $c = 1 \ldots C$ **do**
  Weibull CDF $\omega_c(d_c) = 1 - \exp\left(-\frac{||d_c - \tau_c||}{\lambda_c}\right)^{\kappa_c}$
Reject input if $\omega_c(d_c) > \Omega_t$ for any class c.

2.1. Open set meta-recognition

For a standard deep neural network classifier we follow the EVT based approach based on the features of the penultimate layer [2]. To bound the open-space risk of our variational models we follow the adaptation of this method to operate on the latent space and thus on the basis of the approx-

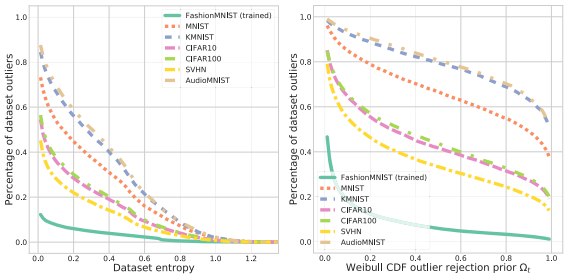
imate posterior in Bayesian inference [16]. In the Bayesian interpretation we obtain a Weibull distribution fit on the distances from the approximate posterior $\boldsymbol{z}(\boldsymbol{x}) \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ of each correctly classified training example. This leads to a bound on the regions of posterior high density as the tail of the Weibull distribution limits the amount of allowed low density space around these regions. Given such an estimate of the regions where the posterior has high density and the model can thus be trusted to make an informed decision, a novel unseen input example can be rejected according to the statistical outlier probability given the Weibull cumulative distribution function (CDF) between the unseen example's posterior samples and their distances to the high density regions. The corresponding procedures to obtain the Weibull fits and estimate an unseen data-point's outlier probability are outlined in algorithms 1 and 2.

3. Experiments and results

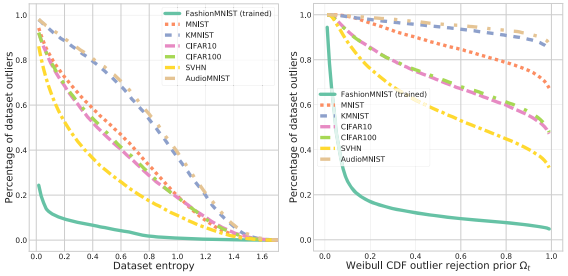
We base our encoder and optional decoder architecture on 14-layer wide residual networks [7, 23], in the variational cases with a latent dimensionality of 60. The classifier always consists of a single linear layer. We optimize all models using a mini-batch size of 128 and Adam [11] with a learning rate of 0.001, batch normalization [9] with a value of 10^{-5}, ReLU activations and weight initialization according to He et. al [6]. For each convolution we include a dropout layer with a rate of 0.2 that we can use for MCD. We train all our model variants for 150 epochs until full convergence on three datasets: FashionMNIST [22], MNIST [14] and SVHN [18]. We do not apply any preprocessing or data augmentation. For the EVT based outlier rejection we fit Weibull models with a tail-size set to 5% of training data examples per class. The used distance measure is the cosine distance. After training we evaluate out of distribution detection on the other two datasets and additionaly the KMNIST [4], CIFAR10 and 100 [13] and the non-image based AudioMNIST [1] datasets. For the latter we follow the authors' steps to convert the audio data into spectrograms. To make this cross-dataset evaluation possible, we repeat all gray-scale datasets to a three channel representations and resize all images to 32×32.

3.1. Results and discussion

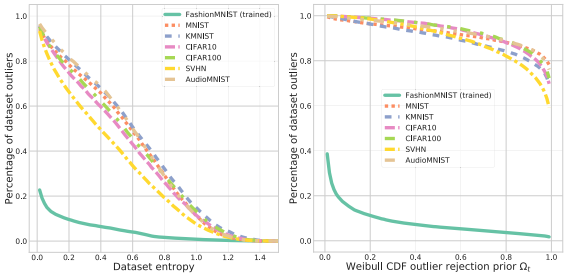
We show outlier rejection curves using both prediction uncertainty as well as EVT based OOD recognition for the three network types trained on FashionMNIST in figure 2. Rejection rates for the variational approaches were computed using 100 approximate posterior samples to capture epistemic uncertainty. When looking at the prediction entropy, we can observe that a standard deep neural network classifier predicts over-confidently for all OOD data. While the EVT based approach alleviates this to a certain extent, the challenge of OOD detection still largely persists. Moving to one of the variational models increases the entropy of OOD datasets, although not to the point where a separation from statistically inlying data is possible. Here, the EVT approach fares much better in achieving such separation. Nevertheless, this separation is only consistent across a wide range of rejection priors with the inclusion of the joint generative model. This is particularly important since this rejection prior has to be determined based on the original inlying validation data, as we can assume no access to OOD data upfront. Notice how this choice impacts rejection rates of the joint generative model to a much lesser extent. In addition we show the variational models of figure 2 panels (b) and (c) in figure 3 with 50 Monte Carlo dropout



(a) Standard discriminative classifier $p(\boldsymbol{y}|\boldsymbol{x})$



(b) Variational Bayes classifier $p(\boldsymbol{y}|\boldsymbol{z})$

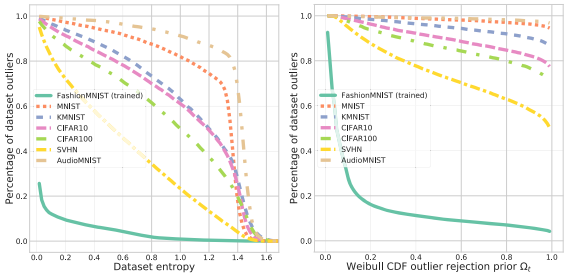


(c) Variational Bayes joint generative model $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$

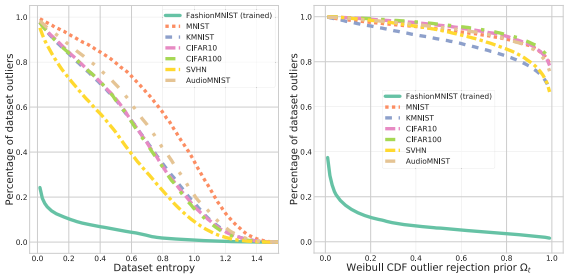
Figure 2: The three different models trained on FashionMNIST and evaluated on unseen datasets. For each model a pair of outlier rejection curves is shown. Left panels depict outlier rejection based on prediction entropy, whereas right panels show the EVT based open set recognition across the range of statistical outlier rejection priors Ω_t.

Outlier detection at 95% trained dataset inliers (%)			FashionMNIST		MNIST		KMNIST		CIFAR10		CIFAR100		SVHN		AudioMNIST	
Trained	Model variant	Test acc.	Entropy	Latent	Entropy	Latent	Entropy	Latent	Entropy	Latent	Entropy	Latent	Entropy	Latent	Entropy	Latent
Fashion MNIST	standard discriminative	93.36	4.903	4.852	38.36	63.29	48.82	76.97	23.75	38.78	25.27	40.23	18.21	30.65	51.28	77.96
	variational discriminative	93.73	4.911	4.826	50.51	67.42	72.23	84.51	43.64	47.13	45.39	47.87	28.79	32.06	74.03	87.20
	variational generative	93.57	4.878	4.992	54.58	91.13	56.31	88.34	48.69	92.96	53.03	93.36	38.87	88.82	55.87	92.23
	variational discriminative - MCD	93.70	4.864	4.887	91.99	95.24	83.84	88.95	79.27	81.84	72.24	76.86	48.24	58.73	97.01	97.56
	variational generative - MCD	93.68	4.899	4.908	84.32	95.05	67.24	88.37	68.40	97.16	68.07	97.51	49.98	94.51	75.59	95.11
MNIST	standard discriminative	99.43	88.04	90.71	4.968	4.873	85.25	85.40	91.06	87.62	92.39	88.47	86.85	85.59	93.88	93.40
	variational discriminative	99.57	97.55	99.86	4.890	4.871	95.18	99.53	99.76	99.98	99.69	99.97	94.37	97.70	98.61	99.65
	variational generative	99.53	95.12	96.60	4.888	4.954	97.15	98.97	98.60	99.81	98.64	99.65	96.53	96.29	99.65	99.98
	variational discriminative - MCD	99.55	99.56	99.93	4.879	4.932	98.82	99.66	99.96	99.98	99.95	99.99	98.32	98.97	99.86	99.90
	variational generative - MCD	99.56	98.61	99.18	4.841	4.873	96.81	99.75	99.73	99.82	99.89	99.89	97.47	98.42	98.95	99.15
SVHN	standard discriminative	97.34	69.67	71.99	18.61	23.48	65.07	74.93	73.96	83.00	72.43	80.34	4.861	4.924	62.75	67.98
	variational discriminative	97.59	75.76	81.00	21.17	24.93	77.14	91.89	82.29	88.68	80.48	88.38	4.879	4.980	72.86	89.36
	variational generative	97.68	75.20	99.13	30.10	70.68	82.88	98.48	81.63	95.14	80.79	93.49	4.893	4.927	72.41	95.26
	variational discriminative - MCD	97.57	84.97	89.71	95.27	94.97	84.48	90.26	85.86	94.94	85.78	93.46	4.962	4.922	81.66	88.61
	variational generative - MCD	97.58	83.73	93.53	100.0	100.0	98.32	97.57	82.16	93.03	80.40	92.77	4.893	4.910	88.16	94.53

Table 1: Test accuracies and outlier detection values of the three different network types described in section 2 when considering 95% of training validation data is inlying. Additional values are provided with Monte Carlo dropout (MCD). The variational approaches are reported with 100 $\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ samples and the optional additional 50 MCD samples.



(a) Variational Bayes classifier $p(\boldsymbol{y}|\boldsymbol{z})$



(b) Variational Bayes joint generative model $p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$

Figure 3: Pair of outlier rejection curves based on prediction entropy (left) and approximate posterior based statistical outlier rejection (right) in analogy to figure 2. Here, panels (a) and (b) correspond to panels (b) and (c) in figure 2 with additional variational Monte Carlo dropout inference.

samples. We have observed no substantial further benefits with more samples. Although this sampling can be computationally prohibitively expensive, we have included this comparison to give a better impression of how distributions on a neural network's weights can aid in capturing uncertainty. In fact, we can observe that in both cases the prediction entropy is further increased, albeit still suffers from the same challenge as outlined before. On the other hand, the EVT based approach profits similarly from MCD with the generative model still outperforming all other methods and achieving nearly perfect OOD detection.

We have quantified these results in table 1, where we report the network test accuracy as well as the outlier rejection rate with rejection priors and entropy thresholds determined according to categorizing 95 % of the trained dataset's validation data as inlying. For all values we can observe that capturing epistemic uncertainty with variational Bayes approaches improves upon a standard neural network classifier both slightly in test accuracy as well as in OOD detection. This improvement is further apparent when using the EVT approach that outperforms OOD detection with prediction uncertainty in all cases. Lastly, the joint generative model is apparent to improve the EVT based OOD detection as the posterior now also explicitly captures information about the data distribution $p(\boldsymbol{x})$.

4. Conclusion

We have provided an analysis of prediction uncertainty and EVT based out-of-distribution detection approaches for different model types and ways to estimate a model's epistemic uncertainty. While further larger scale evaluation is necessary, our results allow for two observations. First, whereas OOD detection is difficult based on prediction values even when epistemic uncertainty is captured, EVT based open set recognition based on a latent model's approximate posterior can offer a solution to a large degree. Second, we might require generative models for open set detection in classification, even if previous work has shown that generative approaches that only model the data distribution seem to fail to distinguish unseen from seen data [17].

References

[1] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek. Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. *arXiv preprint arXiv: 1807.03418*, 2018.

[2] A. Bendale and T. E. Boult. Towards Open Set Deep Networks. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] T. E. Boult, S. Cruz, A. Dhamija, M. Gunther, J. Henrydoss, and W. Scheirer. Learning and the Unknown : Surveying Steps Toward Open World Recognition. *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[4] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep Learning for Classical Japanese Literature. *Neural Information Processing Systems (NeurIPS), Workshop on Machine Learning for Creativity and Design*, 2018.

[5] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation : Representing Model Uncertainty in Deep Learning. *International Conference on Machine Learning (ICML)*, 48, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *International Conference on Computer Vision (ICCV)*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *International Conference on Learning Representations (ICLR)*, 2017.

[9] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning (ICML)*, 2015.

[10] A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Neural Information Processing Systems (NeurIPS)*, 2017.

[11] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[12] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations (ICLR)*, 2013.

[13] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Toronto, 2009.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

[15] O. Matan, R. Kiang, C. E. Stenard, and B. E. Boser. Handwritten Character Recognition Using Neural Network Architectures. *4th USPS Advanced Technology Conference*, 2(5):1003–1011, 1990.

[16] Martin Mundt, Sagnik Majumder, Iuliia Pliushch, and Visvanathan Ramesh. Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition. *arXiv preprint arXiv: 1905.12019*, 2019.

[17] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? *International Conference on Learning Representations (ICLR)*, 2019.

[18] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. *Neural Information Processing Systems (NeurIPS), Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[19] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. *arXiv preprint arXiv: 1906.02530*, 2019.

[20] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMRL)*, 15:1929–1958, 2014.

[21] M. R. P. Thomas, J. Ahrens, and I. Tashev. Probability Models For Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[22] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv: 1708.07747*, 2017.

[23] S. Zagoruyko and N. Komodakis. Wide Residual Networks. *British Machine Vision Conference (BMVC)*, 2016.

Real-world Application of Developed Techniques to Concrete Defect Detection

This section includes qualitative illustrations and practical demonstrations of the developed techniques to concrete defect detection in context of the proposed CODEBRIM dataset, section 1.3. The majority of the displayed results have been reported in technical deliverable reports (Mundt et al., 2018c,a) of the the associated European Union's Horizon 2020 "AEROBI" project under grant agreement No. 687384. It is the only thesis section that has not yet been transcribed to a publicly available manuscript. At its center, the section serves the purpose of putting this chapter's works into the investigated real-world application perspective and provide further intuition in addition to the previously conducted quantitative experiments.

2.3.1 Semantic Segmentation

The first qualitative demonstration is the extension of developed classification models towards the aim of semantic segmentation. Recall that the CODEBRIM dataset has been annotated by humans in a coarse manner, i.e. locations of a defect have been annotated and their corresponding classes labelled. As argued in the respective publication, this has been motivated from a detailed annotation procedure, where each and every individual pixel of an image is semantically labelled, being excessively time consuming on very high-resolution images. At the same time, the quantitative classification experiments that contrast the accuracy versus cropped bounding box patch size trade-off have shown that sub-sampling to lower-resolutions can result in heavy performance decrease. However, in practice such a detailed prediction of a defect might be required for a civil engineer to asses the extent of the damage and rate its severity for a structure's integrity (Koch et al., 2015). The natural question is then how we can obtain such a full semantic segmentation with our deep neural network, without explicitly having to train the network on detailed ground-truths.

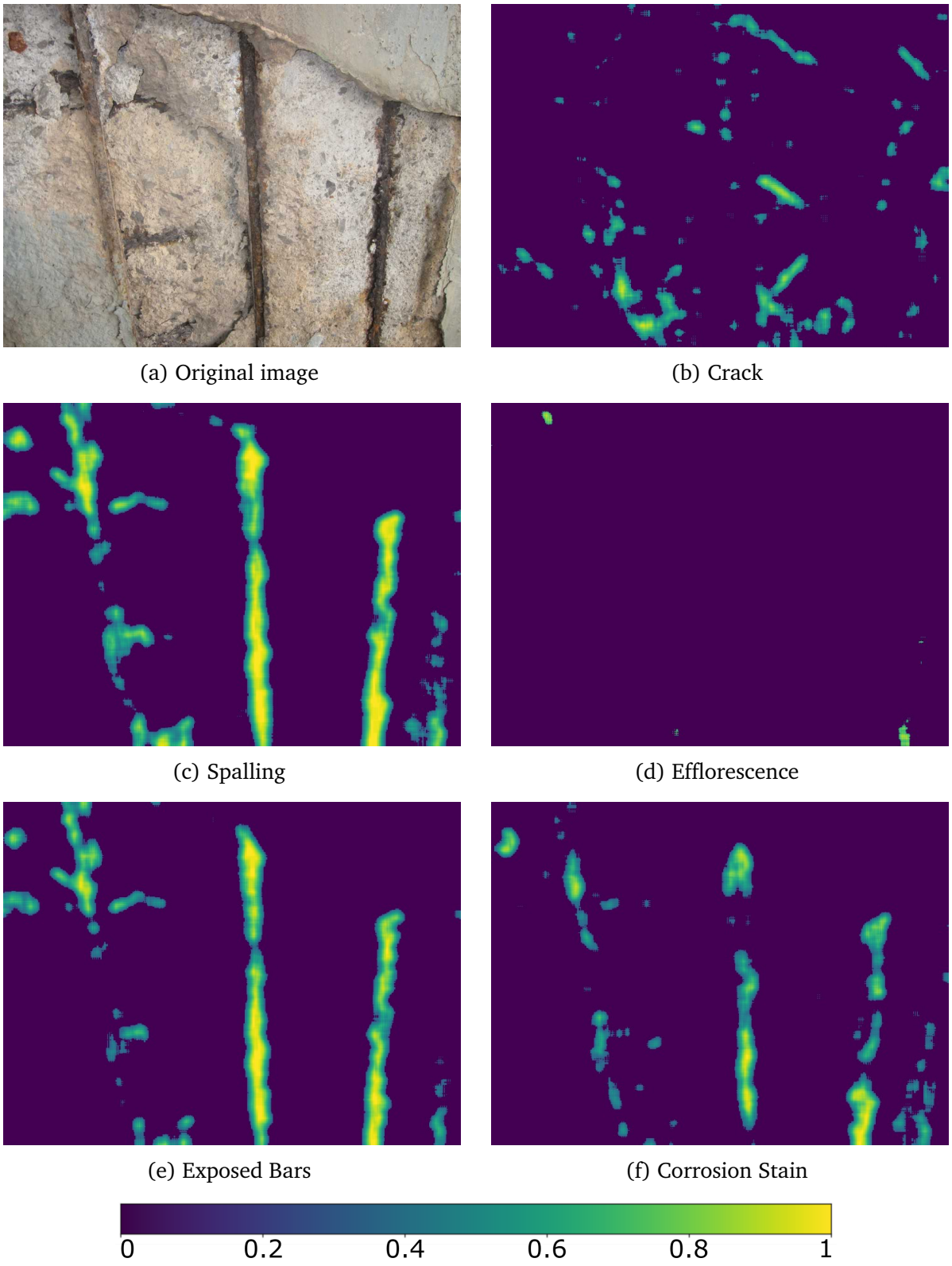
To our advantage, the entire model trained for classification can be treated as the filter of a convolution in the spirit of Bell et al. (2015). As the model has been trained on much smaller patches than the full-resolution images, i.e. the bounding box contents, this resembles a sliding window where the model predicts classes based on local regions. To give a practical example, a conventional AlexNet model trained on 224x224 image patches is computed 6000 times horizontally on an image of width 6000, assuming padding to preserve the spatial dimensionality. This is multiplied by the number of times the model has to also be applied to the vertical dimension of the image, resulting in an overall very large amount

of model forward passes. Although computationally heavy, in practice this can be made computationally significantly less complex than the initial intuition suggests. Given that the convolutional computation on each local patch is independent of the other image regions, a large amount of image regions can be processed in parallel on modern graphics processing units, subject to the limits of the specifically used device. The latter can be seen in complete analogy to the fully parallelized mini-batch computation of independent images in the training procedure. In the particular examples shown in the following, with a conventional consumer NVIDIA GTX 2080 and depending on the exact neural network architecture, this practically reduces the computation time from a scale between minutes to hours, to a range of seconds.

Two qualitative examples are shown in figures 2.1 and 2.2. They contain an original image and five confidence maps, each with the likelihood of a pixel corresponding to one of the five defect types: crack, spalling, efflorescence, exposed bars and corrosion stain. Note that these cannot simply be merged into one prediction, as the task is inherently multi-target, i.e. different defect types can appear and overlap in the same image. This is particularly relevant for defects such as cracks and efflorescence, where the latter's calcium deposit often settles around damp cracks, or spallation leading to partially exposed bars that can corrode over time. A co-occurrence of defects thus generally coincides with increased defect severity. Figure 2.1 shows an image where a respective heavy defect spans almost the entire view. On the one hand, we can observe that the sliding window deep network predictions generally correctly and confidently identifies the exposed bars and the regions where they are corroded. On the other hand, there seems to be confusion with respect to various pixels being attributed to cracks and the spalled area being equated to the bars, although technically almost the entire image shows a missing cover material. Similar observations can be made in figure 2.2, where a corrosion stain is correctly picked up. At the same time, one of the detected defects is further misclassified, i.e. the corrosion stain is also falsely attributed to spalling and the exposition of a small bar, and dirt markings are confused for cracks and efflorescence.

While the classification model thus seems to be able to handle images at completely different scales, largely due to the inclusion of spatial pyramidal pooling induced quasi scale invariance, its use for semantic segmentation is somewhat limited due to an overestimation of defects and partial attribution to false categories. In principle a quantitative evaluation would be necessary to effectively assess whether the above technique satisfies practical desiderata. Apart from a lack of detailed pixel-wise ground-truths, the quantitative evaluation in this form has not been pursued further for two additional reasons. First, in the AEROBI project the limitation of smoothed deep learning predictions that lack local detail

has been addressed through algorithmic fusion with a principled traditional computer vision algorithm based on topological pits to identify cracks in materials Mundt et al. (2018c,d). This also significantly lessens the threat of data population based erratic predictions of the crack category by using the statistics of the instance. This content is not included here, as it is out of scope of the thesis' main matter and because it is not the leading contribution of the author. Second, the observed intermingling of individual defect classes can to an extent be lifted by moving away from a purely discriminative perspective and moving to the earlier introduced view of learning generative factors. In consequence, this can then also be used to flag untrustworthy predictions for an image based on a mismatch of an individual instance's statistics in comparison with the encoded data population. Both of these aspects are qualitatively shown in the subsequent subsection.

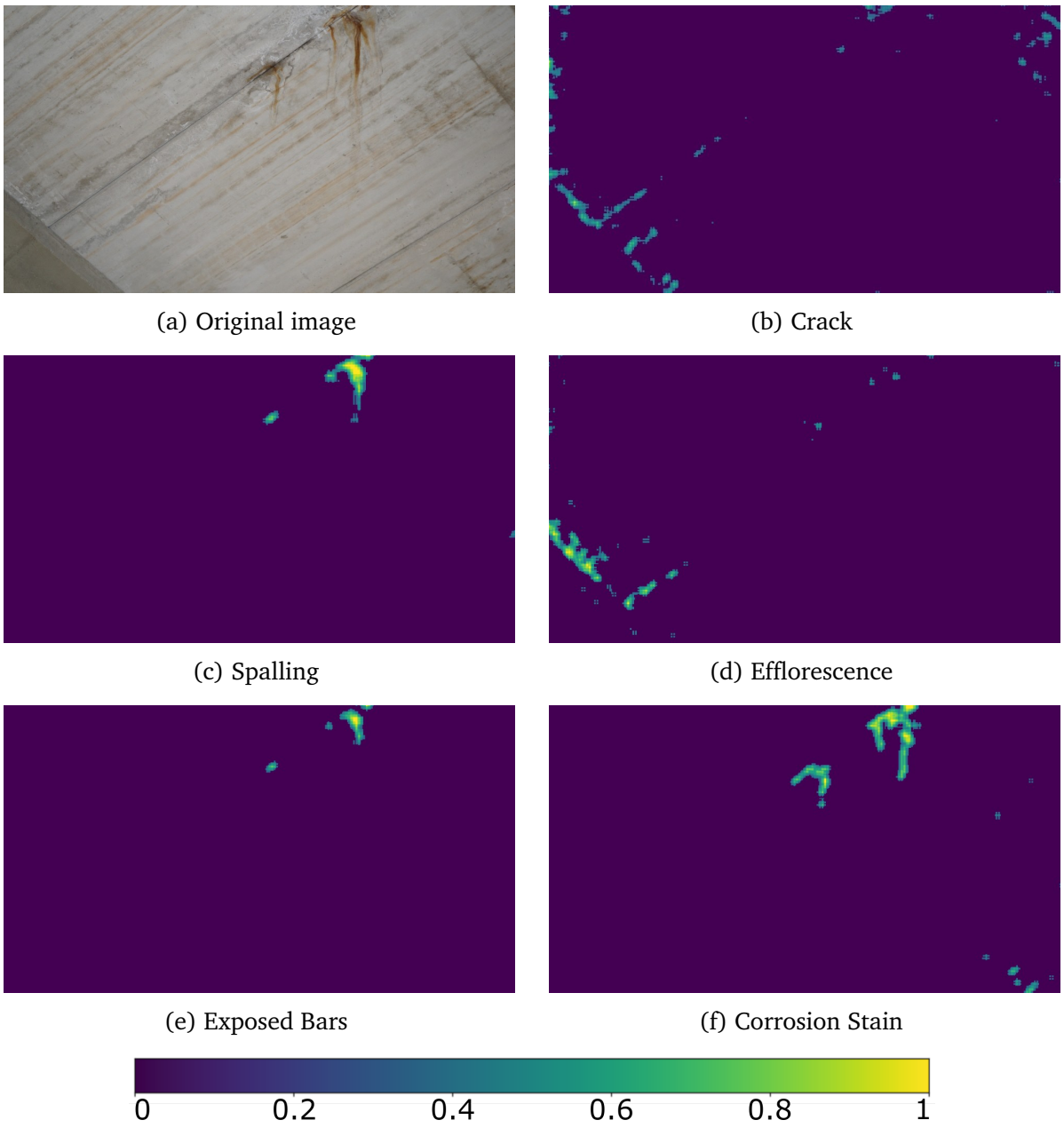


(a) Original image (b) Crack

(c) Spalling (d) Efflorescence

(e) Exposed Bars (f) Corrosion Stain

Figure 2.1: Example for semantic segmentation through sliding window prediction of a deep neural network trained for image patch classification. Maps show the confidence of a pixel being attributed to each defect class. Figure originally appeared in Mundt et al. (2018c).



(a) Original image (b) Crack

(c) Spalling (d) Efflorescence

(e) Exposed Bars (f) Corrosion Stain

Figure 2.2: Example for semantic segmentation through sliding window prediction of a deep neural network trained for image patch classification. Maps show the confidence of a pixel being attributed to each defect class. Figure originally appeared in Mundt et al. (2018c).

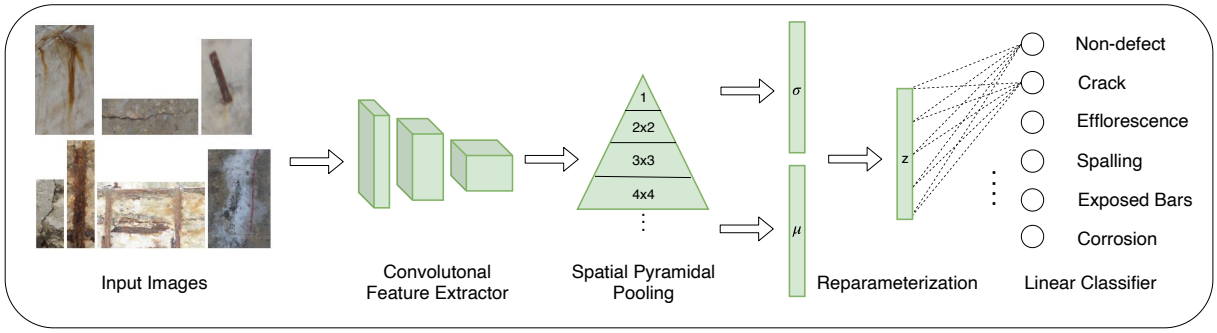
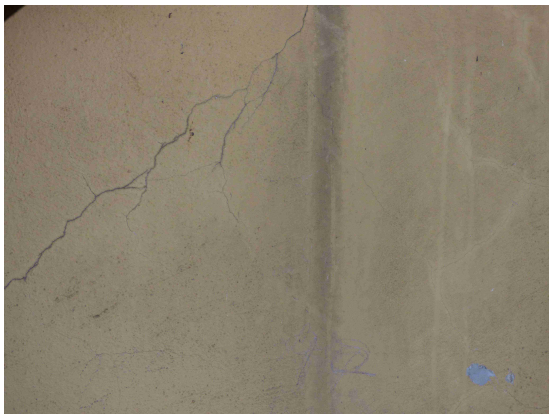


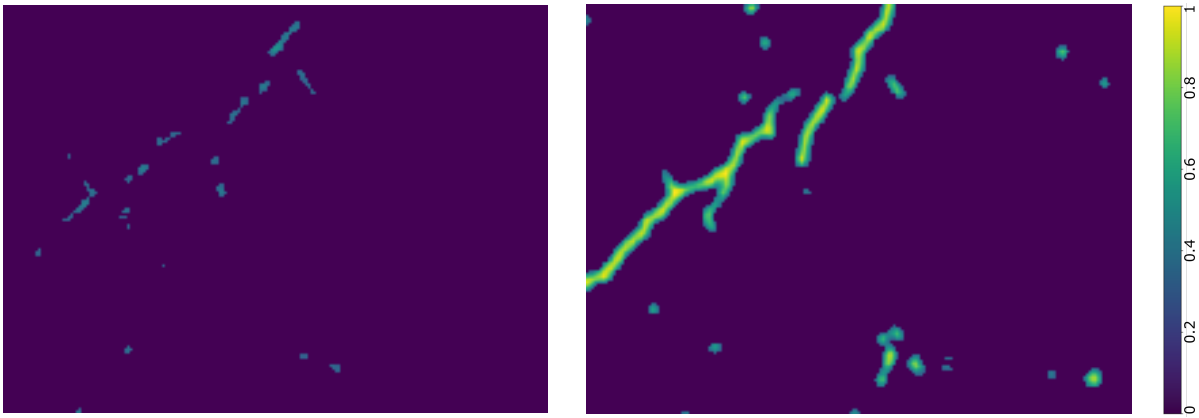
Figure 2.3: Illustration of the extended CODEBRIM pipeline for prediction of defect classes. The convolutional feature extractor and spatial pyramidal pooling correspond to the meta-learned architecture elements found in Mundt et al. (2019a). The decoder that is typically included in training to ensure encoding of the data distribution is not needed for generative factor based classification and is therefore not shown for simplicity. Figure adapted from Mundt et al. (2018a).

2.3.2 Variational Bayesian Meta-learned Architectures and Recognition of Statistically Deviating Open Set Images

The previous works in this chapter have introduced variational Bayes motivated deep generative model variants with open set recognition capacity (Mundt et al., 2019b, 2020b). These were not originally included in the purely discriminative meta-learned models of Mundt et al. (2019a). It is thus interesting to see how respectively obtained insights improve the just demonstrated semantic segmentation real world application. This is done by simply re-training the meta-learned architectures of the CODEBRIM dataset to include encoding of variational Bayes. The original prediction pipeline that includes a meta-learned neural network encoder, followed by a meta-learned spatial pyramidal pooling and classifier is thus adapted to also approximate the posterior distribution and minimizing its Kullback-Leibler divergence (KLD) with respect to a specified Gaussian prior. Note that the previously found architectures are simply extended and re-trained with a fixed 60-dimensional latent space, rather than conducting a novel architecture search that also attempts to find a suitable dimensionality for the latent factors. Such an architecture search that includes auto-encoding of variational Bayes is still a largely open research question. This is because the reinforcement learning reward can no longer consist solely of the architecture's classification accuracy, but would also involve additional terms such as reconstruction losses and the minimization of the KLD. The latter is problematic as it is a natural antagonist, i.e. a regularizer, that needs to be balanced with data likelihood and classification terms, even if encoder and decoder architectures are trivially coupled to mirror one another. Following the earlier made arguments about maintaining a desired mismatch between approximate



(a) Original image



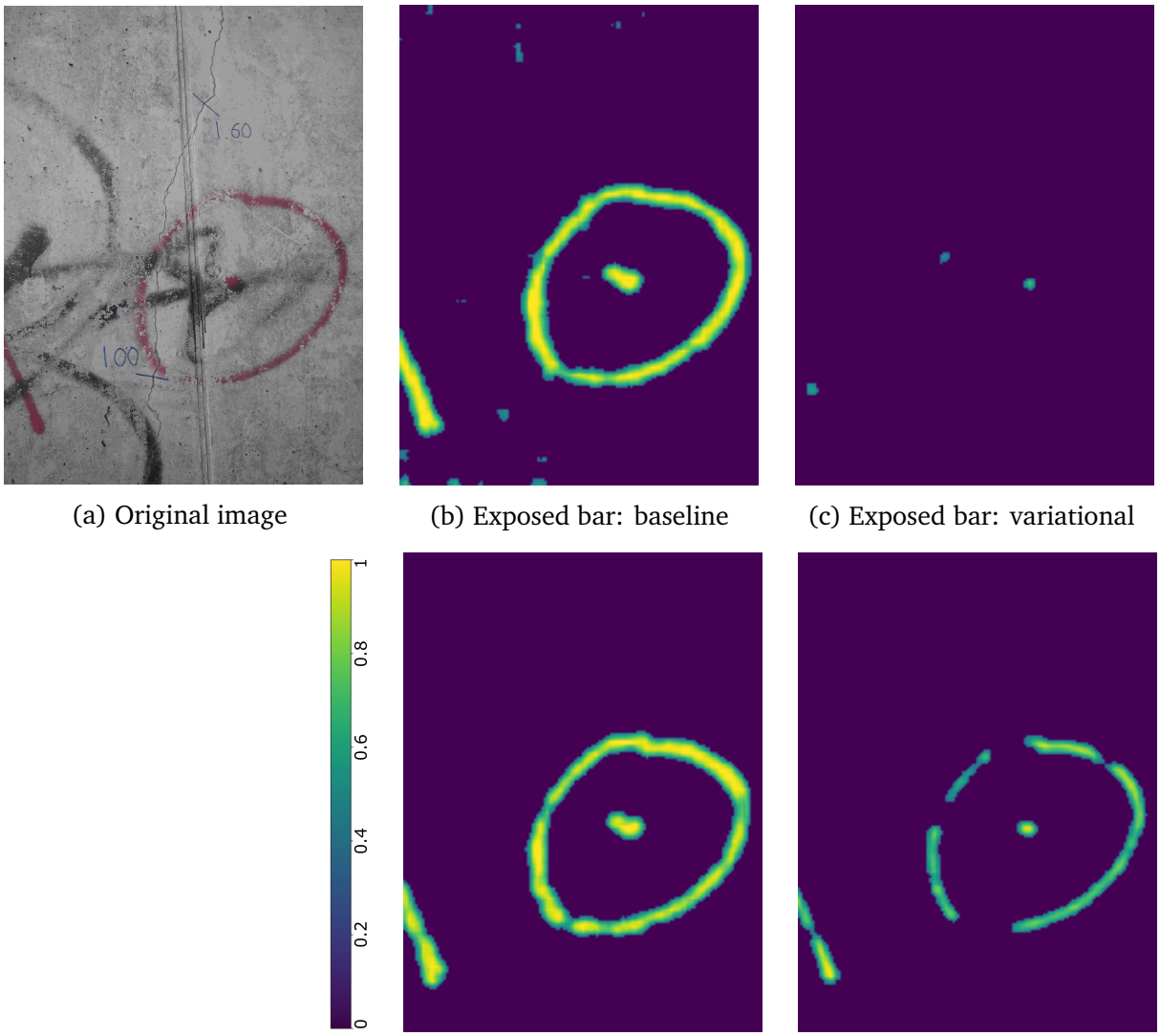
(b) Crack: baseline

(c) Crack: variational

Figure 2.4: Example for semantic segmentation through sliding window prediction of a deep neural network trained for image patch classification. The confidence maps for a pixel being attributed to each defect class are qualitatively contrasted for the conventional purely discriminative and the variational Bayesian generative model.

or aggregate posterior and prior, naively including a reinforcement reward signal to simply minimize the KLD can thus quickly lead to a collapse of the model, as any encoder can simply learn to project anything into a Gaussian distribution and discard the inherent structure of the data. The corresponding pipeline for prediction is visually summarized in figure 2.3, where the convolutional encoder and spatial pyramid correspond to the original meta-learned elements.

A respective qualitative improvement of the obtained segmentation in comparison to the former purely discriminative model, that does not explicitly care about the data distribution, is shown in figures 2.4 and 2.5. In the first example we can observe that a previously missed crack is now estimated considerably more accurately, alas without overcoming the inherent limitation of the patch based sliding window technique naturally overestimating the extent of the defect. In the second example we can see that the explicit encoding of the



(a) Original image (b) Exposed bar: baseline (c) Exposed bar: variational

(d) Corrosion: baseline (e) Corrosion: variational

Figure 2.5: Example for semantic segmentation through sliding window prediction of a deep neural network trained for image patch classification. The confidence maps for a pixel being attributed to each defect class are qualitatively contrasted for the conventional purely discriminative and the variational Bayesian generative model. The latter can be a remedy in further distinction of largely overlapping classes (top), but still regularly contains misclassification (bottom).

generative factors seems to aid in further separating the individual classes. Although the red graffiti present in the image is still largely misclassified as a corrosion stain, the latter seems to no longer be innately tied to the presence of an exposed bar. This can be intuitively understood when looking at the encoded latent space distribution. The respective 60 dimensional Gaussian mean and standard deviation vectors, aggregated and averaged over the encoding of the entire train dataset, are visualized in figure 2.6. Here, the distribution for some classes such as crack and efflorescence is immediately distinguishable. In contrast,

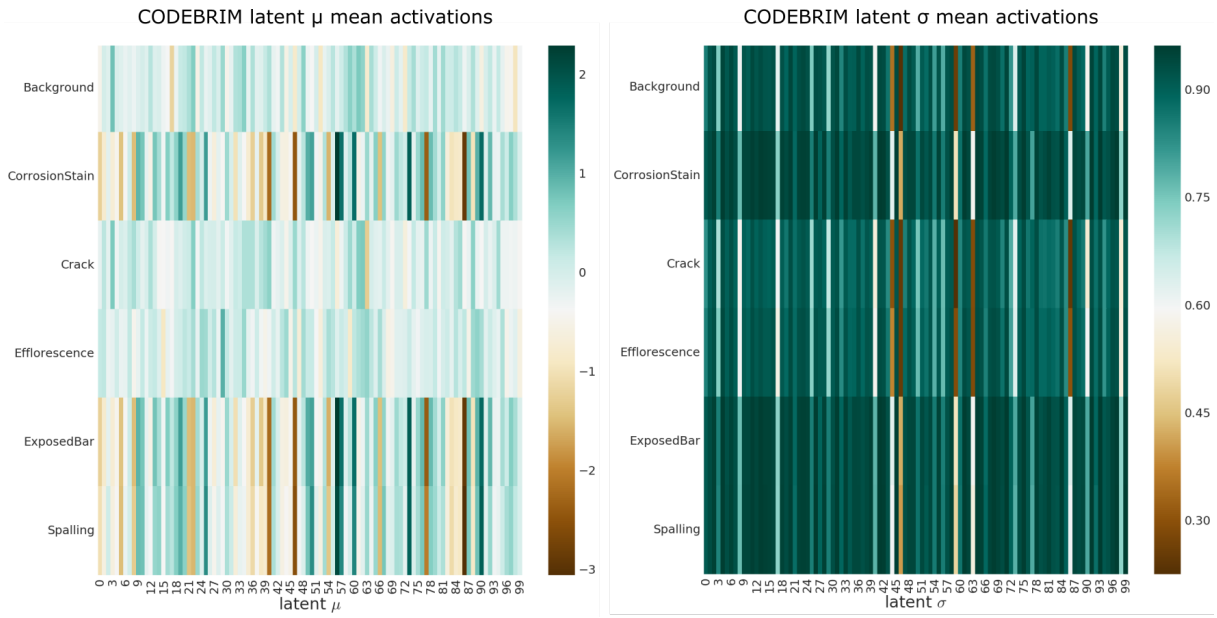


Figure 2.6: 60-dimensional Gaussian mean and standard deviation encodings averaged over the entire CODEBRIM training dataset. It is observable how some defect categories are clearly separable, whereas others have strong distributional overlap and distinction is based on nuances in particular modes. Figure taken from Mundt et al. (2018a).

the heavily interconnected classes of exposed bar, spalling and corrosion stain share many of the distribution's modes and deviate only lightly in particular dimensions.

Once again the generative model's latent space can now be used to construct extreme value theory based probabilistic meta-models to detect distribution outliers. Figure 2.7 shows an untangled illustration of the euclidean distance distribution to the just illustrated average latent class vectors for the popular ImageNet dataset (Russakovsky et al., 2015) and the individual CODEBRIM data instances for which the model has been trained. Based on much larger distances and discrepancy to the learned data distribution, the unseen unknown ImageNet classes are clearly discernible from the observed CODEBRIM data instances.

Naturally, this is a somewhat contrived example as it is perhaps not expected that a model employed for concrete infrastructure defect detection suddenly encounters images of e.g. the myriad of different dog categories comprising the ImageNet dataset. However, note that the figure is shown to further corroborate the quantitative intuition behind the method and recall that experiments of earlier work (Mundt et al., 2019b, 2020b) have shown that simple predictive heuristics would catastrophically fail, even on such a presumably trivial scenario. In practice, the mechanism can of course be applied to more meaningful data. Figures 2.8, 2.9 and 2.10 show three examples where the utility and necessity of the open set recognition is more natural. These examples have deliberately been picked to showcase prototypical use-cases. The first one, figure 2.8, features the generally desired scenario. For the majority, the

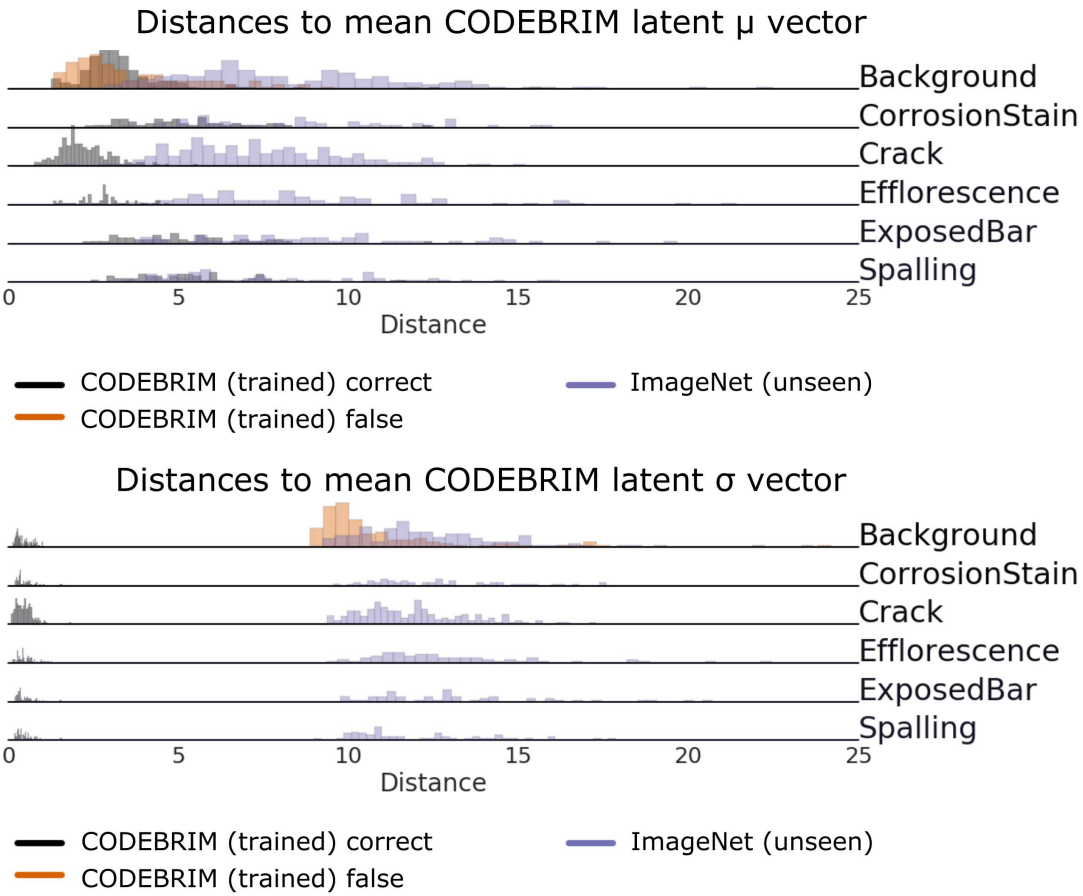
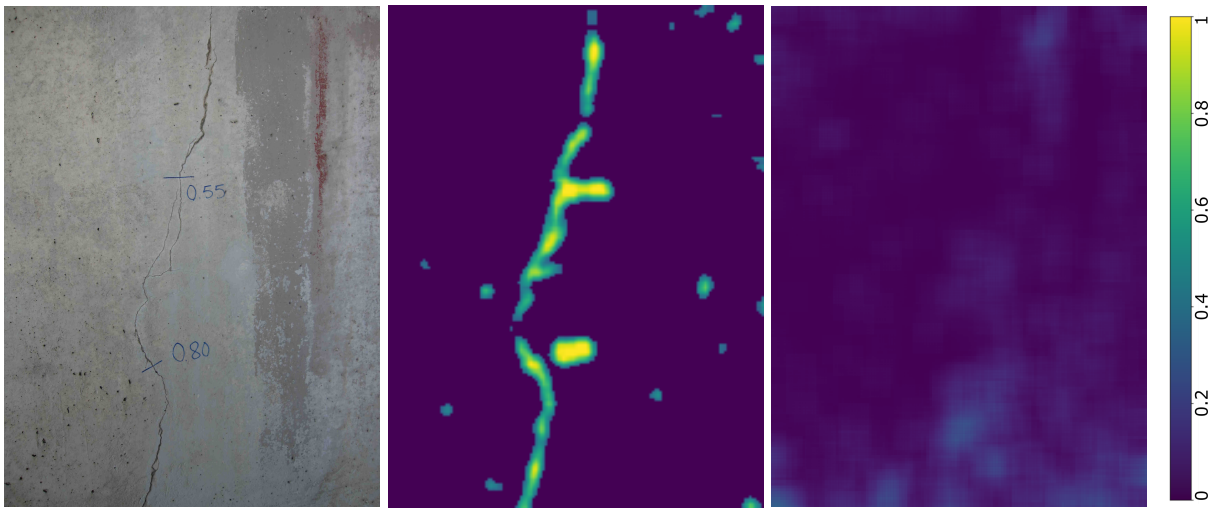


Figure 2.7: Per class euclidean distance distribution of processed CODEBRIM and ImageNet data instances to the average encoded Gaussian distribution mean and standard deviation vectors of a model trained on CODEBRIM. The unseen unknown ImageNet objects are clearly distinguishable from the trained material defects. Figure appeared originally in Mundt et al. (2018a).

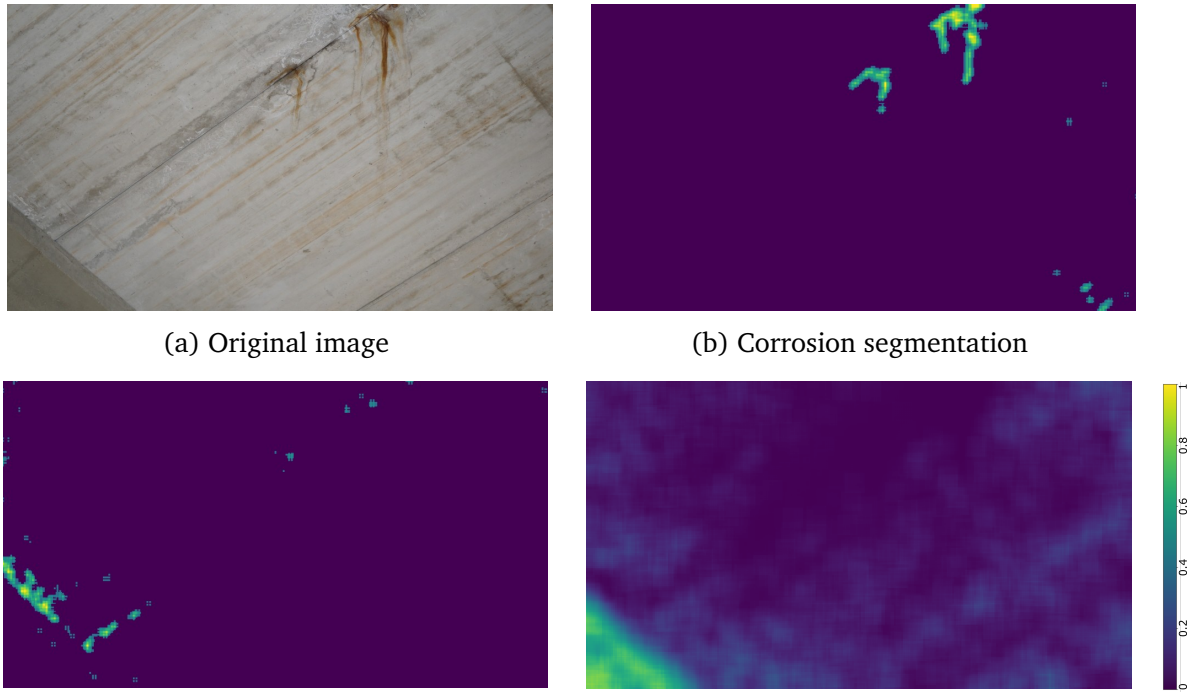
presence of a crack is correctly predicted and the open set recognition algorithm labels every region of the image as statistically similar to the data observed during training. The second example, figure 2.9, shows an instance where assigning an outlier likelihood corresponds to a partial remedy for robust prediction. It is the same image as previously shown in figure 2.1. A corrosion stain is correctly predicted and parts of the image are falsely identified as containing efflorescence due to surface markings. As the corner of the image is slightly blurred and significantly darker than the bright and homogeneously illuminated images of the CODEBRIM dataset, the corresponding area is flagged as statistically deviating. This could give a hypothetical explanation for the misprediction of efflorescence and crack at the border of this region. Alas, the false predictions for exposed bar and spalling would be similarly unaffected, see figure 2.1. This intuitively illustrates that not all false predictions can naively be associated with statistical deviations with respect to the training set and



(a) Original image (b) Crack segmentation (c) Open set outlier probability

Figure 2.8: Example of semantic segmentation with associated open set outlier probability. A crack defect is predicted and the open set probability map suggests that every region is statistically similar to the observed training distribution. Figure appeared originally in Mundt et al. (2018a).

inherent model limitations inevitably persist. The last example is the most obvious, but also crucial example that nevertheless needs to be intercepted. It is an instance where light shining directly into the camera at time of capture led to heavy over-exposure of the image. Although the presence of a crack is partially predicted correctly, there are also several other regions that lack correct output. Independently of whether the prediction of this particular instance is correct or not, prediction of such statistically completely deviating images, for which the model cannot be expected to perform well, should be discarded or at least set aside for human revision. Adoption of the proposed deep neural network models with open set recognition capability could thus be one aspect towards improved interaction between practical machine learning applications and human experts.



(a) Original image

(b) Corrosion segmentation

(c) Efflorescence segmentation

(d) Open set outlier probability

Figure 2.9: Example of semantic segmentation with associated open set outlier probability. Panels (b) and (c) correspond to the semantic segmentation shown in figure 2.1. The open set probability shows large outlier likelihood for the badly illuminated and slightly blurred image corner. This information is an indicator to flag the obtained false efflorescence segmentation as untrustworthy, whereas the correctly identified corrosion stain segmentation remains unaffected. Figure adapted from Mundt et al. (2018a).

(a) Original image

(b) Crack segmentation

(c) Open set outlier probability

Figure 2.10: Example of semantic segmentation with associated open set outlier probability. In this scenario, the presence of a crack is identified correctly in some parts, although heavily overestimated. However, the open set outlier probability flags the entire image as unseen and statistically deviating, likely due to the heavy camera over-exposure. Figure appeared originally in Mundt et al. (2018a).

Chapter 3

Consolidating Viewpoints: Designing Neural Networks for Continual, Active Learning in an Open World

A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning

Martin Mundt MMUNDT@EM.UNI-FRANKFURT.DE
Department of Computer Science and Mathematics
Goethe University
Frankfurt am Main, Germany

Yong Won Hong YHONG@YONSEI.AC.KR
Department of Computer Science
Yonsei University
Seoul, Republic of Korea

Iuliia Pliushch PLIUSHCH@EM.UNI-FRANKFURT.DE
Department of Computer Science and Mathematics
Goethe University
Frankfurt am Main, Germany

Visvanathan Ramesh VRAMESH@EM.UNI-FRANKFURT.DE
Department of Computer Science and Mathematics
Goethe University
Frankfurt am Main, Germany

Abstract

Current deep learning methods are regarded as favorable if they empirically perform well on dedicated test sets. This mentality is seamlessly reflected in the resurfacing area of continual learning, where consecutively arriving data is investigated. The core challenge is framed as protecting previously acquired representations from being catastrophically forgotten. However, comparison of individual methods is nevertheless performed in isolation from the real world by monitoring accumulated benchmark test set performance. The closed world assumption remains predominant, i.e. models are evaluated on data that is guaranteed to originate from the same distribution as used for training. This poses a massive challenge as neural networks are well known to provide overconfident false predictions on unknown and corrupted instances. In this work we argue that notable lessons from open set recognition, identifying unknown examples outside of the observed set, and the adjacent field of active learning, querying data to maximize the expected performance gain, are frequently overlooked in the deep learning era. Hence, we propose a consolidated view to bridge continual learning, active learning and open set recognition in deep neural networks. We empirically demonstrate resulting synergistic improvements when alleviating catastrophic forgetting, querying data, selecting task orders, while exhibiting robust open world application.

Keywords: Continual Deep Learning, Lifelong Machine Learning, Active Learning, Open Set Recognition, Open World Learning

1. Introduction

With the ongoing maturing of practical machine learning systems, the community has found a resurfacing interest in continual learning (Thrun, 1996a,b). In contrast to the broadly practised learning in isolation, where the algorithmic training phase of a system is constrained to a single stage based on a previously collected i.i.d. dataset, continuous learning entails a learning process that leverages data as it arrives over time. In spite of this paradigm having found various application in many machine learning systems, for a review see the recent book on lifelong machine learning by Chen and Liu (2017), the advent of deep learning seems to have steered the focus of current research efforts towards a phenomenon known as "catastrophic interference" or alternatively "catastrophic forgetting" (McCloskey and Cohen, 1989; Ratcliff, 1990), as suggested by recent reviews (Farquhar and Gal, 2018b; Parisi et al., 2019; De Lange et al., 2019; Lesort et al., 2020) and empirical surveys of deep continual learning (De Lange et al., 2019; Lesort et al., 2019; Pfülb and Gepperth, 2019). The latter is an effect particular to machine learning models that update their parameters greedily according to the presented data population, such as a neural network iteratively updating its weights with stochastic gradient estimates. When including continuously arriving data that leads to any shift in the data distribution, the set of learned representations is guided unidirectionally towards approximating any task's solution on the data instances the system is presently being exposed to. The natural consequence is superseding former learned representations, resulting in an abrupt forgetting of previously acquired information.

Whereas current works predominantly concentrate on alleviating such forgetting in continual deep learning through the design of specialized mechanisms, we argue that there is a growing risk towards a very different form of catastrophic forgetting, namely the danger of forgetting the lessons learned from past literature. Notwithstanding the commendable efforts towards preserving neural network representations in continuous training, such a high focus is given on the practical requirements and trade-offs beyond metrics that only capture catastrophic forgetting (Kemker et al., 2018), e.g. inclusion of memory footprint, computational cost, cost of data storage, task sequence length and amount of training iterations, ... (Díaz-Rodríguez et al., 2018; Farquhar and Gal, 2018b), that it could almost be seen as misleading when most current systems break immediately if unseen unknown data or minor corruptions are encountered during deployment (Matan et al., 1990; Boult et al., 2019; Hendrycks and Dietterich, 2019). The seemingly omnipresent assumption of a closed world, i.e. the belief that the model will always exclusively encounter data that stems from the same data distribution as encountered during training, is highly unrealistic in the real open world, where data can vary to extents that are impractical to capture into training sets or users have the ability to give almost arbitrary input to systems for prediction. In spite of the inevitable danger of neural networks generating entirely meaningless predictions when encountering unseen unknown data instances, a well known fact that has been exposed for multiple decades (Matan et al., 1990), current efforts towards benchmarking continual learning conveniently circumvent this challenge. Select exceptions attempt to solve the tasks of recognizing unseen and unknown examples, rejecting nonsensical predictions or setting them aside for later use, typically summarized under the umbrella of open set recognition. However, the majority of existing deep continual learning systems remain black boxes that

unfortunately do not exhibit desirable robustness to respective miss-predictions on unknown data, dataset outliers or commonly present image corruptions (Hendrycks and Dietterich, 2019).

Apart from current benchmarking practices still being constrained to the closed world, another unfortunate trend is a lack of understanding for the nature of created continual learning datasets. Both continual generative modelling, such as the works by Shin et al. (2017); Achille et al. (2018); Farquhar and Gal (2018a); Nguyen et al. (2018); Wu et al. (2018); Zhai et al. (2019), as well as the bulk of class incremental continuous learning works, such as presented by Li and Hoiem (2016); Kirkpatrick et al. (2017); Rebuffi et al. (2017); Lopez-Paz and Ranzato (2017); Kemker et al. (2018); Kemker and Kanan (2018); Xiang et al. (2019), generally investigate sequentialized versions of time-tested visual classification benchmarks such as MNIST (LeCun et al., 1998), CIFAR (Krizhevsky, 2009) or ImageNet (Russakovsky et al., 2015), where individual classes are simply split into disjoint sets and are shown in sequence. In favor of retaining comparability on a benchmark, questions about the effect of task ordering or the impact of overlap between tasks are routinely overlooked. Notably, lessons learned from the adjacent field of active machine learning, a particular form of semi-supervised learning, do not seem to be integrated into modern continual learning practice. In active learning the objective is to learn to incrementally find the best approximation to a task's solution under the challenge of letting the system itself query what data to include next. As such, it can be seen as an antagonist to alleviating catastrophic forgetting. Whereas current continual learning is occupied with maintaining the information acquired in each step without endlessly accumulating all data, active learning has focused on the complementary question of identifying suitable data for the inclusion into an incrementally training system. Although early seminal works in active learning have rapidly identified the challenges of robust application and pitfalls faced through the use of heuristics (Roy and McCallum, 2001; Settles and Craven, 2008; Li and Guo, 2013), the latter are nonetheless once again dominant in the era of deep learning (Beluch et al., 2018; Geifman and El-Yaniv, 2019; Gal and Ghahramani, 2015; Srivastava et al., 2014) and the challenges are faced anew.

In this work we make a first effort towards a principled and consolidated view of deep continual learning, active learning and learning in the open world. We start by providing a review of each topic in isolation and then proceed to identify previously learned lessons that appear to receive less attention in modern deep learning. We will continue to argue that these seemingly separate topics do not only benefit from the viewpoint of the other, but should be regarded in conjunction. In this sense, we propose to extend current continual learning practices towards a broader view of continual learning as an umbrella term that naturally encompasses and builds upon prior active learning and open set recognition work. Whereas the main purpose of this paper is not to introduce novel techniques or advocate one specific method as a universal solution, we adapt and extend a recently proposed approach based on variational Bayesian inference in neural networks (Mundt et al., 2019a,b) to illustrate one potential choice towards a comprehensive framework. Importantly, it serves as the basis of argumentation in an effort to illustrate the necessity of generative modelling as a key component in deep learning systems. We highlight the importance of the viewpoints developed in this paper with empirical demonstrations and outline implications and promising directions for future research.

2. Preamble: continual machine learning

It is likely that the idea of continual machine learning dates back to a similar period of time to the surfacing of machine learning itself. There has been many attempts at defining concepts such as continuous, lifelong or continual machine learning. Often these terms feature negligible nuances and can generally be taken as synonyms. However it seems difficult, and perhaps is not constructive, to attempt to pin-point the exact onset of when something should be referred to as continual or lifelong learning. Instead, in this section, we will present definitions and related paradigms that have come to enjoy great popularity in the machine learning community. Some of these paradigms are already, or if not yet, should be considered subsets of continual learning (CL) and as a standalone paradigm vary primarily in their current evaluation protocols. We will briefly introduce each of these paradigms and then proceed to summarize and identify characteristic differences with respect to the broader term of modern continual learning.

The first widely circulated definition of *lifelong machine learning* (LML) originated in the work proposed by Thrun (1996a,b). This definition is as follows:

Definition 1 Thrun (1996a,b) - Lifelong Machine Learning: *The system has performed N tasks. When faced with the (N+1)th task, it uses the knowledge gained from the N tasks to help the (N+1)th task.*

Here, the unmentioned quintessence is that the data of the first N tasks is generally assumed to be no longer available at the time of learning about the $N + 1$th task, i.e. observed data is not just endlessly accumulated and stored explicitly. While this definition captures the basic idea behind continued learning, it is also ambiguous with respect to the definition of task and knowledge. There has been many attempts to find a more concise definition across the literature over the years. One of the more succinct, yet still decently generic definitions followed in the work of Chen and Liu (2017):

Definition 2 Chen and Liu (2017) - Lifelong Machine Learning*: Lifelong Machine Learning is a continuous learning process. At any time point, the learner performed a sequence of N learning tasks, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$. These tasks can be of the same type or different types and from the same domain or different domains. When faced with the (N+1)th task $\mathcal{T}_{N+1}$ (which is called the new or current task) with its data $\mathcal{D}_{N+1}$, the learner can leverage past knowledge in the knowledge base (KB) to help learn $\mathcal{T}_{N+1}$. The objective of LML is usually to optimize the performance on the new task $\mathcal{T}_{N+1}$, but it can optimize any task by treating the rest of the tasks as previous tasks. KB maintains the knowledge learned and accumulated from learning the previous task. After the completion of learning $\mathcal{T}_{N+1}$, KB is updated with the knowledge (e.g. intermediate as well as the final results) gained from learning $\mathcal{T}_{N+1}$. The updating can involve inconsistency checking, reasoning, and meta-mining of additional higher-level knowledge.*

The authors of this latter definition argue that this definition can be summarized into three key characteristics: continuous learning; knowledge accumulation and maintenance in the knowledge base (KB); the ability to use past knowledge to help future learning. In contrast to the previous definition by Thrun (1996a,b), mainly the notion of a maintained

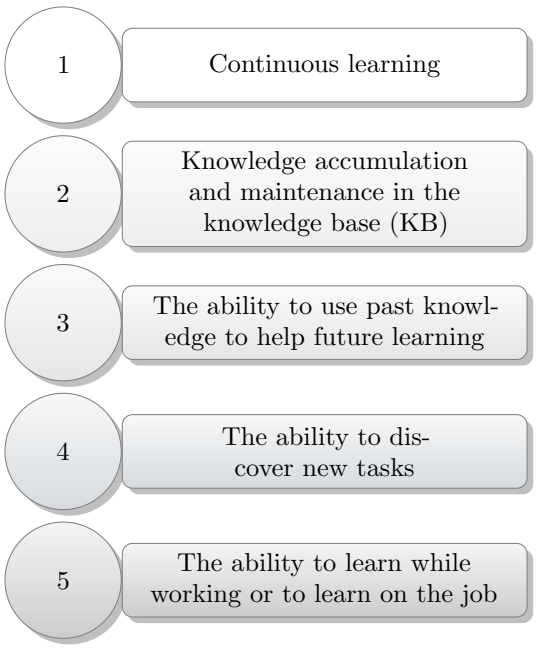


Figure 1: The five main pillars of lifelong machine learning according to Chen and Liu (2017). Note that the first three pillars were originally proposed and the last two added recently in a second edition redefinition to emphasize new frontiers.

knowledge base is introduced. Here LML is now defined such that at any given point in time performance can be optimized for any given task by treating all other tasks as previously presented, irrespective of their original order. Whereas the original definition unidirectionally optimized towards benefiting $\mathcal{T}_{N+1}$ and thus allowing for performance of previous tasks to degrade over time, Chen and Liu (2017) explicitly formulate the preservation of all accumulated information as a fundamental goal of LML. In a recent second iteration of this definition, the authors have added two additional desiderata: the ability to discover new tasks and the ability to learn while working. We have visualized these five essential pillars of LML in figure 1.

Although acknowledged by the authors themselves, this extended definition still lacks with respect to certain aspects:

- a coherent description of domain. This is currently not used unanimously in the literature and often applied interchangeably with task.

- a formalization of knowledge or respective representation thereof in the KB. Typically this is practically constrained to specific applications.

- the essential question of evaluation practice, i.e. choosing, ordering and evaluating the sequence of tasks. This generally requires a human in the loop and considered evaluation scenarios can vary immensely between individual works.

There are many more encountered open questions with LML in practice, especially with respect to modern machine learning algorithms based on deep learning. As the latter is primarily based on the use of neural networks (NN), they will constitute the main focus of this paper. While the presented arguments will often be of generic nature, this has the advantage that the concept of a knowledge base and its maintenance collapses to the question of managing the model's learned representations. At the same time, this can make the question of how to leverage prior information quite involved as representations in NNs are densely entangled within layers, as well as distributed hierarchically across layers. Before delving into a review of contemporary works, their merits and current limitations, we will present various popular paradigms that are related to the former definitions. This will then be followed by a brief summary on evaluation practices to highlight the nuances.

2.1 Related paradigms: subsets of continual learning

Over the course of machine learning development, various different paradigms and evaluation practices have evolved. Throughout this paper, we will come to the already apparent conclusion that CL should ideally be defined as a superset. We will make an attempt towards such a definition at the end of this manuscript. For now, we start by introducing commonly considered machine learning paradigms. As a word of caution, the following definitions should be regarded as non-exhaustive. Even though we have made a considerable effort to provide a comprehensive amount of references, the practical use of certain terminology in particular may still vary largely from community to community. The following shall thus reflect the common use in modern deep learning.

We begin with transfer learning as it can intuitively be regarded as the most related concept. Originally, transfer learning has been proposed as converting a weak learner, one that performs marginally better than random guessing, to one that produces stronger hypotheses (Schapire, 1990). The corresponding formulation that is more specific to neural networks is how the representations obtained by learning through backpropagation can be "recycled" for new tasks (Pratt et al., 1991; Pratt, 1993). This challenge initially wasn't unanimously referred to as transfer learning, but often was referred to as boosting (Freund and Schapire, 1997). A pre-deep learning survey (Pan and Yang, 2010) has summarized efforts and formalized transfer learning in the way used today:

Definition 3 Transfer Learning *(Pan and Yang, 2010): Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T,* transfer learning *aims to help improve the learning of the target predictive function $f_T()$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.*

Here, Pan and Yang (2010) formalize the use of the terms *domain* and *task* in the context of supervised transfer with datasets consisting of N data instances. They are defined by the following quote: *"Given a specific domain, $D = \{\mathcal{X}, p(\boldsymbol{x})\}$, a task consists of two components: a label space Y and an objective predictive function $f()$ (denoted by $T = \{Y, f()\}$), which is not observed but can be learned from the training data, which consist of pairs $\{\boldsymbol{x}^{(n)}, y^{(n)}\}$,*

where $\boldsymbol{x}^{(n)} \in X$ *and* $y^{(n)} \in Y$*"* (Pan and Yang, 2010). The concept of a domain is therefore defined as the pair of marginal data distribution $p(\boldsymbol{x})$ and a corresponding feature space $\mathcal{X}$. As it is generally implied that $\mathcal{X}_S \neq \mathcal{X}_T$ or respectively $p_S(\boldsymbol{x}) \neq p_T(\boldsymbol{x})$, an effortless translation of transfer learning to unsupervised or reinforcement learning settings is possible. Without further extensions, this definition of transfer learning is essentially a narrowed down version of the primitive lifelong learning definition 1, with the nuance that there typically only exist two tasks. It is similarly unidirectional in the sense that the source task is only used to improve learning the new target.

Since then an almost unending amount of works has sprouted, initiated by works that have started the investigation of transferability of deep neural network features beyond low-level patterns (Oquab et al., 2014; Yosinski et al., 2014), i.e. the higher abstractions and task-specific information believed to be encoded in deeper layers of the hierarchy. Weiss et al. (2016) have provided a survey on recent advances. In this context of feature transferability, a variant named *multi-task learning* (MTL) has emerged. Caruana (1997) summarizes the goal of MTL succinctly: *"MTL improves generalization by leveraging the domain-specific information contained in the training signals of related tasks"*. Early works sometimes referred to this as including "hints" (Suddarth and Kergosien, 1990; Abu-Mostafa, 1990) to improve learning. In contrast to transfer learning, generally multiple tasks are considered, with the requirement of the model performing well on all of them. However, in the MTL setting, tasks are all trained jointly and no sequence is assumed, corresponding to typical isolated learning practice. In modern day deep nets, MTL thus culminates in the question of how to exactly share the abundant amount of parameters in the architectural hierarchy, see e.g. the overview provided by Ruder (2017) for variants of sharing architecture portions.

More recently, a very specific form of transfer or multi-task learning has evolved. *Few-shot Learning* (Fei-Fei et al., 2006) developed due to the inability of deep learning techniques to cope with small datasets and empirical risk optimization being unrealiable in small sample regimes. Wang et al. (2020) summarized few-shot learning as a type of machine learning problem, where the dataset only contains a limited number of examples with supervised information for the target domain (and generally no constraints on the source domain). This implies that few-shot learning also tackles the issue of rare cases, apart from computational cost and the issue of data collection and labelling. When there is only one example with a label, it is commonly referred to as *one-shot learning* (Fink, 2005; Fei-Fei et al., 2006). Respectively, if no supervised example is provided, the scenario is referred to as *zero-shot learning* (Lampert et al., 2009). These scenarios are typically regarded under the hood of transfer learning with additional constraints on data availability.

Apart from concerns about reasonably sized datasets, a different concern is as old as the quest for stochastic approximations itself, namely when to conduct updates. Already in the work of Hebb (1949), *online learning*, i.e. incorporating information immediately as data arrives as opposed to collecting batches before updating a model, was a natural requirement. This question has been elemental in later formalization of frameworks for empirical risk optimization (Tsypkin, 1971; Vapnik, 1982). Several works have elaborated on challenges in online learning in NNs (Heskes and Kappen, 1993), more generally online learning and stochastic approximations (Bottou, 1999; Saad, 1999) or specifically online gradient descent (Zinkevich, 2003), the workhorse of modern optimization. Given the instance based update nature, online learning in neural networks is inherently tied to the question of how to avoid

catastrophic interference. It is thus not surprising that with the advent of DL immediate attempts have been made to consider online learning in DNNs (Zhou et al., 2012), see a recent survey by Sahoo et al. (2018), but the quest for online learning nevertheless still revolves around the interaction between online desiderata and stochastic approximations, or the stochastic gradient descent with backpropagation procedure in particular.

While each paradigm arose for a reason and comes with its own value, namely that of providing better distinction to other works in concrete evaluation scenarios, it is important to remember that the emerging taxonomy is full of nuances that are at times indistinguishable in a more general framework. In consequence, evaluation protocols are central to any discussion. We therefore proceed with details of common evaluation methods in deep continual learning and then summarize the main differences to the paradigms introduced in this section for a compact overview.

2.2 Continual learning evaluation

In contrast to isolated machine learning, where the evaluation scenario can often be defined in a straightforward manner by employing performance or satisfying task metrics, continual learning does not directly allow for such an approach. Given that the interest lies in accumulation of information, there are many factors to consider in evaluation of corresponding algorithms. In general it is important to monitor the currently introduced task, yet also investigate semantic drift on previous tasks. One should consider the gain and the ability to leverage representations from task to task in progressive experimentation, yet take note of the task sequence that is crucial to the specific solution obtained. When introducing more tasks, the transfer behavior should be carefully examined, yet interpretation should be treated with caution as not all introduced tasks yield immediate benefits and thus a larger amount of tasks needs to be brought in to the system.

Before continuing with the discussion of evaluation difficulties and metrics, let us take a brief look at some currently employed evaluation methodology (Chen and Liu, 2017), summarized visually in figure 2. It seems that such an evaluation protocol is still largely inspired by the isolated machine learning practices. Whereas the notion of information transfer and the sequence of tasks is considered and benchmarked against isolated learning algorithms, such an approach to evaluating the value of continual learning algorithms disregards the relevance of the task sequence (or permutation thereof), choice of tasks or choice of data. Accordingly, recently developed experimental protocols in deep continual learning (Farquhar and Gal, 2018b; Kemker et al., 2018; Parisi et al., 2019; De Lange et al., 2019; Lesort et al., 2019; Pfülb and Gepperth, 2019; Lesort et al., 2020) seem to mainly occupy themselves with evaluation procedures that are heavily inspired by decades of benchmarking learning algorithms in isolation. As a reminder to the reader, we refer to isolated learning as the practice of end-to-end training on a static dataset and evaluation on its predefined test set, sans changes over time. As such, the majority of current empirical examination equates continual learning benchmarks with the monitoring of catastrophic forgetting in scenarios that are simple sequentialized versions of popular datasets, similarly to the steps shown in figure 2. With few exceptions, this means that existing datasets are simply split into $t = 1, \ldots, T$ sets, where each of these sets is referred to as one task. These task- or time-stamped sets are then presented one by one to a deep learning system. Typically, each

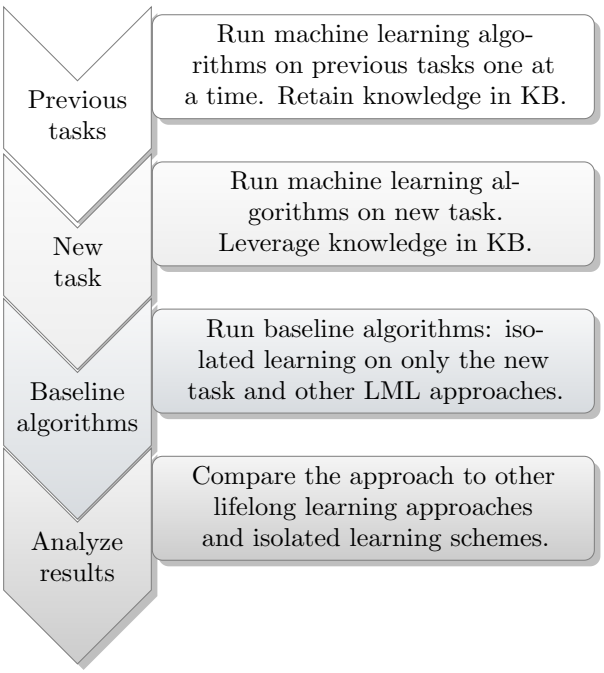


Figure 2: A widely used approach to evaluation of lifelong machine learning algorithms in the literature (Chen and Liu, 2017).

step is assumed to consist of a disjoint set of classes or entire datasets, usually independently of whether the probed task is of supervised, unsupervised or semi-supervised nature, see figure 3 for an illustration. Respectively analyzed metrics (Kemker et al., 2018) are based on this dataset sequentialization and routinely monitor e.g. the degradation of a first task's classification accuracy, the ability to encode new task increments, the overall development of a chosen metric as tasks accumulate or various similar measures to gain an intuition for generative models. It is obvious how this is inspired by isolated learning as these metrics can simply be extracted from a conventional confusion matrix. For this reason, multiple efforts have been made to emphasize the need for more diverse evaluation (Díaz-Rodríguez et al., 2018; Farquhar and Gal, 2018b). Alas the persisting focus on catastrophic forgetting remains visible from the formulated criteria and questions that are deemed necessary to compare methods (Díaz-Rodríguez et al., 2018; Farquhar and Gal, 2018b):

- **Memory consumption:** amount of required memory.
- **Amount of stored data:** how much past data does the method need to retain explicitly?
- **Task boundaries:** does the method require clear task divisions?
- **Prediction oracle:** does the method require knowing the task label for prediction?

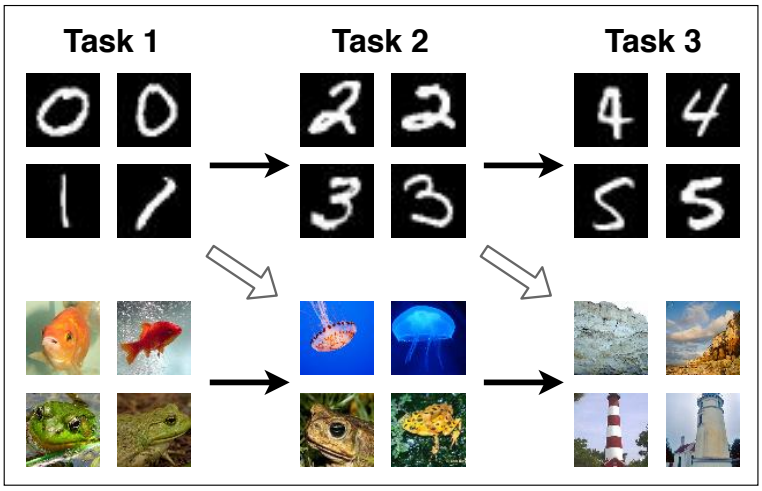


Figure 3: A typical continual learning scenario dividing common benchmark datasets into a sequence of sub-tasks. Here, the digits one through six from the MNIST dataset (LeCun et al., 1998) and the Wordnet ids "n01443537": goldfish, "n01641577": bullfrog, "n01644900": tailed frog, "n01910747": jellyfish, "n09246464": cliff, "n02814860": beacon from the ImageNet dataset (Russakovsky et al., 2015). Common evaluation either follows the filled dark arrows to incrementally learn one dataset or alternatively also switches dataset, as denoted by the hollow light arrows.

- **Amount of forgetting:** how much information is retained as measured through proxy metrics.
- **Forward transfer:** do older tasks accelerate learning of new concepts?
- **Backward transfer:** do new tasks benefit old tasks?

At this stage the reader might already notice that some of these listed items are very particular to specific practices. For example, the idea that a prediction oracle would be required in the first place in order to give task labels is an artefact of several works that consider so called multi-head scenarios. The latter makes use of separate disconnected classifiers per task to circumvent explicitly dealing with task prediction interdependence. There exist recent reviews (De Lange et al., 2019) that base their entire evaluation on such a scenario. Empirical surveys in the context of robotics (Lesort et al., 2019), generative models (Lesort et al., 2020) follow similar trends and conduct a "comprehensive application-oriented study of catastrophic forgetting" (Pfülb and Gepperth, 2019). With catastrophic forgetting being the sole focus, these works at best cover the first three of the five earlier formulated continual learning pillars 1, if and only if they also conduct an analysis on how specific tasks benefit each other. The recent critiques that formulated above questions (Díaz-Rodríguez et al., 2018; Farquhar and Gal, 2018b) therefore present valid attempts to rid current evaluation from such practices that can be seen as inherently violating real continual learning scenarios. Nevertheless, we argue that there is even larger factors at play that transcend these arguments. Although transfer and the sequential nature is considered

and benchmarked against isolated learning, crucial aspects such as the *relevance of the task order* (or permutation thereof), *choice of tasks*, *choice of data* and particularly any form of *robustness* in an open world and with respect to perturbations or attack scenarios are disregarded altogether. Open research areas such as curriculum learning (Bengio et al., 2009), i.e. benefiting from a data ordering of increasing complexity, open world learning (Bendale and Boult, 2016), i.e. equipping the model with awareness of unseen unknown data, and active learning, i.e. self-selecting data to query for the next step, try to address these crucial elements. We argue that it is imperative to take these perspectives into account in the evaluation of continual learning algorithms. Before proceeding to categorize individual works and consequently making an attempt at connecting the paradigms, we give a brief summary of the present evaluation differences.

- **Transfer Learning:** Leverage a source task's representations to accelerate learning or improve a current target task.
  Difference to CL: unidirectional knowledge transfer between two tasks.
- **Multi-task Learning:** Exploit tasks relatedness by forming a joint hypothesis space.
  Difference to CL: isolated learning with multiple tasks
- **Online Learning:** Retaining and improving a task where data arrives sequentially and real-time constraints require online adaptation.
  Difference to CL: typically continued learning of one task over time, however generally applicable to any paradigm.
- **Few-shot Learning:** Transfer or multi-task learning in a small data regime.
  Difference to CL: unidirectional transfer or isolation similar to transfer or multi-task learning.
- **Curriculum Learning:** Finding a suitable curriculum that accelerates or improves training by means of introducing schedules of increasing data instance difficulty or data instance task specificity.
  Difference to CL: isolated learning that prioritizes certain data instances
- **Open World Learning:** At any particular point in time the model needs to be able to identify and reject unseen data belonging to unknown tasks. These could be set aside and learned at a later stage.
  Difference to CL: Current CL is typically evaluated in a closed world scenario.
- **Active Learning:** An iterative form of supervised learning, where the learner can query a user to provide labels for a subset of unlabelled examples that are deemed to yield the largest knowledge gain.
  Difference to CL: data and sampling efficiency is rarely taken into account in CL on predefined benchmarks.

3. An overview and review of three perspectives

We provide a review of the plethora of practices and historically grown methods in the context of deep continual learning, active learning and open set recognition. What may at

first seem like a tour de force review for the reader, is intended to first gain an overview of the vast landscape and the deluge of options. This will aid in delving into details of potential pitfalls and shortcomings, but also in highlighting synergies and the necessity for a consolidated view in consecutive sections. As the latter is the primary focus of this work we will limit our survey to concise summaries and will forgo lengthy elaborations on methodological details that are not essential to a generic understanding.

3.1 Continual learning

As indicated in the introductory section, continual learning should ideally encompass a variety of research questions. Whereas our next section will continue to argue that currently considered scenarios are too reductive, resulting in potential difficulty to chose among existing algorithmic options, we will stick to the typical categorization of existing deep continual works into the three categories of *regularization, rehearsal* and *architectural* approaches, in consistency with recent reviews (Parisi et al., 2019; De Lange et al., 2019; Lesort et al., 2020). We note that a strict organization into these groups is not always possible and hence also provide a forth category for works that combine multiple methods. In later sections we will argue that this is not only advantageous, but conceivably a necessity.

3.1.1 Regularization:

Continual learning approaches based on regularization aim to strike a balance between protecting already learned representations, while granting sufficient flexibility for new information to be encoded. Intuitively, a meaningful balance should be attainable for tasks with sufficient overlap in their high dimensional embeddings, i.e. if a considerable amount of the learned representations are shareable. Existing approaches can be further subdivided into regularization that explicitly protects parameters, which we refer to as *structural*, which constrains changes on every level of a model architecture, or *functional*, that is preserving a model's output for seen tasks while ensuring full adaptability with respect to each individual model stage that leads to the prediction.

Structural Inspired by the neuroscientific stability-plasticity dilemma (Hebb, 1949), successful use of regularization of deep learning models for continual learning requires carefully balancing the trade-off between overwriting acquired representations in favor of sensitivity to new information and preservation of already existing formed patterns. Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) aims to achieve this balance by estimating each parameter's importance through the use of Fisher information and respectively discouraging updates for parameters with greatest task specificity. Synaptic Intelligence (SI) (Zenke et al., 2017) and Memory Aware Synapses (MAS) (Aljundi et al., 2018), where the biologically inspired term synapse is used synonymously with parameter, follow a similar approach by explicitly equipping each parameter with additional importance measures that keep track of past improvements to the objective. Asymmetric Loss Approximation with Single-Side Overestimation (ALASSO) (Park et al., 2019) can be seen as a direct extension to SI and aims to mitigate its limitations by introducing an asymmetric loss approximation that is motivated from empirical observations. Riemannian Walk (RWalk) has generalized EWC and SI by taking into account both the Fisher information based importance, from a perspective of computing distances in the induced Riemann manifold, and the optimization trajectory

based importance score. Incremental Moment Matching (IMM) (Lee et al., 2017) approaches structural regularization from a perspective of Bayesian approximations and matching the moments of tasks' posterior distributions. Uncertainty based Continual Learning (UCL) (Ahn et al., 2019) makes use of Bayesian uncertainty estimates to adaptively regularize weights online. Similarly, Uncertainty-guided Continual Bayesian Neural Networks (UCB) (Ebrahimi et al., 2020) adapts the learning rate in dependence on the uncertainty defined in the probability distribution of the weights.

Functional Functional regularization approaches are generally inspired by "knowledge distillation" (Hinton et al., 2014), an approach originally proposed for model compression. A distillation loss is introduced by storing the prediction of a data sample for future use as a so called soft target. In learning without forgetting (LWF) (Li and Hoiem, 2016) for class incremental continual learning, the soft targets for existing classes are calculated using newly arriving data, even if these predictions might be nonsensical as the freshly added classes do not get correctly predicted yet, in hopes of regularizing towards preserving the output for old tasks. Encoder based lifelong learning (EBLL) (Rannen et al., 2017) applies this concept to the unsupervised learning scenario by applying distillation to autoencoder reconstructions. Knowledge distillation seems to rarely be employed in isolation, but as will be apparent from the list of upcoming combined approaches is a popular technique in conjunction with other mechanisms.

3.1.2 Rehearsal:

As the name implies, rehearsal techniques for continual learning aim to preserve encoded information by replaying data from already seen tasks. Trivially, continual learning could be solved by simply storing and replaying all seen data, albeit at usually intolerable memory expense and growing computation time. Accordingly, a core aspect of rehearsal methods is to find a suitable subset of data that best approximates the entire observed data distribution, commonly referred to as selection of exemplars or construction of a core set. Alternatively, a generative modelling approach can be used to generates instances from a learned latent representation as an encoding of the observed data distribution. Most replay techniques indicate their inspiration to be drawn from the complex biological interplay between hippocampus and neocortex, wake + sleep cycles and dreaming in the brain.

Exemplar Rehearsal GeppNet (Gepperth and Karaoguz, 2016) explores the use of a dual-memory system that implements various short and long-term memory storages that serve to store newly arriving information or provide dedicated replay cycles of previously stored data. Selective experience replay (SER) (Isele and Cosgun, 2018) concentrates on exemplar selection techniques and investigates trade-offs between preferring surprising experiences over rewarding ones, or maximizing distribution coverage. Gradient Episodic Memory (GEM) (Lopez-Paz and Ranzato, 2017) extends the use of a memory that gets replayed episodically with constraints on the gradients to be non-conflicting with updates for previous tasks. A respective extension called Averaged Gradient Episodic Memory (A-GEM) has introduced significant improvements on computational and memory cost for optimization under these constraints. CLEAR (Rolnick et al., 2018) uses experience replay together with off-policy learning to preserve old information and on-policy learning to learn new experiences in deep

reinforcement learning. Bias Correction (BiC) (Wu et al., 2019) rehearses exemplars and additionally corrects for biases in the classification layer.

Generative Generative replay is a specific version of rehearsal where the data to be rehearsed consists entirely of instances sampled from a generative model. Rather than making use of an episodic memory of previously seen data, generated samples of former tasks are typically interleaved with the current task's real data during training. The most elementary version of this procedure was coined pseudo-rehearsal (Robins, 1995), where the generative model is of simple nature. Here, binary patterns are sampled at random, their target value or label computed given the current state of the classifier, and the classifier then needs to maintain the discrimination on these patterns and learn new classes. Such pseudo-rehearsal has then successfully been leveraged in brain-inspired dual-memory architectures that use two distinct networks for acquisition and storage of information with generative rehearsal to consolidate the memory. Two early examples include pseudo recurrent networks (French, 1997) and coupling two reverberating neural networks (Ans and Rousset, 1997). Deep Generative Replay (DGR) (Shin et al., 2017) have introduced a deep learning variant of this practice, where the generative model is taken to be a separate generative adversarial network (Goodfellow et al., 2014) that gets trained in alternation with a classification model. Replay through Feedback (RfF) (van de Ven and Tolias, 2018) proposed generative replay using a single model that handles both classification and generation through the aid of feedback connections. Incremental learning using conditional adversarial networks (ILCAN) (Xiang et al., 2019) follows a similar approach of using a single model, but additionally changes the generative replay component to rehearse feature embeddings instead of aiming at reconstructing original input data. Open-set Classifying Denoising Variational Auto-Encoder (OCDVAE) (Mundt et al., 2019a) further introduces the first approach to naturally integrate open set recognition with deep generative replay in a single architecture. This work will play a vital role for the remainder of this paper and we will demonstrate how suggested ideas can be extended to form one potential basis as means to broaden current continual learning practices.

3.1.3 Architectural:

Architectural approaches attempt to alleviate catastrophic forgetting through modification of the underlying architecture. It might at this point be baffling to the reader why such modifications are listed distinctly from the works presented in previous subsections as they are almost by definition complementary to any method presented so far, and in fact most methods presented in this paper. For historical reasons, we will however stay consistent with former categorization of deep continual learning algorithms (Parisi et al., 2019). The importance of choice of architecture and the need for modifications over time will be another element of our upcoming proposition on an expanded view of continual learning. We will sub-categorize architectural approaches further into implicit and explicit architecture modification, i.e. methods that use a fixed amount of maximum representational capacity and methods which dynamically increase capacity in the process of continued training.

Fixed maximum representational capacity Approaches that use a static architecture rely on task specific information routing through the architecture. An early example is a technique coined activation sharpening towards semi-distributed representations (French,

1992), where the essence is to tune and limit the amount of high neural network activations to a maximum of k nodes, such that there is less activation overlap for different representations and consequently less potential for interference of new examples. While fixed architecture methods differ in the specifically employed technique to disambiguate the learned dense representations, the common denominator is the assumption of an over-parametrized architecture in order to warrant enough initial redundancy to permit overriding parameters without incurring catastrophic interference. PathNet (Fernando et al., 2017) adopted this notion to deep neural networks and used a genetic algorithm to determine pathways through the network deemed particularly useful for a specific task in order to freeze them. Instead of using a separate algorithmic layer to determine task specific network subsets, Piggyback (Mallya et al., 2018) and hard attention to the task (HAT) (Serra et al., 2018) directly learn binary masks and use them to gate information propagation through the network. The UCB-P variant of the earlier introduced regularization approach Uncertainty-guided Continual Bayesian Neural Networks (UCB) (Ebrahimi et al., 2020) confronts this challenge from a Bayesian perspective. They use uncertainty to prune the model and identify binary masks per task to index into the weights' Gaussian mixture distributions.

Dynamic growth Dynamic growth approaches administer representational capacity much more explicitly. The trivial solution would be to simply have one model per task and devise a mechanism to select the appropriate path for an input. Alas, such an arrangement doesn't fully leverage information from one task to positively transfer to another or respectively newly arriving information to aid already acquired tasks. First works in deep learning however nearly follow this naive but also intuitive approach to simply train on a task and consequently freeze all learned representations, such as demonstrated in Progressive Neural Networks (PNN) (Rusu et al., 2016). The amount of weights is then increased for a new task, with the twist that formerly learned representations laterally transmit their output to the new tasks' representations but not vice versa. Expert Gate (Aljundi et al., 2017) is comparable and differs mainly in the introduction of a gating mechanism that automates the choice of a suitable expert in an ensemble. Recent perhaps more practical approaches can be viewed as once again drawing their inspiration from decades of biological findings and discussion on neurogenesis. The latter refers to the process of creation and incorporation of new neurons into the existing system, see the reviews by Aimone et al. (2014); Vadodaria and Jessberger (2014). For the last two decades it has now been acknowledged that this process persist beyond early stage human development and continues its function in adults (Gross, 2000). The seminal work of dynamic node creation in neural networks (Ash, 1989), where additional units are added whenever the loss plateaus, has thus found a renaissance in modern deep learning. Neurogenesis deep learning to accomodate new classes (NDL) (Draelos et al., 2017) and lifelong learning with Dynamically Expandable Networks (DEN) (Yoon et al., 2018) have adapted this heuristic approach for use in continual deep learning. The former by adding units whenever the reconstruction error of an autoencoder surpasses a predetermined threshold in the spirit of Zhou et al. (2012), the latter based on an empirically found value of the classification loss in supervised learning. Reinforced Continual Learning (RCL) (Xu and Zhu, 2018) or Learn-to-Grow (Li et al., 2019a) further attempt to overcome the challenge of finding suitable loss cut-offs and cast dynamic unit addition into a meta-learning framework in order to separate the learning of the network structure and estimation of its parameters.

3.1.4 Combined Approaches:

We list a number of, largely very recent works, that primarily advance the state of the art on a set of benchmark datasets by blending techniques from the previous categories. One of the most popularly cited works is iCarl (Rebuffi et al., 2017), which couples a knowledge distillation based regularization approach with rehearsal of exemplars, assembled through a greedy herding procedure (Welling, 2009). Variational Continual Learning (VCL) (Nguyen et al., 2018) similarly fuses use of an episodic memory of exemplars with parameter regularization, but from a perspective of approximate Bayesian inference. FearNet (Kemker and Kanan, 2018) has later critiqued iCarl as a viable technique due to its heavy dependency on quantity of data in order to be successful. They have therefore additionally incorporated generative rehearsal to compensate the need to store large subsets of the original dataset. Variational Generative Replay (VGR) (Farquhar and Gal, 2018a) can be seen as concurrent to VCL, where instead of exemplar rehearsal generative replay is made use of. Memory replay GAN (MRGAN) and Lifelong GAN (LLGAN) (Zhai et al., 2019) are recent complements to these works and deviate in that they are based on GANs instead of variational inference in pure autoencoders. Whereas MRGAN uses a functional regularization approach to align the generator's output, LLGAN further applies such distillation loss based regularization across multiple places in the architecture to regularize encoders and discriminators. On the architectural front, Variational Autoencoder with Shared Embeddings (VASE) (Achille et al., 2018) adopts dynamic architecture growth in conjunction with generative replay. Their proposal is to allocate additional representational capacity for new concepts, determined through larger reconstruction loss in a variational autoencoder, however, is limited to expanding the latent space and leaving the rest of the architecture static. Lifelong Learning for Recurrent Neural Networks (LLRNN) (Sodhani et al., 2019) combines training of long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) with gradient episodic memory based exemplar rehearsal and a capacity expansion approach named Net2Net (Chen et al., 2016), which provides the means to transfer learned representations from an architecture to a larger untrained one before continuing to train the latter. While some of these works clearly exploit natural synergies, a generally desirable practice, we note that this can sometimes come at the expense of detailed analysis and comprehensive understanding of individual key ingredients and their necessity. While we agree that all approaches in this subsection pursue commendable directions, we argue that considerable future analysis is still required. We will discuss corresponding details and suggestions in later sections.

3.2 Active learning

Rather than focusing on the question of how to preserve representations in incremental continual learning, the topic of active learning asks the reverse question of how to pick data increments for future inclusion. Generally, this is cast into the framework of semi-supervised learning. Here, it is assumed that the model is trained on labelled data $\boldsymbol{X}_L = \{\boldsymbol{x}_L^1, \ldots, \boldsymbol{x}_L^n\}$, and a larger pool of unlabelled data $\boldsymbol{X}_U$ exists. This is motivated from data acquisition being relatively cheap in the modern world, as opposed to human intensive data labelling that often requires highly skilled experts. The task of an active learner is thus to extract a set of M data instances $\{\boldsymbol{x}_U^1, \ldots, \boldsymbol{x}_U^m\}$ from the pool of unlabelled data, such that a maximum gain in performance on the inspected task is expected if a human in the loop provides the

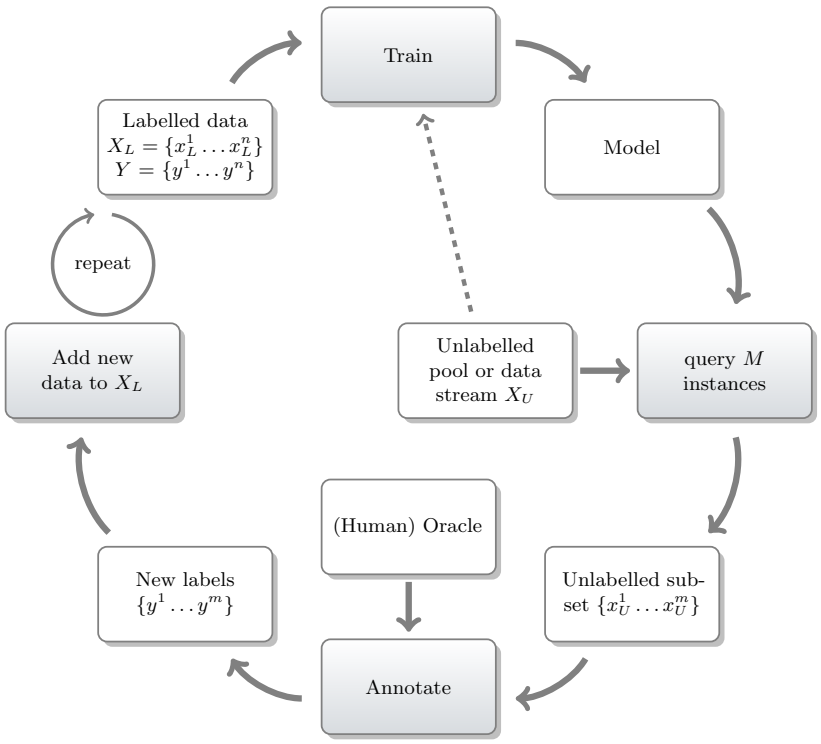


Figure 4: Active learning cycle that repeatedly expands a labelled dataset by querying and then annotating data instances from a larger unlabelled pool. The dashed arrow from the latter to the training process indicates the common closed world active learning scenario, where the presence of all data at all times is assumed. Respective works typically include the entire unlabelled dataset into the training procedure by employing methods from semi-supervised learning. Shaded parts of the diagram correspond to processes, whereas light components represent objects.

additional labels $\{y^1, \dots, y^m\}$ for further training. The underlying mechanism on which the query is based is referred to as the acquisition function and forms the main pillar of active learning research. We have visualized this active learning cycle in figure 4.

There is multiple conceivable evaluation variants to gauge the usefulness of active learning acquisition function choices. They either explicitly assume the entirety of the unlabelled data to be accessible and usable upfront, or contrarily the query being informed solely by the available labelled data. Independently of the latter, the practical assessment of active learning strategies is generally conducted in a closed world scenario, i.e. the entire pool of unlabelled data is expected to stem from the same data distribution as the initially labelled set and the oracle is assumed to be infallible. In a crucial distinction to continual learning, evaluation of active learning however accumulates data and grows the labelled set, focusing primarily on the cost reduction of labour intensive annotation. In consequence, an

active learner is deemed successful if each data query provides significant benefit over simply picking and labelling data at random.

"A probability analysis of the value of unlabelled data for classification problems" (Zhang and Oles, 2000) provides an early analysis of the requirements for benefiting from semi-supervised or active learning approaches. The authors consider two types of models: parametric $p(\boldsymbol{x}, y|\boldsymbol{W}) = p(\boldsymbol{x}|\boldsymbol{W})p(y|\boldsymbol{x}, \boldsymbol{W})$ and semi-parametric: $p(\boldsymbol{x}, y|\boldsymbol{W}) = p(\boldsymbol{x})p(y|\boldsymbol{x}, \boldsymbol{W})$. In the latter, the data probability $p(\boldsymbol{x})$ is decoupled and can have an unknown (or non-parametric) form independent of the weights $\boldsymbol{W}$, as is common in most discriminative models such as logistic regression or most neural networks. They argue that these models are particularly suited for active learning, as opposed to parametric models such as Gaussian mixtures being particularly suitable for semi-supervised learning. This is because they do not need to rely on potentially inaccurate estimates of the entire data distribution when only a fraction of the data is observable. However, we will see in the subsequent review that both of these model types have been used to form different perspectives to address active learning and come with their respective advantages.

As with the majority of techniques, early active learning methods have rapidly cross-pollinated into applications with deep neural networks. However, due to the black-box nature of deep non-linear neural networks, many of these approaches are based on simple heuristics or approximations to uncertainty quantities that no longer have tractable closed-form solutions. We will start with these heuristic approaches, as they are often trivial to transfer to deep learning, and then continue to summarize more principled approaches, which can turn out to be genuinely challenging in the context of deep learning.

3.2.1 Uncertainty Heuristics

One theoretically sound approach to querying useful data is based on entropy (Shannon, 1948) sampling and other information theoretic acquisition functions (MacKay, 1992). An early approach based on training two neural networks to estimate query areas in binary classification problems (Atlas et al., 1990) remarks that this is difficult for neural networks as they are often overly confident in their outputs. This overconfidence is going to be one of the main subjects of our next major section on learning in an open world. Interestingly, while paid painstaking attention in early literature, this aspect seems to often be overlooked in the era of deep learning. Simply using neural network prediction confidence, predictive entropy or other derived heuristics (Lewis and Gale, 1994) are still practically employed in comparisons today (Geifman and El-Yaniv, 2019). This is because many approaches have been shown to empirically work well in specific contexts, although there is no guarantee for them to succeed. Early works have shown uncertainty sampling based active learning for logistic regression (Lewis and Gale, 1994) and neural networks (Seung et al., 1992; McCallum and Nigam, 1998) based on "query by committee", an approach to estimate uncertainty by using an ensemble of neural networks. This idea has later found a one-to-one translation to deep ensembles for active learning (Beluch et al., 2018). Naturally, most black-box deep neural networks are not equipped with mechanisms to gauge uncertainty properly outside of using multiple parallel models. Bayesian active learning by disagreement (BALD) therefore provides an attempt at avoiding the necessity of ensembles and instead uses Monte Carlo Dropout (Gal and Ghahramani, 2015; Srivastava et al., 2014) to calculate points of high

variance in the output (Gal et al., 2017). This has empirically been demonstrated to be effective and has been extended in Bayesian Generative Active Learning (BGAL). Here, BALD is used to query samples and then the labelled set is further augmented with generated examples (Tran et al., 2019). Deep incremental learning with Neural Architecture Search (iNAS) (Geifman and El-Yaniv, 2019) does not propose a new query mechanism and instead provides an evaluation of above acquisition functions in the context of architecture selection. They include the option of progressive architecture growth after each query, to illustrate that small models generally fare better in a small data regime, whereas large models are required when a certain degree of task complexity is reached. We will revisit this as an imperative insight in our later discussions.

3.2.2 Version Space and Expected Error Reduction:

A theoretically more substantiated approach to basing the acquisition function on heuristics is to query data that provably reduces the expected error. Clearly, this is beyond the current understanding of deep neural networks, but has been shown to be feasible in the context of parametric models such as Gaussian mixture models (Cohn et al., 1996) or naive Bayes (Roy and McCallum, 2001). These works use the concept of a version space (Mitchell, 1982), i.e. the consistent set of hypotheses that separate the data in the induced feature space. An appropriate active learning strategy is to sequentially and monotonically reduce the size of this version space. In models such as SVMs for binary classification this is intuitively explained based on the margins (Tong and Koller, 2001), where new points are chosen according to hyperplanes that maximize the restriction with respect to the set of possible hyperplanes for correct classification. The latter was later extended to a multi-class SVM based approach (Joshi et al., 2009), however still based on multiple binary classifiers. This allowed for theoretical guarantees on sample complexity and necessary amount of queries to be analyzed with respect to these binary classification problems with linear decision boundary in the context of greedy active learning strategies (Dasgupta, 2005). Whereas "learning active learning from data" (Konyushkova et al., 2017) provides a recent effort to train a meta-learning based regressor to predict expected error reduction for binary classification using random forests, the idea has not been adapted to deep neural networks yet.

3.2.3 Representation based approaches:

Although version space reduction can come with provable guarantees, respective application to deep neural networks is inconceivable before a mature theory of how their hypotheses are formed has evolved. At the same time, Roy et al. (Roy and McCallum, 2001) have pointed out that the earlier summarized uncertainty sampling, or estimates thereof through ensembles, are generally insufficient. They argue that they are prone to querying outliers, as a result of sampled instances being viewed in isolation and without regarding the underlying density of the full data distribution. Similar conclusions were empirically observed in the large scale empirical evaluation of active learning for text applications (Settles and Craven, 2008). As a solution, the authors suggest a representation based information density measure, and although heavy to compute, it implicitly takes into account the underlying data distribution. This can be seen as an approach that is orthogonal to minimizing the version

space, where now typically the distribution coverage on the entire dataset according to the model representations is maximized instead of reducing the number of possible hypotheses. The often necessary core assumption is thus the presence of the entire unlabelled pool of data and its auxiliary use in optimization of the labelled set. We have attributed a third category of active learning to approaches that follow this objective.

Active learning using pre-clustering (Nguyen and Smeulders, 2004) uses a k-medoids algorithm in conjunction with a SVM or logistic regression to select data from the pre-clustered embedding of the unlabelled pool. Similarly, SVM based core vector machines (Tsang et al., 2005) use a set of minimum enclosing balls to create a core set that best approximates the entire distribution. Li et al. estimate information density by using the unlabelled data in a Gaussian process (Li and Guo, 2013). The idea in these works have since been abstracted to deep neural networks. Sener and Savarese (2018) base their active learning procedure on construction of core sets based on a k-medians algorithm. Shui et al. (2020) achieve distribution coverage by matching distributions through minimization of the Wasserstein distance in Autoencoders (WAAL). Variational adversarial active learning (VAAL) (Sinha et al., 2019) approximates the data distribution by learning the latent space in a variational autoencoder (Kingma and Ba, 2015) and simultaneously trains a latent based adversarial network to discriminate between unlabelled and labelled data.

In complement to these works, various query-synthesizing methods have been proposed (Zhu and Bento, 2017; Mahapatra et al., 2018; Mayer and Timofte, 2020). Here, the challenge of active learning is tackled by using a deep generative model to generate informative queries. Instead of querying from an unlabelled pool directly, generative adversarial active learning (GAAL) (Zhu and Bento, 2017) and "efficient active learning using conditional generative adversarial network" (Efficient cGAN AL) (Mahapatra et al., 2018) both train GANs to synthesize and label queries. The core assumption is the ability to adequately capture the data distribution to generate meaningful instances. The usefulness of the generated samples with respect to a classifier can then either be assessed through uncertainty heuristics or by matching the synthesized data with samples from the pool and retrieving the most similar instance. The latter has been demonstrated in Adversarial Sampling for Active Learning (ASAL) (Mayer and Timofte, 2020).

In our later discussion, we will argue that the assumption of upfront presence of all data should, and in fact can be lifted when a natural bridge to the other paradigms is constructed. We proceed to conclude our review by delving into what will constitute the glue: learning in an open world and open set recognition.

3.3 Open set recognition

The term open set recognition was formally coined only recently (Scheirer et al., 2013; Bendale and Boult, 2015). However, its foundation and associated challenge in neural networks dates back to at least several decades before, when discriminative neural networks were found to yield overconfident mispredictions on unseen unknown data (Matan et al., 1990). To get an intuitive understanding, let us briefly consider the types of data we can expect our model to encounter. As soon as we move beyond the closed world benchmark scenario, we can no longer expect our trained models to be tested exclusively on some held-out data from the same distribution as observed during training. In the earlier introduced transfer learning

parlance, for prediction, data can thus generally not be presumed to originate from the same domain. We can now distinguish three types of possible inputs to our model (Scheirer et al., 2013):

1. **Knowns**: examples belonging to the distribution from which the training set was drawn. The model's prediction is accurate and confident.

2. **Known unknowns**: unknown instances that a model cannot predict confidently. Examples can optionally be labelled as not being affiliated with the set of known concepts for explicit training of negatives. Prediction uncertainty can indicate a model's awareness of its limitation.

3. **Unknown unknowns**: unseen instances belonging to unexplored, unknown data distributions or classes for which the prediction is generally overconfident and false.

The broader inspiration for this categorization is commonly attributed (Naylor, 2010; Scheirer et al., 2013) to a notorious, machine learning unrelated, quote by Rumsfeld (2002): *"We know that there are known knowns; these are things we think we know. We also know there are known unknowns; that is to say we know there are some things that we do not know. But there are also unknown unknowns; these are the ones we don't know, we don't know!"*. In the context of neural networks, known unknowns can be identified through gauging model uncertainty or relying on derived related heuristics, in correspondence to many of the methods employed in the active learning setting. However, as detailed in a recent survey (Boult et al., 2019), separating the known data from the essentially indistinguishable high-confidence mispredictions for unknown unknowns is far from trivial.

As any machine learning model is trained on a finite dataset, and the imaginable set of unknown unknowns is infinite, we refer to the challenge of recognizing the latter as open set recognition in analogy to prior works (Scheirer et al., 2013, 2014; Bendale and Boult, 2015, 2016; Boult et al., 2019). Formally, these works define the closed space as a union of balls S_K that enclose the entire training set $\boldsymbol{X}_K$, whereas the open space $\mathcal{O}$ constitutes the remainder of the input or feature space: $\mathcal{O} \subset= \mathcal{X} - \mathcal{S}_K$. Correspondingly, works that provide attempts at addressing open set recognition aim to find the respective boundaries between known and unknown spaces (Scheirer et al., 2013, 2014; Bendale and Boult, 2015; Lee et al., 2018b; Mundt et al., 2019a,b; Yoshihashi et al., 2019). We will review these works last in favor of historically preceding approaches based on explicit inclusion of negative classes and rejection through anomalies in prediction patterns, even though the latter have been argued to be insufficient for open set recognition (Matan et al., 1990; Scheirer et al., 2013; Boult et al., 2019).

The above widespread categorization can technically be extended to encompass a fourth category, by splitting the knowns into *known knowns* and the set of *unknown knowns* (Munro, 2020). We do not consider this further distinction as the existence of unknown knowns can be condensed to either a wilfully ignorant false prediction, because we in fact know the concept but choose to nevertheless treat it as unknown, or the more charitable alternative in which our chosen machine learning model has an inherent inability to represent the investigated concept and its structure altogether. We also note that there is other related concepts, such as novelty detection (Bishop, 1994) or equipping classifiers with rejection options. These are different in such that they are typically still evaluated in the close world

and data is generally still expected to reside in a similar domain. The aim is to recognise outliers of the distribution that are uninformative or represent a particularly interesting rare event. Although these works can have considerable merit in their respective closed world application context, we do not review them in favor of the more generic open set recognition, where considered inputs are allowed to be of almost arbitrary nature. We further note that we naturally cannot provide every example that has ever attempted open set recognition through simple heuristics like using the output values to distinguish examples.

3.3.1 Prior Knowledge

A conceivably simple effort to address unknown unknowns is by assuming that the human modeller has enough awareness about what forms of unknown inputs to expect during deployment to directly incorporate this prior knowledge into the model. As inclusion of prior knowledge into neural networks and other types of deep models turns out to be remarkably complex, the natural analogue is to steer efforts towards dataset design. "Inference with the universum" (Weston et al., 2006) has accordingly proposed to embrace prior knowledge by representing it through a collection of "non-examples", and hence letting the optimization algorithm decide how to include the presented information into the model. Unfortunately, this does not provide a general solution for open set recognition as upfront knowledge can only ever truly cover the family of known unknowns. At best, a mere work-around for major failure cases is therefore supplied, although without any associated guarantees for remaining unknown unknowns. This lack of guarantees is further enforced by the necessity to rely on machine learning algorithms extracting the information and composing abstractions from the supplied "non-example" data population.

Since then, the idea to include a "background" concept has been adopted so widely across applications, that singling out and thus giving preference to select works is difficult. Take as an example large-scale datasets surrounding the task of material classification and semantic segmentation. Because there is an abundance of material types, it has become the de-facto standard to collapse any available imagery that is connected to less important materials or where meager amounts of data are available into a single "other" material (Cimpoi et al., 2015; Bell et al., 2015). Not only is it impractical to gather data for every material variation, but also unknown unknowns can feature other significant statistical deviations, due to e.g. previously unencountered illumination, acquisition and sensor differences, superposition of dirt and surface markings, or any type of perturbation and previously unencountered noise. Imaginably, in real applications beyond a closed world, inclusion of an endless universe is by definition infeasible. Nevertheless, multiple recent works follow this route and propose mechanism to calibrate output confidences in deep models (Lee et al., 2018a), formulate a discrepancy loss between knowns and known unknowns (Yu and Aizawa, 2019), or modify the embedding to explicitly separate them, e.g. in semantic categorical and contrastive mapping (SCM) (Feng et al., 2019) or the Objectosphere loss (Dhamija et al., 2018). Although these approaches are not tantamount to a comprehensive solution, we note that they can still in principle be sufficient for tasks in partially constrained environments that naturally limit the world's openness.

3.3.2 Predictive Anomalies

From an unsuspecting angle, a model will consistently yield accurate predictions only for observed data and produce highly uncertain output otherwise, yet still generalize correctly to data that is from the same domain but has not been included in training. In this view, determining a prediction threshold and obtaining an uncertainty estimate is sufficient to recognize any form of unknowns. This can work surprisingly well in models with thorough understanding of the decision boundary and its neighbourhood, such as the Transduction Confidence Machine-k Nearest Neighbors (TCM-kNN) (Li and Wechsler, 2005). Even though it is well known that the entangled dense representations of neural networks result in overconfident predictions on any data (Matan et al., 1990; Boult et al., 2019), a variety of practical approaches nevertheless proposed to simply rely on a hinge loss to reject during classification (Bartlett and Wegkamp, 2008) or even to take the straightforward route and directly trust the softmax confidence (Hendrycks and Gimpel, 2017). As the quantitative outcome leaves room for improvement, multiple works have argued that uncertainty estimation is required to corroborate the decision to gain awareness of the unknown. In deep networks this could be achieved by assessing the variations of stochastic forward passes through a neural network with dropout (Srivastava et al., 2014; Kendall et al., 2017; Miller et al., 2018), as a variational Bayesian approximation to a distribution on the weights (Gal and Ghahramani, 2015), or by empirically estimating the output's variability with respect to introduced perturbations, such as done in ODIN (outlier detection in neural networks) (Liang et al., 2018), and by calibrating the prediction accordingly (Lee et al., 2018a). In similar spirit, an often employed argument is that generative modelling is required to obtain meaningful prediction values that allow to recognize out of distribution samples. For this purpose, Lis et al. (2019) use image resynthesis and equate detection of unknown concepts with identification of discrepancies in poorly reconstructed image regions. Likewise, one-class novelty GAN (OCGAN) (Perera et al., 2019) generates examples from sparsely populated latent space regions in order to use them in explicit training of a binary out-of-distribution classifier. Although predictions and uncertainty from generative models have been shown to improve outlier and adversarial attack detection in contrast to purely discriminative models (Mundt et al., 2019a,b; Li et al., 2019b), there is strong empirical evidence that this is still insufficient to provide a generic solution (Nalisnick et al., 2019; Ovadia et al., 2019; Mundt et al., 2019a,b). It is clear that former reported cases of success can be attributed to the specific constrained empirical studies and we illustrate some remarkably simple failure cases of prediction confidence and entropy in figure 5, even when uncertainty is assessed with Monte Carlo Dropout. This is to provide an intuitive picture of the challenge of open set recognition with neural networks and to summarize and repeat the findings of the much more detailed experiments presented in numerous prior works (Mundt et al., 2019a,b; Nalisnick et al., 2019; Ovadia et al., 2019).

3.3.3 Meta-recognition

Rather than assuming that predictions are somehow calibrated for any data, a more rigorous approach is to prevent overconfident misclassification by confining the model to the known closed space and averting any prediction from little-known open areas in the first place. Whereas it is evident how to achieve this when explicitly modelling the distribution, such as

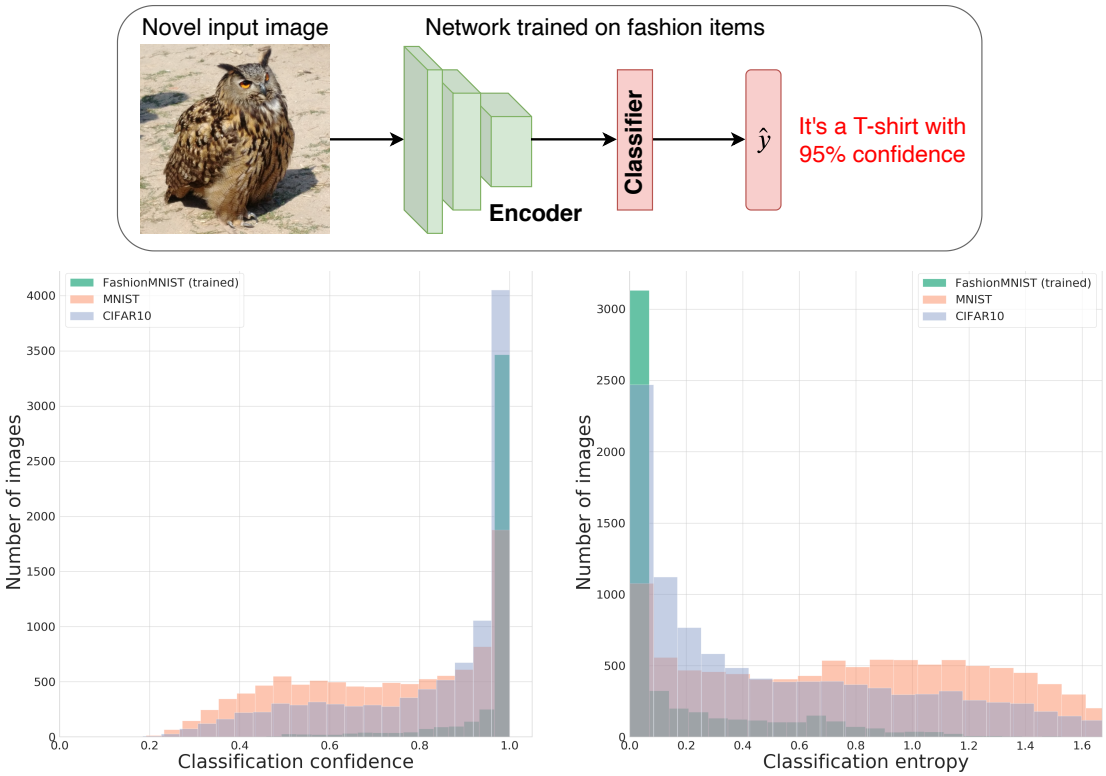


Figure 5: Top panel: Qualitative illustration of the challenge of open set recognition. A neural network that has been trained to discriminate fashion items misclassifies the unknown concept of an owl and assigns it to the t-shirt class with very high confidence. Bottom panel: A quantitative example of a deep wide residual neural network trained on the FashionMNIST dataset, asked to classify unrelated unencountered digits and objects from the MNIST and CIFAR10 datasets. Even though uncertainty is estimated using 50 Monte Carlo Dropout passes, misclassified unseen data still overlaps significantly with the known dataset in prediction confidence or entropy. Knowns and unknowns are largely indistinguishable. The shown quantitative results are a reproduced subset of our previous work investigating the limits of deep neural network uncertainty for open set recognition (Mundt et al., 2019b).

done in probabilistic mixture models, a straightforward approach is not typically applicable in the often complex feature hierarchies of modern discriminative machine learning approaches. A common technique is thus to resort to meta-recognition on top of the empirically emerged features obtained through black-box optimization procedures. Scheirer et al. (2014) give an intuitive example based on support vector machines. Here, the menace of erratic predictions for unknown unknowns results from examples being projected close to the linear decision boundary, while at the same time being mapped arbitrarily far away from the training data along a different dimension. The authors therefore define a compact abating probability (CAP) model, where the key idea is to make use of insights from extreme value theory (EVT). The essential notion is to take into account inherently present extreme statistical differences in the long tail of an extreme value distribution, here the Weibull distribution,

and subsequently monotonously decrease a data point's probability of belonging to the observed closed set with increasing distance from the observed data population. In other words, a prediction is discarded in sparsely populated areas, independently of a sample's proximity to the decision boundary. Bendale and Boult (2016) have extended this approach to discriminative deep neural networks, where the above meta-recognition idea is transferred to the network's penultimate layer. They propose the OpenMax algorithm that lowers softmax prediction probabilities with increasing distance from the average penultimate layer's activation values. A strongly related approach has been proposed in Lee et al. (2018b), where the affinity of a data point to the known set is measured based on a Mahalanobis distance in the feature space of the penultimate layer. More recent works have come to the conclusion that although the latter approaches have a strong theoretical foundation for open set recognition, they are still limited by activation values in discriminative neural networks being optimized exclusively towards predicting a correct class (Yoshihashi et al., 2019; Mundt et al., 2019a,b). In particular, the penultimate layer activation values do not generally encode all the information about the data $\boldsymbol{x}$ that might be required for open set recognition. "Classification Reconstruction learning for Open-Set Recognition" (CROSR) (Yoshihashi et al., 2019) has thus suggested to additionally append a generative model's latent variable $\boldsymbol{z}$ to the OpenMax classification procedure. Concurrently, open set classifiying denoising variational autoencoders (OCDVAE) (Mundt et al., 2019a,b) translate the EVT based meta-recognition to a variational Bayesian setting. Here, the open set recognition is based directly on the approximate posterior in a deep generative model, which enables a natural interpretation based directly on the underlying generative factors of the data distribution $p(\boldsymbol{x})$, instead of activation value heuristics. We believe that this approach offers one potential framework to consolidate research in active learning, open set recognition and continual learning. We will correspondingly revisit the underlying approach, detail specific methods and introduce extensions in the next section.

4. Bridging perspectives: past insights and the challenge of evaluation

In the previous sections, we have kept up the tradition to treat continual machine learning, active learning and open set recognition as three distinct challenges. For convenience we provide a visual summary of the taxonomy in diagram 6. Distinctly categorized approaches are rarely coupled and synergies exploited only in select works, such as the combined continual learning approaches. More importantly, the intersection between the three machine learning paradigms remains largely unexplored. Highlighting the necessity for unification of the latter into a single viewpoint is the primary purpose of this work. The remainder of the paper will now serve the purpose of revealing the natural interface. In fact, by identifying former lessons, stressing shortcomings of prevailing evaluation practices and bridging seemingly forgotten connections, we develop a wholistic view that simplifies the deluge of ongoing research questions into a single intuitive framework. To better understand why this is imperative for future progress, let us briefly recall the earlier mentioned predominant evaluation routines and link insights from prior works to their current limitations.

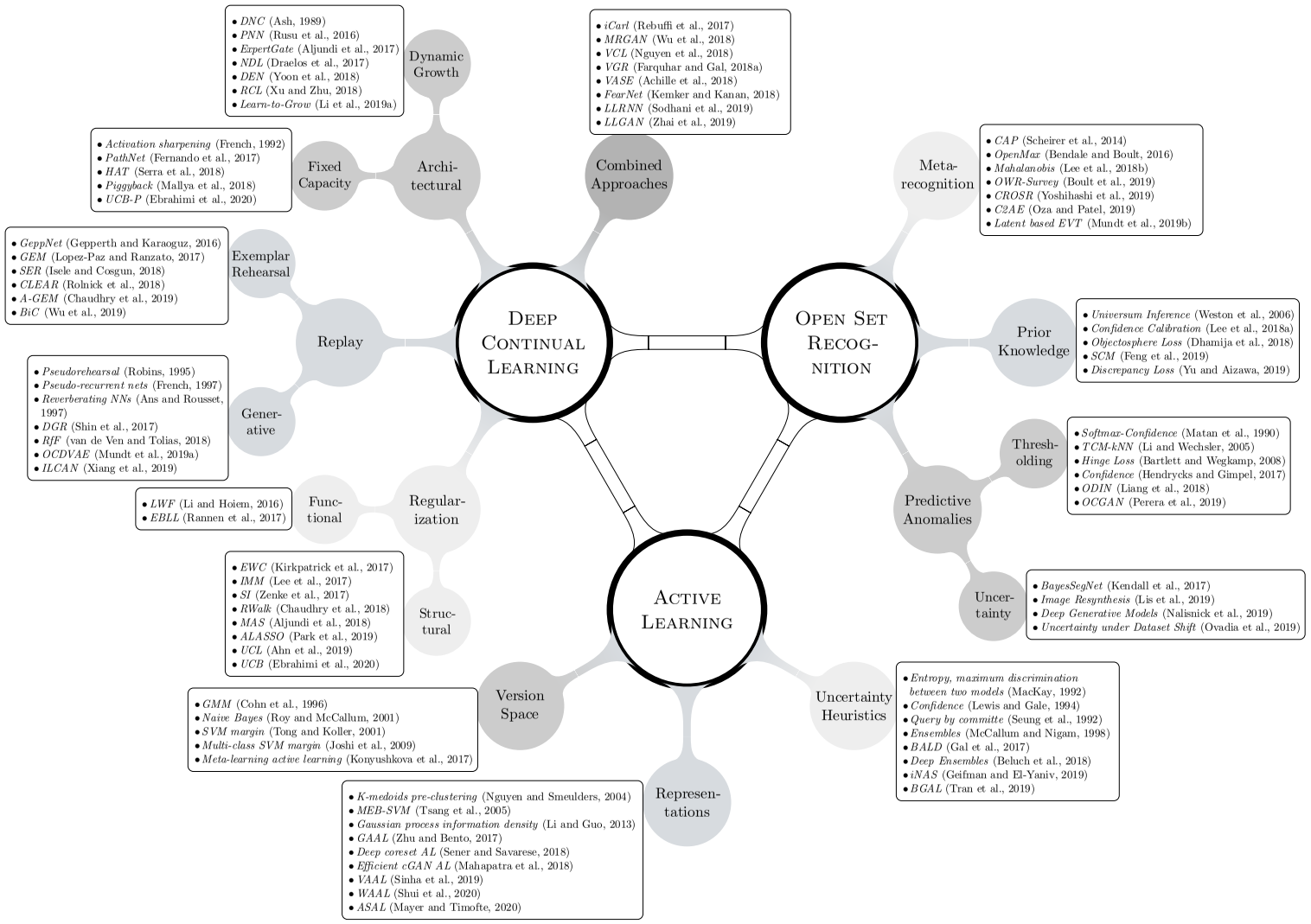


Figure 6: Visual taxonomy of neural network based methods for continual learning, active learning and open set recognition.

If we look back at figure 2 and the corresponding section's discussion, we recall that deep continual learning typically collapses its practical evaluation to measuring catastrophic forgetting between task increments. These task increments belong to simple sequentialized versions of existing benchmark datasets and a continual learning technique is deemed successful if the model that is trained over time approaches the expected performance when trained in isolation. In almost complete analogy, active learning evaluation revolves around accuracy gains between query steps. In the majority of the aforementioned related works, the focus is exclusively on whether a specific query mechanism surpasses another in terms of quickly approaching the overall error achieved on a complete dataset. For empirical benchmarking purposes, the model is simply trained in isolation on multiple selected subsets of known data, where the difference between these subsets corresponds to the inclusion of one active query.

Before we continue with the limitations of such evaluation protocols, we emphasize that our intention at no point in this paper is to discredit and devalue the bulk of previously proposed methods. However, we would argue that claimed advances of individual methods are in grave danger from their constrained benchmark evaluation being non-indicative of the actual machine learning progress on a larger scale. We believe a major contributing factor is that key insights from past, often neural network unrelated, literature have surprisingly gone unnoticed or have been written off in the era of deep learning. To attach a slightly provocative connotation, we have termed these overlooked insights forgotten lessons. Although the term "forgotten" certainly is an exaggeration with regard to the ML field as a whole, the absence of derived practical implications is strongly manifested in deep learning evaluation schemes.

4.1 Forgotten lessons from past literature

Forgotten lesson 1: *Machine learning models are by definition trained in a closed world, but real-world deployment is not similarly confined. Discriminative neural networks yield overconfident predictions on any sample.*

Independently of whether additional metrics such as training speed-ups through representation transfer, computational cost or memory consumption are taken into account, currently considered experimentation features closed world train and test sets. This is occasionally amplified by continual learning works assuming the presence of a task oracle for testing or respectively the assumption of an infallible oracle to yield flawless data when labelling active learning queries. As such, open issues concerning continual training of a model or active learning queries in an open world are generally neglected. However, real-world deployment almost always inhabits an open world. In the extreme case, the model has to handle data from completely unknown type in previously unfamiliar conditions, think outdoor environments or uncontrolled arbitrary user inputs in web-based applications. Instead of the common overconfident misprediction that falsely attributes this data to any known concept, a multiple decade old seemingly forgotten insight (Matan et al., 1990), any machine learning model should at least be equipped with the ability to identify unencountered scenarios and warn the practitioner. As a much milder, but heavily realistic form of an open world, even commonly occurring corruptions are disregarded, think blur or camera noise in images. The menace of the latter has recently been demonstrated in deep learning by Hendrycks and Dietterich (2019), where the authors

empirically demonstrate that current deep neural networks not only exhibit severe instability with respect to various simple perturbations, but advances in neural network architectures are reflected in only diminutive changes in robustness. Whereas certainly this hazard is universal to all machine learning research that is deployed in practice, continual and active learning are particularly prone to the threat of corrupted and unknown data as their goal is to accumulate knowledge from previously unseen sources already in the training process itself.

Forgotten lesson 2: *Uncertainty is not predictive of the open set. Active learning resides in an open world and common heuristics based query mechanism are susceptible to meaningless or uninformative outliers.*

Although early works have rapidly identified the fallacy that uncertainty sampling is a meaningful strategy to query (Roy and McCallum, 2001; Settles and Craven, 2008) in active learning or respectively detect unknown unknowns (Matan et al., 1990; Atlas et al., 1990), the belief that uncertainty provides a generic solution seems to have resurged with the advances of deep learning. This is apparent from the many approaches in our previous literature review basing querying strategies or detection of unseen examples on heuristics that rely on output variability or similar entropic quantities, see the branches labelled with uncertainty and predictive anomalies in our literature review diagram 6. Indeed, the challenge of accurate uncertainty quantification in deep learning is already genuinely difficult and does provide advantages in contrast to less principled empirical thresholding. However, paying homage to the detailed argumentation of the recent review by Boult et al. (2019), any machine learning model is still trained in a closed world scenario, independently of whether e.g. a Bayesian formalism is employed to obtain uncertainties. Predictions for y are known to be overconfident, uncertainty is not calibrated for points outside of $p_{train}(\mathbf{x})$ and the posterior is often unusable, regardless of how well it is approximated.
In other words, given any parameters ϕ and an unknown unseen input example $\mathbf{x}^*$, we don't know if evaluating $q_\phi(\mathbf{z}|\mathbf{x}^*)$ will produce something meaningful. This issue is by no means exclusive to detecting unknown unknown examples, but comes with the same implications for realistic active learning scenarios. Take for example a more realistic set-up beyond a crafted benchmark where data is scarce and the investigated domain is demanding even for experts. The earlier reviewed VAAL has considered such a scenario with medical imaging, where correct oracle labelling and a noiseless image cannot always be expected. Sample selection based on uncertainty does not protect the query from such noise and there is a large chance that meaningless outliers are included into the system.

Forgotten lesson 3: *Confidence or uncertainty calibration, as well as explicit optimization of negative examples can never be sufficient to recognize the limitless amount of unknown unknowns.*

At a first look, one might believe that impressive successes where demonstrated with approaches that extend the basic idea of "inference with the universum" (Weston et al., 2006). Explicitly using prior knowledge in terms of expectations on what form of inputs can be anticipated, or respective inclusion of negative data that is believed to play a role in deployment, are popularly exhibited by works that have identified and attempt to address the first two lessons. The common presumption across all these works is the upfront presence of a larger, possibly unlabelled, dataset that can explicitly be included into the optimization

process. Just as supposed out-of-distribution examples are made use of to modify loss functions and calibrate the output for detection of unknown unknowns (Bell et al., 2015; Lee et al., 2018b; Yu and Aizawa, 2019; Dhamija et al., 2018; Feng et al., 2019), active learning techniques often resort to conditioning their procedure on the entire data pool (Nguyen and Smeulders, 2004; Sener and Savarese, 2018; Li and Guo, 2013; Shui et al., 2020; Sinha et al., 2019), e.g. through clustering (Nguyen and Smeulders, 2004; Sener and Savarese, 2018) or fitting a generative model to the unseen data (Li and Guo, 2013; Shui et al., 2020). Unfortunately, this impedes evaluation beyond a constrained closed set benchmark and more realistic continual and active learning scenarios where data becomes available at different times cannot be considered. In a sense the problem seems to be addressed from a reverse perspective. Instead of acquiring explicit knowledge about the nature of the trained data distribution, the challenge is sidestepped by reformulating it as an optimization problem that attempts to find the boundary between known and an existing set of unseen data, which by definition then does not consist of unknown unknowns. Thus, we receive no guarantees, as the pool of unlabelled data at any point in time is limited and can never truly approximate the unknown space.

Apart from this obvious argument that it is impossible to include all forms of variations and exceptions upfront, else we could have just modelled and hand-crafted the entire system from the start instead of falling back on purely data driven approaches, previous works have also asserted that the particular form of representations of discriminative deep neural networks can further confound predictions. The early work of French (1992) has already pointed out that a major complication of continually training neural networks is their distributed representations and has subsequently investigated mechanism to obtain semi-distributed representations with sharp activations that are concept specific. We argue that with the onset of deep learning the challenge of distributed representations is further magnified due to distribution across the layer hierarchy. First, consider as an example a neural network that is trained to discriminate cars from aeroplanes, a scenario often assumed when incrementally training the popular CIFAR10 dataset (Krizhevsky, 2009). As the neural network is not explicitly encouraged to encode information about the data distribution, the obstacle of predicting overconfidently on unseen data is further magnified by the ubiquitous option for any classifier to differentiate a concept based on a combination of noise patterns, the absence of a specific pattern, or background patterns altogether (Xiao et al., 2020). In the car versus aeroplane scenario, depending on how well and diverse the dataset is constructed, this could be as trivial as distinguishing the two classes by identifying the presence of some feature that describes the sky. As neural networks have been demonstrated to rely heavily on texture rather than object boundaries (Geirhos et al., 2019), this is not far fetched. In fact, a prominent recent work on "Unmasking Clever Hans" predictors (Lapuschkin et al., 2019) has shown that the decision making of a discriminative deep neural network can be based on entirely trivial features, such as a certain object always occurring at a specific location in every image or almost imperceivable photography tags. "Adversarial examples are not bugs they are features" (Ilyas et al., 2019) takes this one step further and empirically showcases how classes can be distinguishable solely based on noise patterns. In a trivial case of our above car versus aeroplane example, presenting the trained model with images of ships that feature the similarly blue background of the sea is then not surprisingly resulting in overconfident misclassification. Using ships as a background class could initially solve this

problem of attributing blue to aeroplanes. However, if a significant portion of our learned features were indeed to be composed of noise, background and adversarial patterns, then we would argue that overconfident mispredictions are impossible to overcome, as the extent of data on which these features activate is inconceivable to any human modeller. We believe this makes the approach to handle outlying and unknown unknown data through prior knowledge even less feasible.

Forgotten lesson 4: *Data and task ordering are essential. Although this forms the quintessence of active learning it is yet untended to in continual learning.*

It is well known that each dataset instance does not contribute equally to the overall objective. This forms the foundation and rationale behind active learning. In general, when conducting active learning queries, there is a trade-off between exploring the unknown space and exploiting more of the already known to avoid misclassification (Joshi et al., 2009). Alas, the implications of the latter statement are more nuanced and go beyond the simple question of whether a certain subset spans the entire data distribution. As an example, Joshi et al. (2009) found certain active learning strategies to benefit primarily from creating a class imbalance, as more difficult classes might require a denser sampling than others. Bengio et al. (2009) have similarly found that sorting data in a curriculum that introduces classes into the training process according to their difficulty improves the obtained accuracy. Recently, Hacohen et al. (2020) have empirically observed that deep neural networks seem to build such a curriculum inherently during the training process. Consistently across multiple architectures, they always learn the same examples first when given access to the entire dataset, even though the mini-batch stochastic gradient descent shuffles the data differently every time. Intuitively, this notion of learning according to some measure of complexity seems only natural, as describing some inputs necessitates less complex and nuanced patterns than others.

Even though there is significant empirical evidence that data selection and task order plays a vital role for any learned algorithm, modern deep continual learning, to the authors' astonishment, seem to pay little attention to a careful experimental design.

Out of the numerous works of the previous review, less than a handful of works consider the question of task order at all. The rest remains in the comfort of benchmark datasets, where the classes are split and introduced in sequence for continual learning according to a class id that often just reflects an alphabetic ordering. However, there is no rigorous investigation of the effect of task order. Two out of the four works that examine task order (Serra et al., 2018; Isele and Cosgun, 2018) only randomize the order across multiple experimental repetitions to obtain an average performance estimate. The other two (De Lange et al., 2019; Javed and Shafait, 2018) follow this practice, but go even further and make the statement that task ordering has minimal influence towards continual learning methods. We will later demonstrate that this is obviously not the case, and can simply be attributed to the experimentation being a narrow trial of five randomly obtained orderings without any attached semantics. When selecting tasks from the overall pool of available data according to their similarity or dissimilarity with the already observed data distribution, we will observe a major divergence of obtained results.

Whether or not having access to all future tasks in order to select an ideal order is unrealistic in real-world continual learning scenarios, we believe task ordering to be an

imperative factor that should be considered when designing our benchmarks to further our understanding. In particular, we note that a very common practice to reduce the computational cost of incrementally learning large scale datasets such as ImageNet (Russakovsky et al., 2015) is to extract subsets (Rebuffi et al., 2017; Wu et al., 2019; De Lange et al., 2019; Park et al., 2019). The main problematic here is that selecting e.g. 50 or a 100 from a larger pool of 1000 classes heavily influences the achievable result and using random selection mechanisms essentially renders works unreproducible.

Forgotten lesson 5: *Parameter and architecture growth are not distinct methods to address any particular challenge such as catastrophic forgetting. They are at the core of the learning process.*

We do not truly believe that the above lessons is forgotten, however, feel the need to call attention to it because an entire branch of continual learning seems to treat parameter addition and architecture growth as a separate solution. Our main goal for techniques that modify architectures on the fly is to point out that these should be analysed with particular caution. On the one hand, methods that use neural networks that are highly over-parametrized can implicitly expand their effective representational capacity due to the abundance of parameters when encountering new data. Investigated algorithms could thus always implicitly be accompanied with some form of representational expansion, depending purely on the initial choice of architecture. On the other hand, in active learning it has been shown that training in small sample scenarios is not only computationally more efficient with smaller neural networks but also yields more accurate estimates in these early stages if less representational capacity is available (Geifman and El-Yaniv, 2019). Whereas the latter statement might seem obvious to some reader, we note that this behaviour makes it tremendously difficult to attribute gains of active learning or continual learning experiments to a specific technique in contrast to innate advantages of the used architecture at any point in time.

4.2 Open set recognition: the natural interface between continual and active learning

As indicated in the previous sections, contemporary continual and active learning are prone to an alarming amount of threats due to their development and evaluation inhabiting a closed world. In this section we argue that awareness of an open world is not only required to overcome the threat of designing a non-robust system, but provide the natural means to merge techniques into a common perspective.

Recall that a majority of continual learning techniques alleviates the challenge of catastrophic inference by regularizing parameters for known tasks, rehearsing a subset of data from known tasks or respectively generating it with a generative model. Independent of the specific algorithm, a key concern is thus to identify exemplars, learn the generative factors of our known tasks or determine the parameters that are responsible for the majority of previously seen data. At the core, we need to thus find a good approximation of the known data distribution. In active learning, our task is very much alike, although the underlying question seems to be of reversed nature. Instead of protecting or sampling from the known data distribution, a query is conducted with respect to yet unobserved distributions. In a

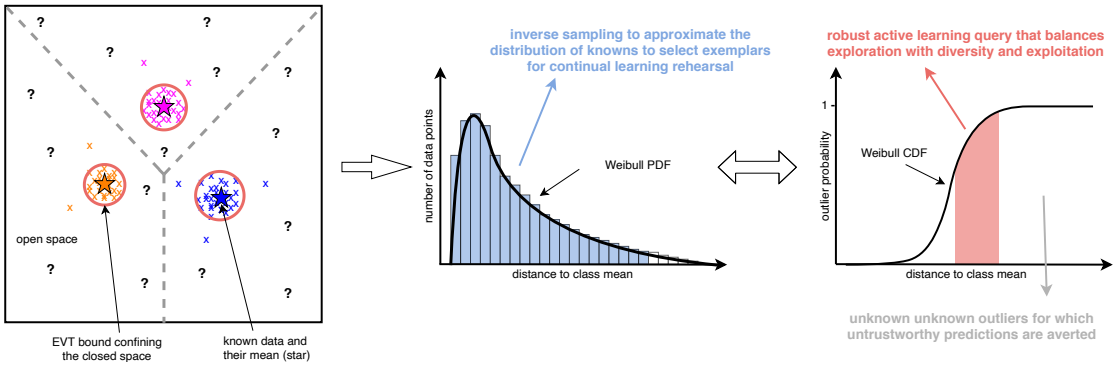


Figure 7: Conceptual diagram to illustrate how extreme value theory based meta-recognition in neural networks can serve as a common denominator to protect knowledge in continual learning, conduct principled queries in active data selection, while having the capability to reject or set aside unknown unknown data at any point in time. The leftmost figure of an embedding showcases the threat of the open space, where any examples that are very far away from known clusters always get falsely assigned to a known class and can be arbitrarily close to the decision boundary. The mid panel shows how a Weibull distribution, which models the extreme distance values to the mean of the correctly predicted trained data in a heavy tail, can enclose the known space (suggested by the red circles in the embedding). The corresponding cumulative distribution function in the right panel can be used to reject or set aside outliers and balance active learning queries to sample diverse, yet meaningful data (shaded red area). Alternatively either curves can be sampled inversely to select a subset of inlying data to approximate the entire known distribution in continual learning rehearsal (shaded blue area).

similar distinction to the continual learning mechanisms, query-acquiring active learning methods pick samples that are estimated to yield the best model improvement, whereas query-synthesizing methods attempt to tackle this challenge through generative modelling by generating these most informative examples.

Interestingly, in open set recognition, the task is to precisely gauge the boundary between the seen known data distribution and yet unseen unknown data. Although the original motivation stems from a perspective of outlier detection and thus model robustness in practical application in the presence of unknown unknowns, knowing this boundary also gives us the means to restrict a continual learning technique to protect the already seen knowns or respectively query active learning examples that are sufficiently statistically different without the fear of selecting uninformative noise. We argue that in general this forms the natural interface between active and continual learning.

We follow previously reviewed works that employ EVT based meta-recognition to identify unknown unknowns and schematically illustrate our proposed unified framework in figure 7. We will delve into the mathematical details of its realization in deep neural networks in the next section. For now, consider a generic embedding as a result of some deep neural network encoding. In the figure's leftmost panel, we have visualized an example embedding for three

classes, with their mean indicated by a star and a potential decision boundary by dashed lines. In order to confine predictions to the known space, EVT based meta-recognition makes use of data instances with extreme distance values to the average embedding of a class. Typically, a Weibull distribution is used to model the distance distribution for the entire dataset and capture samples that feature stronger deviation in a heavy tail. In the original works that have proposed this model for open set recognition (Scheirer et al., 2013, 2014; Bendale and Boult, 2015), the cumulative distribution function is then used to estimate whether a new unseen example should be regarded as an unknown unknown, outlying data point. In our own previous work (Mundt et al., 2019a), we have identified this technique to also be fundamental in judging whether a randomly sampled latent vector is proximate enough to the observed data such that it results in a clear output of a generated model.

We now close the circle and tie this method to retention of a core set for continual learning, as well as a query mechanism for active learning, while retaining the method's innate ability to reject and set aside unknown unknowns. First, we postulate that the Weibull distribution for each data point's distance to the mean embedding equips us with a tool to approximate the known distribution with a subset. Specifically, we can employ inverse sampling from the Weibull probability density function to create a set of distance values with an arbitrary prior on how much of the distribution's tail should be disregarded, i.e. how many outliers are already assumed to be inherently present in the original dataset. Practically, we can then approximate the data distribution with a subset by selecting data instances whose embedded value lies closest to the drawn sample. Alternatively, as indicated in the diagram, we could discretize the distribution and sample a certain number of examples from each bin. Conversely, for active learning, we are less interested in sampling from the known distribution, but much more in the heavy tail. To our advantage, the long tail models data that is statistically deviating, but can still be attributed to the distribution of interest. We can thus balance exploitation with exploration. First and foremost, data instances for which the outlier probability is unity are avoided altogether in order to prevent sampling of uninformative noise or other corrupted data. Recall, that this is the primary pitfall of uncertainty sampling. At the same time, we want to avoid samples that have a minute probability of being an outlier, as these samples are too similar to previously observed data and are therefore also uninformative due to redundancy. As such, we can constrain our query to the center area of the cumulative distribution function (CDF), illustrated by the shaded area under the CDF in the diagram. The rationale for this approach can intuitively be understood by looking back at the theoretically grounded works of version space maximization. We can implicitly reduce this space of possible hypothesis, even in complex models such as neural networks, as we incrementally expand the radius of the ball that encloses the closed space by sampling carefully along its boundary with each active learning query. This way, we avoid the vast open space and the redundant highly dense areas of known data, while making sure that previously unseen information is acquired.

Before we proceed with one imaginable realization of this unified framework in neural network and its mathematical formalism, we note that there is two works that have previously initiated a bridge between active learning and open set recognition, alas have not fully built it yet. The recently introduced open world learning (Bendale and Boult, 2016) and the concurrently named cumulative learning (Fei et al., 2016) advance the pure open set

identification step by proposing to set aside the unknown unknowns and including them into a later active learning cycle. Whereas these works made first steps towards formulating learning in an open world, they however assume the presence of labels for the entire dataset and the addition of classes itself is in the form of a fixed sequence that is injected by the human. The system is limited as it does not self-select which classes or instances should be learned next, nor does it protect its knowledge for continual learning, where the assumption of availability of all data at all times is lifted. As a result, the empirical evaluation is simply an investigation of the performance on the entire test set at each state of the growing known training set. Finally, the suggested open world learning (Bendale and Boult, 2016) is based on nearest mean classifiers based on simple SIFT features and is yet to be extended to the context of modern deep neural networks.

5. Uniting perspectives with deep generative neural networks

How can we realize our proposed unified framework in a meaningful way in deep neural networks? As emphasized by prior work (Yoshihashi et al., 2019; Mundt et al., 2019b), identification and correlation of unseen data with average activation patterns of known data is not necessarily sufficient in discriminative models, even when extreme values are modelled to obtain closed space boundaries, see prior works (Mundt et al., 2019a,b) for empirical verification. This is because a neural network based classifier is generally not encouraged to aggregate the whole information describing the data, merely the features that allow for class distinction. These features themselves, come with a variety of further pitfalls, as summarized in the forgotten lessons. In our own previous work (Mundt et al., 2019a,b), we have overcome this limitation by formulating the problem from a perspective of deep generative models trained with variational Bayesian inference, i.e. variational autoencoders (VAE) (Kingma and Welling, 2013). We will lean on this viewpoint, follow the notation of prior works and extend it towards one potential solution to consolidate continual and active learning through open set recognition.

The rationale to build upon VAEs is rather straightforward: the Bayesian formulation lets us learn about the distribution of seen data $p(\mathbf{x})$ by capturing it through latent variables $\mathbf{z}$. However, as $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) dz$ is untractable, we do this by optimizing a lower-bound to the marginal distribution $p(\mathbf{x})$, since the densities of the marginal and joint distribution are related through Bayes rule $p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x},\mathbf{z})}{p(\mathbf{x})}$. As we do not know our real posterior $p(\mathbf{z}|\mathbf{x})$, we typically resort to variational inference and introduce a variational approximation $q(\mathbf{z}|\mathbf{x})$ to the posterior. In a neural network, this approximation $q(\mathbf{z}|\mathbf{x})$ is learned through the parameters of a probabilistic encoder, whereas a probabilistic decoder is trained for the joint distribution $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ and thus forms the generative component. This generative model can effortlessly be augmented to additionally discriminate classes by including their label into the latent variable, e.g. by enforcing a linear class separation on $\boldsymbol{z}$. The corresponding factorization and generative process is then simply $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})$ (Mundt et al., 2019a,b). Such formulation of a classifying variational autoencoder comes with the main advantage that using latent variables $\mathbf{z}$ allows us to base our decision regarding unknown unknowns on the underlying generative factors of variation and whether an example is close to the high density regions of our approximated data distribution.

5.1 The boundary between known and unknown

The first step towards open world aware active and continual learning is to train the above mentioned classifying variational autoencoder, followed by determining the boundary between the open and closed spaces for the observed distribution with the help of EVT. For ease of readability, we repeat the training and fitting procedure described in our previous work (Mundt et al., 2019a,b). The model's probabilistic encoder and decoder are trained jointly by minimizing the divergence between the variational approximation $q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and a chosen prior $p(\boldsymbol{z})$, typically $\mathcal{N} \sim (0, I)$, and the conjunction of reconstruction loss and the linear classification objective, parametrized through $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ respectively. For a dataset consisting of $n = 1, \ldots, N$ elements, the following lower bound to the joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$ is thus optimized:

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\xi}\right) = &- \beta KL(q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)}) \,||\, p(\boldsymbol{z})) \\ &+ \mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)})}\left[\log p_{\boldsymbol{\phi}}(\boldsymbol{x}^{(n)}|\boldsymbol{z}) + \log p_{\boldsymbol{\xi}}(y^{(n)}|\boldsymbol{z})\right]\end{aligned} \tag{1}$$

At any point in time of training this model, there is a natural discrepancy between the prior and the approximate posterior. The added β factor in above equation serves the purpose of controlling this gap. Whereas one could believe this distributional mismatch to be an undesired property, we recall the arguments conjectured in multiple previous works (Hoffman and Johnson, 2016; Burgess et al., 2017; Mathieu et al., 2019). In essence, they state that the overlap of the encoding needs to be reduced in order to avoid indistinguishability, but at the same time prevent latent variables to consist of individual uncorrelated data points that resemble a pure look-up table. In the intuitive picture of diagram 7, think of the former as multiple classes collapsing and thus being inseparable, and the latter as the dense clusters being scattered to allow differentiation of each and every single data point without a strong encoding of correlations. Therefore, the actually captured encoding of the data distribution should not simply be assumed to correspond to the prior, but rather corresponds to an empirically determinable distribution referred to as the aggregate posterior:

$$q_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathbb{E}_{p(\boldsymbol{x})}\left[q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right] \approx \frac{1}{N}\sum_{n=1}^{N} q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}^{(n)}) \tag{2}$$

Using EVT to find the boundaries of this distribution now corresponds to identification of our model's closed space. For emphasis, we repeat that this is necessary because VAEs generally assign non-zero density to any point in the latent space, the analogue of overconfident classifier predictions (Nalisnick et al., 2019; Ovadia et al., 2019), and that this boundary is not analogous to the extent of the prior because low density areas exist inside the prior as well. Practically, an EVT based fit can be obtained by empirically accumulating the mean latent variable for each class c for all correctly predicted known data points $m = 1, \ldots, M$:

$$\bar{\boldsymbol{z}}_c = \frac{1}{|M_c|}\sum_{m \in M} \mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x}^{(m)})}\left[\boldsymbol{z}\right] \tag{3}$$

and defining a respective set of latent distances as:

$$\Delta_c \equiv \left\{ f_d\left(\bar{\boldsymbol{z}}_c, \mathbb{E}_{q_\theta(\boldsymbol{z}|\boldsymbol{x}_t^{(m)})}\left[\boldsymbol{z}\right]\right)\right\}_{m \in M_c} \tag{4}$$

Here, f_d represents a chosen distance function, which prior works have typically chosen to be either euclidean or cosine distance (Scheirer et al., 2013, 2014; Bendale and Boult, 2015; Mundt et al., 2019a). As this set represents the distances to the class conditional aggregate posterior, we can fit a Weibull distribution with parameters $\rho_c = (\tau_c, \kappa_c, \lambda_c)$ on Δ_c to model the trustworthy regions of high density that represent the observed data distribution, where the heavy-tail indicates a decaying reliability:

$$\omega_{\boldsymbol{\rho}}(\boldsymbol{z}) = \frac{\boldsymbol{\kappa}}{\boldsymbol{\lambda}} \left(\frac{|f_d\left(\bar{\boldsymbol{z}}, \boldsymbol{z}\right) - \boldsymbol{\tau}|}{\boldsymbol{\lambda}} \right)^{\boldsymbol{\kappa}-1} \exp\left(- \frac{|f_d\left(\bar{\boldsymbol{z}}, \boldsymbol{z}\right) - \boldsymbol{\tau}|}{\boldsymbol{\lambda}} \right)^{\boldsymbol{\kappa}} \tag{5}$$

Here, τ defines the location, λ the scale and κ the shape of the distribution. We can now make use of this distribution to pinpoint the observed data distribution, as a surrogate to the otherwise highly complex aggregate posterior. We proceed to highlight its various use cases in the following sections.

5.2 Approximate posterior based open set recognition

As described in previous works (Mundt et al., 2019a,b), the most direct use of the aggregate posterior based Weibull parameters $\boldsymbol{\rho}$ is the identification, rejection or storage of unknown data. Using the corresponding cumulative distribution function (CDF) to the probability density function of equation 5, we can now estimate any data instance's statistical outlier probability for every known class:

$$\Omega_{\rho_c}(\boldsymbol{z}) = 1 - \exp\left(- \frac{|f_d\left(\bar{\boldsymbol{z}}_c, \boldsymbol{z}\right) - \tau_c|}{\lambda_c} \right)^{\kappa_c} \tag{6}$$

When we have observed multiple classes, we will typically take the minimum $\min\left(\Omega_{\boldsymbol{\rho}}\right)$ of this equation across all known classes c and the respective mode's parameters ρ_c. This expresses the basic condition that a data point should be considered as a statistical anomaly only if its outlier probability is large for each known class. A respective decision should thus be based on the class where the smallest deviation to known data is observed. The more dissimilar a sample is with respect to the observed data distribution as approximated by the aggregate posterior, the more the outlier probability will approach unity. Irrespective of whether a machine learning algorithm is developed for active learning, continual learning or in fact any other paradigm, this robustness towards unknown unknown data is essential for any practically deployed system that operates outside of extremely narrow conditions.

5.3 Outlier and redundancy aware active queries

Equation 6 gives us the direct means to estimate a sample's similarity with the already known data. For active learning this almost directly translates to the informativeness of a query. Small CDF values signify large similarity or overlap with already existing representations, larger values indicate previously unobserved data. Naively, one would follow the earlier strategies developed in uncertainty based active learning and simply query batches that consist of the most outlying data points. However, this would neither grant protection from exploring noisy, perturbed and uninformative data, nor balance it with exploitation to fester partially known concepts. Our proposition is thus to query a variety of data that is well

distributed across the center part of the CDF, i.e. data that surpasses an outlier probability of e.g. 0.5 and at the same time is limited on the upper end by e.g. a value of 0.95. As explained in the earlier introduction of the framework, this is tantamount to sampling on the outer edge of the sphere that encloses the currently known closed space. Naturally, as a repetition of the ultimate statement of the last subsection, if the employed active learner is simultaneously deployed or used in application once it has finished learning, avoiding predictions for unknown unknown data is imperative.

5.4 Core set selection for continual learning rehearsal

In contrast to active queries that need to select meaningful unknown data, in the currently formulated continual learning paradigm the main goal is to protect the known knowledge while learning a predetermined new task. We will question the role of the order prearrangement in the next subsection. Here, we focus on open world aware techniques to preserve previously acquired representations. Depending on available memory, the most successful approaches either store and rehearse a small subset of exemplars or alternatively generate data for former tasks with a generative model. In our previous work (Mundt et al., 2019a) we have shown how we can use equation 6 to reject samples from the prior $\boldsymbol{z} \sim p(\boldsymbol{z})$ that do not fall into the obtained bounds of the aggregate posterior for generative rehearsal. The choice for this sampling with rejection originated from the decision to employ the cosine distance, which collapses the distance to a scalar. A different distance function, such as a euclidean distance per dimension would allow to directly inversely sample a highly multi-modal Weibull distribution, i.e. with one mode per dimension per class. Independently of the selected distance metric, we can leverage inverse sampling for the construction of a small data subset. Specifically, drawing at uniform from the inverse of the CDF in equation 6 is guaranteed to yield samples that approximate the aggregate posterior:

$$f_d(\bar{\boldsymbol{z}}, \boldsymbol{z}) = \Omega^{-1}(p|\boldsymbol{\tau}, \boldsymbol{\lambda}, \boldsymbol{\kappa}) = \boldsymbol{\lambda}\left(-\log\left(1-p\right)^{\frac{1}{\boldsymbol{\kappa}}}\right) - \boldsymbol{\tau} \tag{7}$$

The core set can now simply be obtained by picking the data points that are closest to the obtained distance values, if the chosen distance metric collapses the distance to a scalar, or directly to the latent vector, if the chosen distance metric preserves the dimensionality. Note that we have chosen to inversely sample the CDF of equation 6 in favor of a more compact equation. It should however be clear that equation 5 can alternately be sampled equivalently. The advantage of such a core set selection procedure is that we always attempt to approximate the underlying distribution, with the quality being defined by the desired amount of exemplars, while excluding statistical anomalies by limiting outlier probability values to e.g. $p < 0.95$. As anticipated, the latter plays the additional crucial role of robust application when the system has finished learning and is deployed.

5.5 Class incremental curricula and task order

Continual learning methods are mostly evaluated in the context of class incremental learning. The classes of a benchmark dataset are typically split into disjoint sets and introduced to the learner in alphabetical or class index sequence. Due to the large computational effort of training neural networks to convergence on long task sequences, several works choose to evaluate on subsets of classes (Rebuffi et al., 2017; Wu et al., 2019; De Lange et al., 2019;

Park et al., 2019). An important remaining question is thus how such evaluation affects comparability and reproducibility, or more generally the role of task order. As mentioned earlier, selecting a meaningful ordering is in most cases non-trivial. Large-scale dataset such as ImageNet are often composed by scraping data from the internet, social media or through uncontrolled acquisition that prioritizes as large as possible datasets. We as humans thus lack the knowledge to build an intuitive learning curriculum when paired with our lack of understanding of deep neural network representations. Consequently, scarcely any works have attempted to address this challenge beyond a simple randomization of the class order. Fortunately, we can provide at least a partial remedy to the seemingly arbitrary class incremental evaluation setting. Although we do not have access to explicit data distributions for any task, equation 6 allows us to assess the similarity of new tasks with the aggregate posterior for known tasks. In the spirit of our earlier formulated active learning query, we can start with any task t and proceed to select future tasks $t \in T$ that feature the least overlap with already encountered tasks (or most overlap, depending on what is desired):

$$t_{\texttt{next}} = \arg\max_{t \in T} \left\{ \mathbb{E}_{p_t(\boldsymbol{x})} \Omega_{\boldsymbol{\rho}} \left(\mathbb{E}_{q_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})} \left[z \right] \right) \right\} \tag{8}$$

To provide an example, if our objective was to incrementally expand a system to recognize individual animal species, one assumption could be to accelerate training by always including the species that is most similar to what has already been learned, as this could be hypothesized to require only small representational updates. An alternative objective could be to design a system that expands its knowledge in an attempt to cover and generalize to an as large as possible variety of concepts. In this scenario, one could choose to always include the next task with the smallest amount of overlap with existing tasks to maximize learning of diverse representations.

We could now delve into a philosophical debate on when it is reasonable to assume access to future tasks in continual learning to undergo above selection, and when the task sequence is unavoidably dictated by other external factors. We refrain from this discussion at this point and will instead focus on highlighting the large effect on performance when the task order is chosen by above mechanism in the following empirical investigation. At the very least, we hope that this will invoke a more careful and consistent evaluation on existing benchmarks, instead of picking arbitrary data subsets, selecting different random class orders and nevertheless attempting to compare results across methods.

6. Experimental verification and analysis

In this section we provide the empirical verification for the earlier introduced framework and its specific realization in deep neural networks. For this purpose, we start with a quantitative comparison of exemplar selection mechanisms to prevent catastrophic forgetting in continual learning and querying strategies in active learning. Here, we will first show that the proposed common EVT based foundation surpasses several conventionally employed techniques. We then proceed to further highlight the method's superiority in the open world. In contrast to most methods that are developed with a unidirectional focus on improving a specific active learning or continual learning benchmark, our framework has the critical advantage of not breaking down in the presence of corruptions that commonly occur in practical application

in the wild. To conclude the experimental section, we investigate the role of task order for evaluation. We show that a task curriculum constructed through our framework consistently results in considerable improvements.

We base our experiments on the MNIST (LeCun et al., 1998), CIFAR10 and 100 datasets (Krizhevsky, 2009). Although these datasets could be regarded as fairly simple, they are advocated as the predominant benchmarks in all of the presented continual learning works and still present a significant challenge in this context. They are further sufficient to point out major differences between methods, particularly with respect to robustness, showcasing a disconnect with real application and realistic evaluation. We use a 14 layer wide residual network (WRN) (Zagoruyko and Komodakis, 2016; He et al., 2016) encoder and decoder with a widening factor of 10, rectified linear unit activations, weight initialization according to He et al. (2015) and batch normalization (Ioffe and Szegedy, 2015) with $\epsilon = 10^{-5}$ at every layer, to reflect popular state-of-the-art practice. To avoid finding elaborate learning rate schedules or resorting to other excessive hyper-parameter tuning, we use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.001 and a sufficiently high-dimensional latent space of size 60 for all training. We use this common setting to corroborate our wholistic view and describe further details for specific experiments in consecutive subsections.

6.1 Exemplar selection and core set extraction

Before we dive into a quantitative comparison of methods that aim to alleviate catastrophic forgetting through the selection and maintenance of a core set, we need to address a potential evaluation obstacle. In continual learning works, the typical evaluation relies on monitoring the decay of a metric over time when training is conducted on new tasks and old tasks are retained by continued training on a few select exemplars. However, there seemingly is no common protocol of how these exemplars are interleaved. Apart from obvious factors such as the amount of chosen exemplars, works such as variational continual learning (Nguyen et al., 2018) use the exemplars only at the end of each task's training cycle to fine-tune and recover old tasks, whereas most other works (Rebuffi et al., 2017; Isele and Cosgun, 2018; Wu et al., 2019) simply concatenate exemplars with newly arriving data. Ultimately, the different works make use of different methods for exemplar selection and attempt to compare their effectiveness through the final metric, even though they are generally not trivially comparable due to their distinct choices of the training procedure.

To highlight this argument we have trained the typical split MNIST and CIFAR10 scenarios, where classes are introduced sequentially in pairs of two and only the new task's data is available to an incrementally growing single head classifier. The old task is approximated through a core set of size 2400 and 3000 respectively, i.e. we pick 240 and 300 exemplars per class that correspond to retention of 4% and 6% of the original data. We train the model for 150 epochs per task to assure convergence and interleave exemplars selected by our proposed EVT approach in three different manners: 1.) We conduct the predominant naive concatenation of the core set with the new task's data and continue training with mini-batch gradient descent that samples data uniformly (unbalanced mini-batch sampling). 2.) We recognize that the former combination and sampling leads to a heavy imbalance as the core set size is generally much smaller than the new task's available data. We naively correct this through weighted sampling that samples a mini-batch such that it consists in

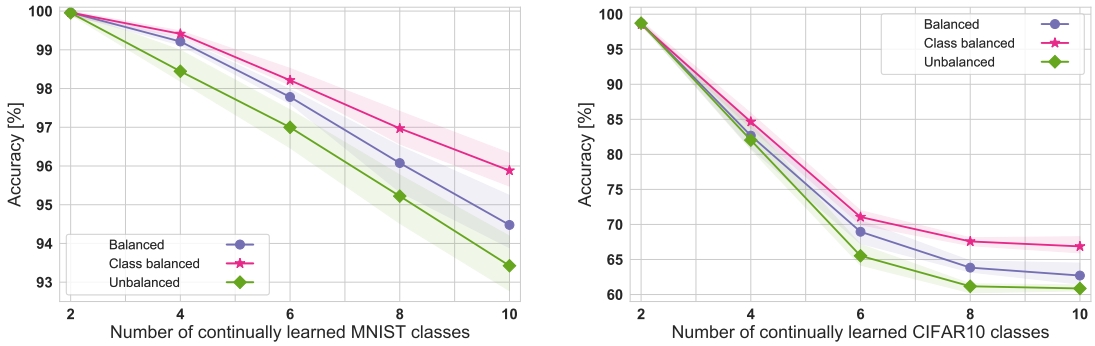


Figure 8: Influence of mini-batch sampling in continual learning with core sets on MNIST and CIFAR10. The green squared line represents unbalanced sampling, the naive practice of sampling mini-batches uniformly from the concatenated pool of the new task's data and the retained core set. The purple dotted line weights the sampling to oversample the much smaller core set to balance the mini-batch equally. The latter is further corrected with respect to classes in the pink starred line, where the sampling is adjusted to draw mini-batches that are comprised of the same amount of instances per class independently of their origin. We have repeated the experiments five times, illustrated by the shaded regions ranging from the minimum to the maximum obtained values. We can observe that such training details result in very significant performance differences beyond the statistical deviations of a specific core set selection strategy. This imposes an additional challenge in the evaluation of core sets for continual learning. Core sets have been selected with the proposed EVT based method and consist of 240 and 300 exemplars per class for MNIST and CIFAR10 respectively.

equal portions of former tasks' exemplars and new task's data, generally oversampling the exemplars (balanced mini-batch sampling). 3.) We identify that the latter weighted balanced sampling always results in an equal amount of exemplars and new data in a mini-batch, independently of the number of classes that the core set or the new task increment are comprised of. To correct for the number of classes, we further investigate class balanced sampling, where each mini-batch is sampled such that each class is equally represented. To give an example, if we have seen two tasks of two classes and proceed to learn the next task, the core set with its four classes will be oversampled to constitute two thirds of a mini-batch and the remaining third is made up of the two classes of the third task.

We show the obtained empirical continual learning accuracies in figure 8. With gaps of over 5% it is evident that balancing mini-batches is essential. More so, it is clear that a comparison of different core set works, just because they have used a similar core set size, can result in an apples to pears comparison if other aspects such as the detailed training procedure and mini-batch sampling are not taken into account. As our main focus is to analyze the core set selection strategies and their limitations, we proceed to compare different core set selection strategies in isolation from the precise continual learning setting. In analogy to Bachem et al. (2015) and the "reverse accuracy" evaluated in LLGAN (Zhai et al., 2019), we first train the model on the entire dataset, then select core sets of different sizes, and finally retrain the model exclusively on the core set to assess the approximation quality of

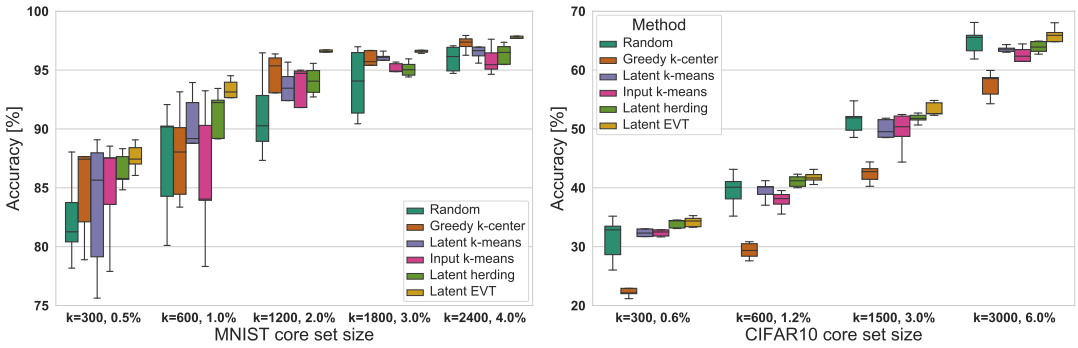


Figure 9: Training accuracy on core sets constructed by different popular strategies. Results for different core set sizes, characterized through their size k and the respective percentage of the dataset, are illustrated in a box plot to show the median, first and third quartile and minimum and maximum values obtained from five experimental repetitions. If viewed without color, methods are displayed from left to right in order of the legend from top to bottom.

our strategy. We repeat this entire procedure five times to gauge statistical consistency and estimate deviations. Without a doubt, methods that select a core set that yields a better approximation of the overall population and results in larger accuracies when trained in isolation, also provide better means to alleviate catastrophic forgetting in continual learning. We compare six different methods:

1. **Random:** select exemplars uniformly at random.
2. **Greedy k-center:** greedy k-center approximation (Gonzalez, 1985) for coreset selection as used in Variational Continual Learning (Nguyen et al., 2018). In essence, exemplars get picked one by one to obtain a cover of the distribution by maximizing their distance in latent space to all existing data points in the core set.
3. **Input k-means:** k-means clustering with k being equal to the number of exemplars. Raw data points get selected that are closest to each obtained mean. Suggested as an alternative to greedy k-center in variational continual learning (Nguyen et al., 2018).
4. **Latent k-means:** analogous to above input based k-means, but with the difference that the clustering is conducted on the lower dimensional latent embedding.
5. **Latent herding:** an adaptation of the herding procedure, used by Rebuffi et al. (2017); Wu et al. (2019), to operate on the latent space instead of an arbitrary neural network feature space. Herding greedily selects exemplars one by one such that each exemplar addition best approximates the overall data's mean embedding.
6. **Latent EVT:** our proposed EVT based inverse Weibull sampling introduced in sections 4 and 5.

We show the obtained accuracies by training on differently sized core sets selected by the above mechanisms in figure 9. As expected, random sampling features large variations, with

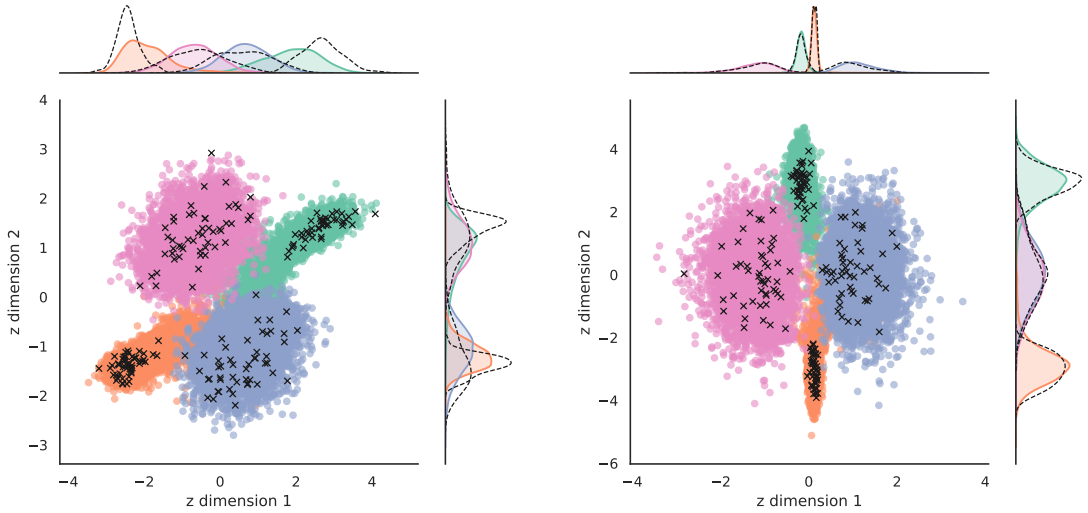


Figure 10: Visualization of the aggregate posterior for a model with two-dimensional latent space trained on the first four classes of the CIFAR10 dataset and 200 selected core set exemplars. The left panel shows the greedy k-center approach, whereas the right panel shows our proposed EVT based core set construction. Classes are color coded points and the core set elements are illustrated through black crosses. A kernel density estimate of the per class aggregate posterior (in color) and the corresponding distributional approximation of the selected core set elements (dashed black) are added on each dimension. In contrast to the greedy k-center approach that features large discrepancies, insignificant differences are observable for our proposed method, painting an intuitive picture for our methods quantitative success of figure 9.

the best attempts rivalling the other methods and in the worst case yielding substantially worse results. The k-means methods both perform similarly, with the latent space version operating on a lower-dimensional embedding showing minor improvements over the clustering obtained on the original image data. The smaller the core set size, the worse these methods seem to perform. This is not surprising and Bachem et al. (2015) have already argued that k-means with well separated clusters with sufficiently different amount of data points per cluster can be prone to inaccurately estimating multiple cluster centers in highly populated areas versus none in more sparsely populated clusters. This is further amplified by k-means generally necessitating a sub-sampled initialization to operate in high dimensions and at large scale. As such, we also observe larger variations for these methods. Latent herding is subject to much less overall variation and seems to initially do very well. However, in contrast to the proposed latent based EVT procedure, we notice an increasing gap in accuracy with larger core set sizes. Intuitively, we attribute this to herding picking increasingly redundant samples due to the objective relying exclusively on the best mean approximation, which does not simultaneously tend to diversity. Our latent based EVT approach that aims to approximate the underlying distribution features by far the least deviation and consistently outperforms all other methods.

To provide a better intuition, we have re-trained the model with a two-dimensional latent space to visualize the aggregate posterior and compare it with the selected core sets. Figure 10 shows the latent embedding with the first four CIFAR10 classes. The colored points correspond to the embedding of the entire set of data points and the respective curves correspond to kernel density estimates of the aggregate posterior. The black crosses indicate the points selected for a small core set of size 200, i.e. 50 per class. The left panel illustrates the greedy k-center approach, whereas the right panel shows the EVT aggregate posterior based approximation. Evidently, the approximation of the distribution is almost impeccable for our proposed approach, with the greedy alternative leaving much to be desired. We argue that this is due to the greedy k-center procedure optimizing for a cover based on maximal distances, alas without explicitly replicating the density or taking into account inherently present outliers and unrepresentative examples. While this might not be much of an issue for the highly redundant clean MNIST dataset, the arbitrarily collected real world data of the CIFAR10 dataset entails complete failure for the greedy k-center approach. In fact, by introducing a few naturally occurring image corruptions, we will show that such lack of robustness can be observed for all but our proposed method in a later section. Before we dive into this aspect of robust application in the open world, we first proceed with a quantitative analysis of the active learning perspective.

6.2 Active queries

In addition to the last section showing the advantages of our proposed framework for the construction of core sets that approximate the aggregate posterior, we empirically demonstrate the benefits when conducting EVT based queries for active learning. Recall that active learning is challenging because we generally desire to query batches of informative data at a time instead of querying, re-training and re-evaluating one by one. This is particularly imperative for computationally expensive deep learning and adds a further constraint of not only querying meaningful samples, but also making sure to query diversely without too much redundancy between the queried examples. We consider this typical deep active learning scenario for MNIST and CIFAR10, where we start with a random subset of 50 and 100 data points respectively, train for 100 epochs to assure convergence and then make a query to include 100 further data points. We then proceed to train the network with the additional instances before repeatedly querying and training again. In a crucial distinction to the majority of active learning works that only investigate the quality of the query by re-training the entire model from scratch, we do not reset our weights in continued incremental training. This implicitly introduces a stronger impact of ordering and further acknowledges that not only labelling, but also training itself is expensive. Each experiment is repeated five times, alas always with the same initial random subset to preserve comparability between individual repetitions and across methods.

We investigate popular metrics and mechanisms on which current deep active learning is based. The majority of these are techniques that attempt to take optimal action without explicitly approximating the entire set of unknown data. To estimate and account for uncertainty we make use of Monte Carlo Dropout (MCD) (Gal and Ghahramani, 2015) where appropriate. Although we believe that there is an inherent limitation in earlier introduced approaches that explicitly use the entire unlabelled pool for optimization, we

also investigate the proposed technique to query based on a k-means core set extracted from the unknown data (Nguyen and Smeulders, 2004; Sener and Savarese, 2018). Whereas we certainly regard such methods as valuable in a closed world context, we note that these methods are infeasible without prior knowledge outside of a constrained pool or for sequentially arriving data subsets. As we will see in the next section, they feature little robustness to nonsensical data that might be present in the pool, as the entire unlabelled pool is included and assumed to be useful. The metrics and methods that we investigate are:

1. **Random:** sampling uniformly at random from the unlabelled pool.
2. **Reconstruction loss:** in our particular scenario, because our proposed framework includes a generative model, we can query examples based on largest reconstruction loss. This is typically unavailable in a purely discriminative neural network classifier.
3. **K-means core set:** use the entire unlabelled pool to base the query on an extracted core set that is equivalent in size to the query amount. Nguyen et al. had suggested such pre-clustering (Nguyen and Smeulders, 2004) and it was later used in deep active learning with k-means as the core set algorithm (Sener and Savarese, 2018).
4. **MCD - classification confidence:** query based on lowest softmax confidence (Lewis and Gale, 1994). As neural network classifiers are known to be overconfident, we additionally gauge uncertainty with MCD as a suggested remedy by Gal et al. (2017).
5. **MCD - classification entropy:** query based on largest predictive entropy (MacKay, 1992). Similar to lowest confidence, we use uncertainty from MCD to obtain better entropy estimates (Gal et al., 2017).
6. **Latent EVT:** our proposed EVT based approach that balances exploration with exploitation by querying instances that distribute across outlier probabilities, but limited by an upper rejection prior to avoid uninformative outliers.

We first note that we have included classification confidence and entropy with MCD because omitting uncertainty estimates resulted in no improvement of the active learning query upon simple random selection. This has previously been argued and corresponds to the empirical observations made by Sinha et al. (2019). For our proposed EVT approach we empirically distribute the query uniformly across examples that fall into the range of 0.5 to 0.95 outlier probability, as estimated by equation 6. Although it never occurred in practice, we note that it would likely be preferential to extend this range to the lower end if not enough samples in the pool were available in the mentioned range, rather than including complete outliers. We will provide empirical evidence for this in the next section.

Figure 11 shows the quantitative results of our active learning experiments. On both datasets, the k-means based core set is either similar or slightly worse than simply sampling at random. This reflects our previous observations in the core set continual learning section. On the contrary, the uncertainty based methods surpass random sampling. Using largest reconstruction loss similar results can be accomplished, although at the additional computational expense of calculating the decoding. However, all methods are significantly outperformed by our proposed latent EVT method at all times. The respective rationale

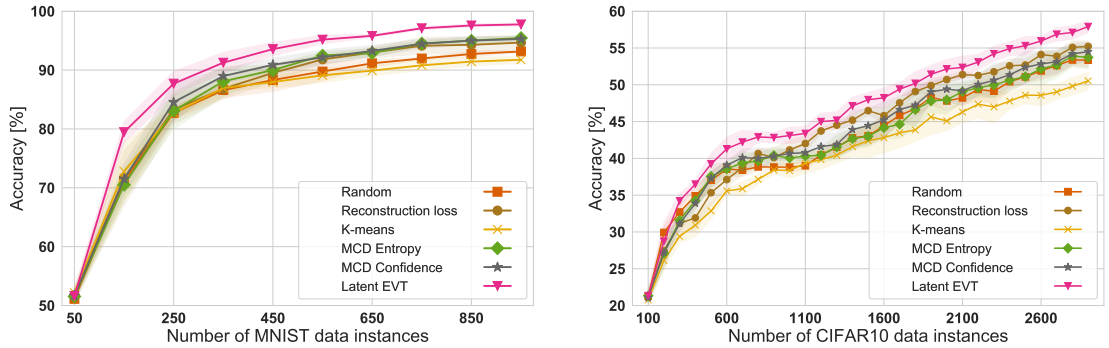


Figure 11: Active learning accuracy for different methods on the MNIST and CIFAR10 datasets. All experiments start with the same randomly sampled 50 and 100 dataset examples. In each step, an additional 100 data instances are queried from the remaining unlabelled pool and included for further continued training. Results show the average over five experiments, with the shaded areas ranging from the minimum to the maximum obtained values.

behind this improvement is quite intuitive. Our strategy balances completely novel examples with less novel examples that are still required to strengthen the existing learned features. More importantly, it rejects uninformative outliers that are inherently present in the pool, a threat that uncertainty based methods can be particularly prone to. This threat is magnified with even less knowledge about the acquired dataset and even more unconstrained data acquisition. The past two subsections have focused on showing our methods advantage in the typical continual and active learning benchmark perspective in the closed world scenario, devoid of any analysis with respect to robustness. In the next section we extend this evaluation to analyze each individual methods' behavior in the presence of corruptions.

6.3 Robustness to open world corruptions

Prior works that address open set recognition or in general application of machine learning algorithms in an open world have argued that prediction on previously unseen unknown classes results in inevitable misprediction (Scheirer et al., 2013, 2014; Bendale and Boult, 2015, 2016; Boult et al., 2019). For example, if a user is given the freedom to provide any image input to a neural network based classifier, an arbitrarily chosen image's prediction will be indistinguishable from the typical training set output. We have previously empirically demonstrated that the proposed EVT based approach overcomes this challenge, much in contrast to relying on uncertainty based measures that fail to even distinguish the most trivially disparate datasets such as visual and audio data (Mundt et al., 2019a,b). Although this poses a serious threat to building a user's trust, just imagine your own faith in a classifier that assigns an image of a car the label of a t-shirt (recall the earlier figure 5), we can naturally question if this scenario could simply be circumvented by including guidelines with respect to the expected model input, i.e. "this model has been trained on fashion-items, it is not designed for other types of data". The more sensible solution would be to have the model reject unknown unknown data. Whether or not we consider the latter scenario as meaningful, unknown unknown data is not necessarily always composed of completely dissimilar classes.

A perhaps at least equivalently large threat is data that is statistically deviating for other reasons: corruption and perturbation. In any real-world scenario, we can no longer assume that our machine learning model is faced exclusively with the carefully curated data that benchmarks are comprised off. Often a simple change in camera can dramatically skew the statistics of the acquired image. In an almost endless list, low lighting conditions can introduce various forms of noise, small jitter can cause blur, weather conditions change, the condition of the object of interest correspondingly changes, ... The gullible solution would again be to attempt to model all forms of corruptions and perturbations, but this simply connects back to the infeasibility of the earlier introduced "inference with the universum" approach.

In a recent effort to benchmark the performance against 15 types of various corruptions, Hendrycks and Dietterich (2019) have shown that none of the developed neural network models feature any intrinsic robustness, even if they converge to more accurate solutions on the initial benchmark. This was concluded from experiments where neural networks are trained on the uncorrupted benchmark dataset and evaluated on the corrupted data. We extend this evaluation by investigating the presence of a minor portion of corrupted data in the training process, as can realistically be assumed for active or continual learning. We examine whether common query strategies in active learning and core set construction in continual learning are robust, or whether querying and including this unrepresentative corrupted data into core sets leads to performance degradation in comparison with the clean benchmark. We believe that this is critical for two reasons: 1.) The necessity to carefully curate every single example in the unknown data pool can outweigh the active learning human labelling effort and thus renders active learning ineffective in the first place. 2.) Data cleaning itself is extremely challenging and it is often not immediately clear whether the inclusion of a data instance is beneficial or is accompanied by side effects.

We make use of corruptions across four categories: noise, blur, weather and digital corruptions, as introduced by Hendrycks and Dietterich (2019). These can further be distinguished into 15 types: low-lighting Gaussian noise, electronic shot noise, bit error impulse noise, speckle noise, Gaussian blur, defocus blur, glass blur, zoom blur, motion blur, snow, fog, brightness, contrast, saturation and elastic deformations. Each corruption is algorithmically generated with five discretized levels of severity, of which the first two are at times barely discernible from a typical image by a human. We accordingly corrupt 7.5% of the data across these 75 corruptions. We add the additional constraint that each image can only be corrupted once. Note that in principle some corruptions, such as noise resulting from low lighting conditions and out of focus blurring, could occur simultaneously. We have deliberately chosen this amount of corruption to, on the one hand be small enough to not affect overall performance if trained on the entire dataset, on the other hand be larger than the core set size or active learning query amounts used in previous sections. Hypothetically, in the absolute worst case this could result in only corrupted images being selected and the entire chosen set being much less representative of the complete dataset than a selection of clean examples would be. We repeat the previous CIFAR10 experiments under these conditions. For better visualization and quantification we do not show plots, but have instead picked three evenly spaced points of figures 9 and 11.

We show the originally obtained results in direct comparison with the results obtained under inclusion of the corrupted data in tables 1 and 2. From these quantitative results

Table 1: Active learning with and without partial dataset corruption. Uncorrupted values correspond to those visualized in figure 11.

	Accuracy [%]: mean $^{+\text{difference to maximum}}_{-\text{difference to minimum}}$					
CIFAR10 queries, dataset size	8, 900		18, 1900		28, 2900	
Dataset	regular	corrupted	regular	corrupted	regular	corrupted
Random	$38.80^{+0.69}_{-1.75}$	$38.97^{+1.03}_{-1.87}$	$47.81^{+2.02}_{-3.93}$	$47.91^{+2.13}_{-3.58}$	$53.36^{+1.17}_{-2.34}$	$53.53^{+1.13}_{-2.42}$
Reconstruction loss	$41.14^{+2.06}_{-3.89}$	$38.26^{+0.64}_{-1.89}$	$50.70^{+0.69}_{-1.50}$	$46.49^{+0.82}_{-2.13}$	$55.22^{+1.37}_{-1.92}$	$50.85^{+1.03}_{-1.57}$
K-means	$38.34^{+1.46}_{-2.63}$	$36.05^{+1.65}_{-2.53}$	$45.08^{+1.50}_{-3.23}$	$42.93^{+1.59}_{-3.65}$	$50.52^{+0.94}_{-3.15}$	$47.58^{+1.93}_{-3.39}$
MCD Entropy	$40.05^{+1.15}_{-2.99}$	$38.83^{+0.68}_{-1.03}$	$47.96^{+2.91}_{-5.28}$	$44.73^{+0.61}_{-1.02}$	$53.72^{+2.35}_{-4.76}$	$50.06^{+0.37}_{-0.75}$
MCD Confidence	$40.67^{+0.87}_{-1.89}$	$37.93^{+0.35}_{-0.81}$	$49.40^{+2.86}_{-4.44}$	$47.16^{+1.29}_{-3.22}$	$54.51^{+1.15}_{-3.13}$	$51.91^{+1.78}_{-2.67}$
Latent EVT	$44.67^{+0.32}_{-0.63}$	$43.79^{+0.74}_{-1.72}$	$51.66^{+1.05}_{-1.69}$	$51.12^{+0.38}_{-0.91}$	$57.43^{+0.51}_{-1.09}$	$56.83^{+0.41}_{-0.78}$

Table 2: Coreset selection and training with and without dataset corruption. Uncorrupted values correspond to those visualized in figure 9.

	Accuracy [%]: mean $^{+\text{difference to maximum}}_{-\text{difference to minimum}}$					
CIFAR10 coreset size	300		600		1500	
Dataset	regular	corrupted	regular	corrupted	regular	corrupted
Random	$31.23^{+3.94}_{-9.14}$	$30.35^{+1.88}_{-5.92}$	$39.52^{+3.61}_{-7.95}$	$39.05^{+1.99}_{-5.89}$	$51.43^{+3.33}_{-6.12}$	$51.01^{+2.30}_{-4.49}$
Greedy k-center	$22.82^{+3.05}_{-1.65}$	$22.19^{+1.76}_{-3.37}$	$29.33^{+1.50}_{-3.23}$	$29.48^{+1.91}_{-5.11}$	$42.41^{+1.97}_{-4.13}$	$42.37^{+1.49}_{-2.44}$
Latent k-means	$32.76^{+2.29}_{-3.35}$	$29.00^{+2.12}_{-4.05}$	$39.49^{+1.71}_{-4.17}$	$35.71^{+1.69}_{-4.08}$	$50.01^{+1.80}_{-3.28}$	$48.52^{+2.59}_{-3.86}$
Image k-means	$32.85^{+2.57}_{-3.76}$	$30.74^{+1.43}_{-3.16}$	$37.86^{+1.66}_{-3.98}$	$36.38^{+0.90}_{-2.75}$	$49.62^{+2.83}_{-8.09}$	$48.23^{+1.78}_{-2.50}$
Latent herding	$33.92^{+0.61}_{-1.45}$	$33.81^{+0.82}_{-1.39}$	$41.13^{+1.18}_{-2.29}$	$40.77^{+1.34}_{-1.57}$	$51.87^{+1.12}_{-1.85}$	$51.06^{+2.43}_{-2.30}$
Latent EVT	$34.16^{+1.10}_{-2.27}$	$34.18^{+1.07}_{-2.55}$	$41.78^{+1.34}_{-2.57}$	$41.67^{+1.37}_{-2.53}$	$53.35^{+1.48}_{-2.53}$	$53.28^{+1.06}_{-2.17}$

it is evident that only two techniques are robust in active learning: random sampling and our proposed EVT based approach. The logical explanation is that random sampling on average will pick roughly 7.5% corrupted data, of which another 40% feature only minor low severities. The small amount thus only has minor effect on the optimization. The EVT based algorithm is similarly unaffected as it does not query statistical outliers in the first place, or if it includes corrupted examples then only those with minor severity that are statistically still largely similar to the uncorrupted data. All other methods are prone to the corrupted outliers in one way or another. Classifier uncertainty and reconstruction loss tend to pick very corrupted examples by definition, the k-means approach will have shifted centers or falsely query from new clusters that are centered around corruptions of the unknown pool. Looking at the quantitative accuracy values, we can in fact even conclude that all these methods perform worse than a simple random query. The continual learning core set construction picture is quite similar. Here, we can observe corruption robustness for random sampling, latent herding and our proposed approach. Latent herding is robust to outliers because it picks samples greedily one by one to best approximate the mean, which intuitively involves picking the next best example that is close to the class mean and does not involve outliers (potentially only in a minor fashion through a drifted mean if the outliers are not embedded symmetrically around the class mean). However, the issue of including redundant

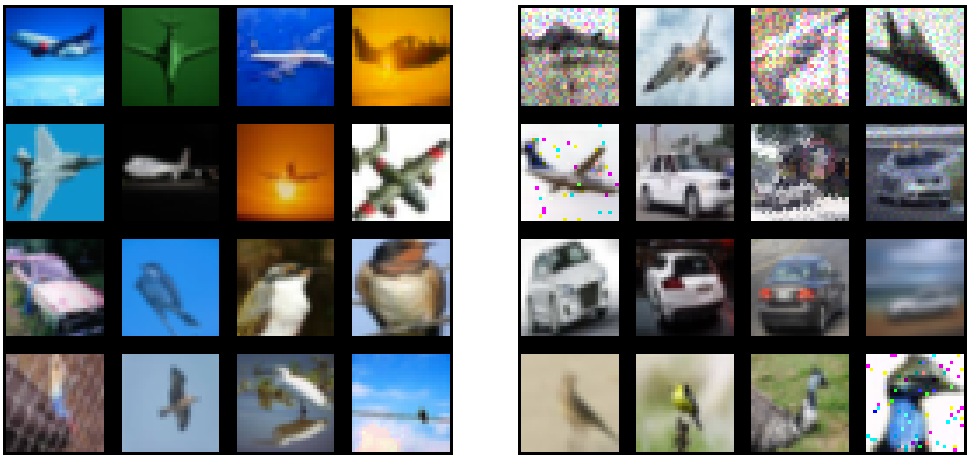


Figure 12: Typically selected dataset examples in the core set construction using a greedy k-center algorithm. Qualitative illustration is intended to provide intuition for a method's failure. The left panel shows how picked exemplars from an uncorrupted dataset are unrepresentative of the average image, with unusual backgrounds, occlusion and scaling issues. The right panel shows how the core set is comprised of many corrupted examples if a small portion of the dataset is corrupted, a lack of robustness that many methods in tables 1 and 2 suffer from.

samples into the core set remains unaddressed, and our EVT based method nevertheless outperforms all other approaches.

Interestingly, the greedy k-center approach also seems to be robust to the corruptions, although it performs equally miserably to the uncorrupted scenario. Recall that this algorithm greedily chooses the next data point for inclusion in a farthest-first traversal, by maximizing the distance to all presently existing core set elements. In other words, outliers are always queried as they by definition are farthest away. Only after a sufficiently large cover is obtained will representative data be queried. Because such unrepresentative outliers are already present in the uncorrupted data, the performance is consequently always low for small core set sizes. To visually illustrate this statement we show a uniform sub-sample of the acquired core set for the first four classes with and without corruption in figure 12. In the left panel we can observe the core set being comprised of atypical aeroplanes with deep green or black background, a captured overexposed sunset, partially occluded cars and birds by bushes and fences or images where the animal is almost not discernible and comprises only a fraction of the image. Arguably these do not represent good exemplars. In the right panel, we can see that in the presence of corruption, the core set is comprised of noisy, blurry and otherwise distorted images. Ultimately neither of these core sets are a particularly good approximation of the dataset, intuitively explaining the abysmal performance of this technique.

6.4 Choosing the curriculum - the importance of task order

As detailed in the earlier introduction of our framework, we can apply our proposed EVT based active learning strategy to the construction of a continual, class incremental learning curriculum. In this context, a task's outlier probability is synonymous with its dissimilarity to already accumulated tasks. Conversely, a task that is deemed to be largely inlying has a large representational overlap with existing knowledge, even though it might have been assigned a distinct label. In the best case scenario, this implies that only fine-tuning is necessary to sufficiently include a proximate task. In the worst case scenario, the representational entanglement severely limits the discriminability. Unless a major addition or overhaul of the learned representations ensues, this leads to confusion with existing concepts. In contrast, most outlying tasks are hypothesized to be distinct enough to not interfere with previous tasks, assuming the old task's data is still available or a continual learning mechanism prevents its catastrophic forgetting.

We investigate the importance of task order and whether the construction of a curriculum beyond alphabetical class order provides substantial learning benefits. For this purpose we consider four conceivable scenarios:

1. **Class sequential ordering:** learn the classes in order of their integer class label. For many datasets this is in alphabetical order.

2. **Random order:** randomized class order.

3. **Most outlying, dissimilar tasks first:** determine the next class to add by evaluating equation 8, i.e. pick the next class that is most outlying and dissimilar with respect to the already seen classes.

4. **Most inlying, similar tasks first:** determine the next class to add by evaluating equation 8, but with a minimum over task outlier probabilities to include the most similar task in each increment.

Note that for all strategies we always start with the same first task for comparability. To make sure that obtained results and found curricula are not just a result of sheer luck, we repeat each experiment five times, report the average and the minimum and maximum obtained accuracies at each step to gauge deviations. We conduct experiments on two datasets: the CIFAR100 and the AudioMNIST (Becker et al., 2018) dataset. We follow the typical continual incremental learning procedure of adding classes in pairs of two. We chose the first dataset because it allows for the construction of a long task sequence. We chose the latter because it represents a non-image dataset and previous work has observed that some classes can provide strong retrospective improvement (Mundt et al., 2019a), an early indicator that the class ordering should be investigated further. In order to show the impact of task ordering, we provide an analysis, both, when independently evaluated from, or coupled to specific techniques that alleviate continual learning catastrophic forgetting. As such, we evaluate CIFAR100 in what is typically referred to as a continual learning upper-bound, i.e. the maximum obtainable accuracy given a specific model choice and training procedure in which the data of each task is simply accumulated with each subsequent task. For the AudioMNIST we use generative replay to prevent catastrophic forgetting,

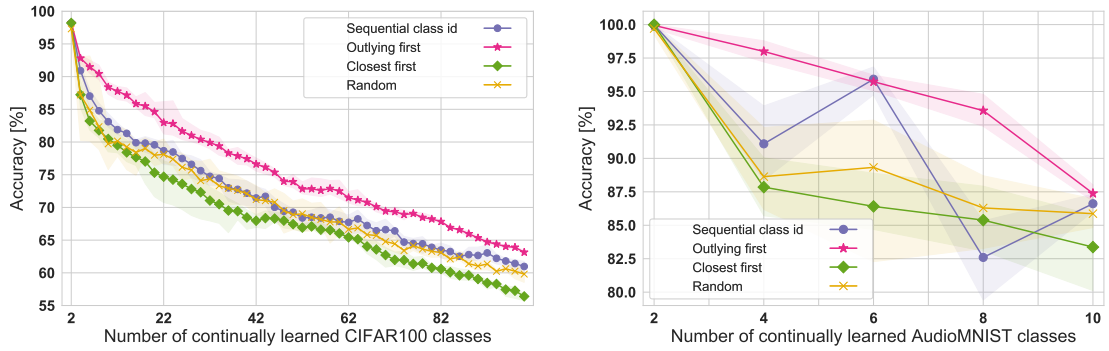


Figure 13: Continual learning accuracy of learning classes in increments of two in dependence on the choice of task order. Top panel shows the incremental upper-bound, i.e. a simple accumulation of the real data, for the CIFAR100 dataset. The bottom panel shows obtained performance on the AudioMNIST dataset with alleviated catastrophic forgetting through generative replay. For each of the order selection mechanisms the experiment has been repeated five times. The corresponding average together with the maximum and minimum deviation are reported respectively.

where old tasks' data is rehearsed based on the trained generative model. We do not make use of any data augmentation.

The achieved accuracies at each task increment are shown in figure 13. We can observe that for the CIFAR100 dataset, random sampling seems to yield a very similar accuracy trajectory in comparison to sequentially learning the classes in order of their alphabetical class id, resembling earlier observations (De Lange et al., 2019; Javed and Shafait, 2018). However, in contrast to the conclusion that task order is negligible, we can observe that our proposed framework's selection schemes, that rank order the data according to their similarity with the existing encoding, paint a dramatically different picture. Selecting the most dissimilar task for inclusion consistently improves the accuracy by several percent, even at the end of training. Conversely, including tasks that are very proximate to existing concepts results in an all-time performance decrease. We hypothesize that this is due to the classifier experiencing immediate confusion. Our initial classes consist of "apples" and "aquarium fish" and the query consensus across repeated experiments is to continue with selecting the classes "pears" and "whale" or "shark". The opposite strategy that prioritizes dissimilarity in the curriculum instead includes unrelated classes such as "lawnmower", "mountain" or "oak". We believe that this allows the model to more rapidly acquire a diverse set of representations.

We can draw almost analogous conclusions for continually learning the AudioMNIST dataset with generative replay. Here, we additionally see that the conventional order of learning the sounds from "zero" to "nine" is accompanied by a pattern of repeated retrospective improvement. The first task increment results in a larger accuracy drop, that is rectified through backwards improvement of the next task increment. This pattern repeats for the next two classes and its consistent strong emergence is only visible when learning sequentially in order of class id. The accuracy at any time is again best for our proposed measure of dissimilarity and worst when selecting according to task proximity. For the latter,

in analogy to the earlier hypothesized confusion of the classifier, the generative model is faced with difficulty to disambiguate the resembling classes and produce unambiguous output.

Our results indicate that using active learning techniques in continual learning can have critical impact on the achieved performance. More so, the results provide an important signal for reproducibility and significance of various conjured continual learning benchmarks. In a world of benchmarking methods and regularly claiming advances when a method surpasses another by 1-2 %, the observed absolute discrepancy between the different task orders for CIFAR100 is as large as 10%. This is a substantial gap. Whereas we obviously believe that there is value in analyzing and contrasting different techniques to alleviate catastrophic forgetting on a common dataset, it is clear that there is still much we need to learn about neural network training and evaluation that can only be discovered by moving away from our current rigid benchmarks.

7. Conclusion: towards a wholistic definition of deep continual learning

We have presented a common viewpoint to naturally unite robust continual and active learning in the presence of the unknown. For each aspect, we have conducted an empirical investigation that demonstrated the benefits of the viewpoint's realization in a variational Bayesian deep neural network framework. Needless to say, each of our individually presented experiments can be extended with multiple facets and several nuanced applications can be derived and thoroughly investigated. At this point, we remark that we do not wish to claim that our proposed method provides the generally best solution or selects optimal task sequences. Although our framework clearly shows quantitative promise, our main goal is to highlight the importance of the introduced consolidated viewpoint. In the ideal case, we would encourage future works to adopt our framework or take a similarly wholistic approach. At the very minimum, we would expect future works to rethink current practices and question whether current benchmarks are a realistic reflection of our desiderata for continual machine learning systems. As illustrated throughout the paper, this necessitates stepping out of our closed world benchmark routines. In hopes of providing some guidelines for the latter, we make an attempt at a revised continual learning definition and suggestions towards more systematic assessment.

Definition 4 *Continual Machine Learning - this work: The learner performs a sequence of N continual learning tasks, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$, that are distinct from each other in terms of shifts in the underlying data distribution. The latter can imply a change in objective, transitions between different domains or inclusion of new modalities. At any point in time, the learner must be able to robustly identify unseen unknown data instances and rank order them according to similarity with existing tasks, in order to actively build a learning curriculum. If the system is desired to be supervised, a human in the loop may group and label the set of identified unseen unknowns to explicitly guide future learning. When faced with a selected (N+1)th task $\mathcal{T}_{N+1}$ (which is called the new or current task) with its data $\mathcal{D}_{N+1}$, the learner should leverage its dictionary of representations to accelerate learning of $\mathcal{T}_{N+1}$ (forward transfer), extend the dictionary with unique representations obtained from the new task's data (this can be completely new types of dictionary elements), while simultaneously maintaining and improving the existing representational dictionary with respect to former tasks (backward transfer).*

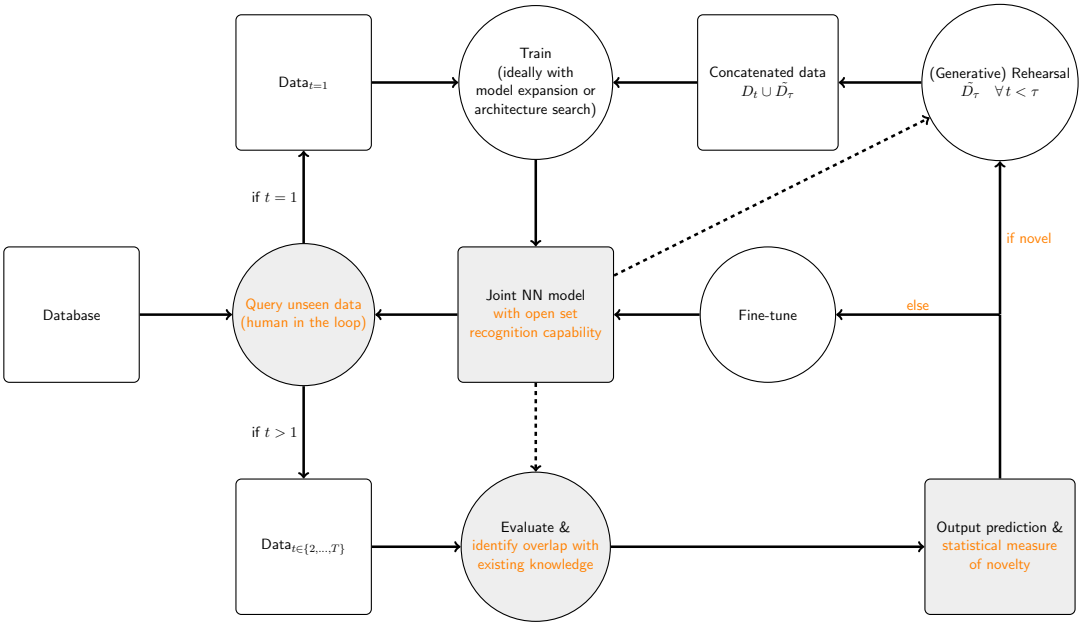


Figure 14: A suggestion for a more comprehensive, system oriented evaluation. In contrast to the conventional continual learning pipeline, the system is extended with a (optionally human in-the-loop) data querying mechanism and a measure of novelty that is used for robust application in the open world and to select adequate ensuing optimization techniques. These suggested additions to the conventional continual learning process are emphasized through orange text and shading in the diagram. Rectangles represent objects and circles correspond to processes. Dashed arrows indicate a process' dependency on the model.

In comparison with former continual learning definitions, reiterated at the beginning of this paper, the definition is now extended to include active data queries, the corresponding importance of data choice and task order, in coherence with awareness of the open world.

7.1 Outlook: a suggestion for a more comprehensive, system oriented evaluation

We show one example of how a revised outline of a continual learning system that satisfies the above definition could look like in figure 14. Again, this example can be realized with our proposed specific EVT based framework, although several other implementations are conceivable. The main idea of the system can be summarized as follows: After initial training on some seeding data, ideally by finding a baseline architecture through architecture search or through progressive architecture growth, a new task is queried through an inherent model mechanism to optimize the effect of order and the queried data is consecutively labelled. Alternatively, specific data can be introduced by a human in the loop, if it is desired that the system is constrained to very specific tasks. The new data is then evaluated with respect to existing tasks and associated with a measure of novelty. This measure of novelty serves the dual purpose of introducing robustness into the system when applied in the wild, and at the same time is used as the foundation to decide on how to proceed with further optimization. If

the overlap with existing knowledge is very large, it is sufficient to conduct minor fine-tuning steps. If there is a large amount of expected novelty, the optimization needs to proceed with a mechanism to protect previously acquired tasks, typically through means of core set or generative rehearsal. Because the amount of expected novelty is large, it is recommended to then continue training with model expansion in order to ensure sufficient representational capacity is available to accommodate entirely new concepts. The cycle is then repeated.

In comparison with the classical continual learning evaluation pipeline, presented in the beginning of this work in figure 2, we thus suggest to extend the system with essential robust evaluation and active queries to address questions concerning the importance of input data selection. As demonstrated, integration of these aspects can be achieved through prediction of a statistical measure of novelty based on overlap with existing knowledge, e.g. with our suggested posterior based EVT open set recognition approach. This measure of novelty serves a natural triple purpose: 1.) Rejection or setting aside of unknown unknown data in robust application. 2.) Querying data from an unlabelled pool in a suitable order that provides large expected benefit to the model. 3.) If the data order is pre-imposed, e.g by a human or a stream, the novelty metric can be used to dynamically switch the training procedure to incorporate dissimilar novel data, while preserving prior representations through extensive continual learning mechanisms that alleviate catastrophic forgetting, or to simply fine-tune in the presence of sufficient overlap with previously seen data.

Even though the advantages of expanding the effective representational capacity during training are clear, we have put the use of model expansion and progressive architecture search in brackets. Although its use is theoretically and empirically desirable, we understand that this ideal evaluation involves several challenges that can limit its practicality. It is clear from previously discussed works that continuous model growing is advantageous, but we note that heavily over-parametrized models have shown satisfactory results. We thus encourage future research to first and foremost focus on the questions about benchmark construction, data point selection, and the voiced concerns regarding robust application in the open world. We would then expect future work to additionally include model expansion techniques.

We anticipate that this work leads to increased awareness of the dangers of our current closed world practices and the necessity of expanding our views towards more realistic real-world relevant evaluation. In doing so, we believe that further synergies between presently separately treated machine learning paradigms will be exposed and can be exploited. This should ultimately lead to improved, more robust and simpler machine learning systems.

References

Yaser S. Abu-Mostafa. Learning from hints in neural networks. *Journal of Complexity*, 6(2): 192–198, 1990.

Alessandro Achille, Tom Eccles, Loic Matthey, Christopher P. Burgess, Nick Watters, Alexander Lerchner, and Irina Higgins. Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies. *Neural Information Processing Systems (NeurIPS)*, 2018.

Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based Continual Learning with Adaptive Regularization. *Neural Information Processing Systems (NeurIPS)*, 2019.

James B. Aimone, Yan Li, Star W. Lee, Gregory D. Clemenson, Wei Deng, and Fred H. Gage. Regulation and function of adult neurogenesis: from genes to cognition. *Physiological reviews*, 94(4):991–1026, 2014.

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory Aware Synapses: Learning what (not) to forget. *European Conference on Computer Vision (ECCV)*, 2018.

Bernard Ans and Stephane Rousset. Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Life Sciences*, pages 989–997, 1997.

Timur Ash. Dynamic Node Creation in Backpropagation Networks. *Connection Science*, 1 (4):365–375, 1989.

Les E. Atlas, David A. Cohn, Richard Ladner, Mohamed A. El-Sharkawi, Robert J. Marks II, M. E. Aggoune, and D. C. Park. Training connectionist networks with queries and selective sampling. *Neural Information Processing Systems (NeurIPS)*, 1990.

Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for Nonparametric Estimation - the Case of DP-Means. *International Conference on Machine Learning (ICML)*, 37: 209–217, 2015.

Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, 2008.

Soeren Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. *arXiv preprint arXiv: 1807.03418*, 2018.

Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the Materials in Context Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The Power of Ensembles for Active Learning in Image Classification. *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Abhijit Bendale and Terrance E. Boult. Towards Open World Recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

Abhijit Bendale and Terrance E. Boult. Towards Open Set Deep Networks. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. *International Conference on Machine Learning (ICML)*, 2009.

Christopher M. Bishop. Novelty detection and neural network validation. *IEE Proceedings: Vision, Image and Signal Processing*, 141(4):217–222, 1994.

Léon Bottou. Online Learning and Stochastic Approximations. In *Online Learning in Neural Networks*, pages 9–42. 1999.

Terrance E. Boult, Steve Cruz, Akshay R. Dhamija, Manuel Gunther, James Henrydoss, and Walter J. Scheirer. Learning and the Unknown : Surveying Steps Toward Open World Recognition. *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-VAE. *Neural Information Processing Systems (NeurIPS), Workshop on Learning Disentangled Representations*, 2017.

Rich Caruana. Multitask Learning. *Machine Learning*, 28:41–75, 1997.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. *European Conference on Computer Vision (ECCV)*, 2018.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. *International Conference on Learning Representations (ICLR)*, 2019.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2Net: Accelerating learning via knowledge transfer. *International Conference on Learning Representations (ICLR)*, 2016.

Zhiyuan Chen and Bing Liu. *Lifelong Machine Learning*, volume 33. Morgan and Claypool, 2017.

Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep convolutional filter banks for texture recognition and segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. *Neural Information Processing Systems (NeurIPS)*, 2005.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *arXiv preprint arXiv: 1909.08383*, 2019.

Akshay R. Dhamija, Manuel Günther, and Terrance E. Boult. Reducing Network Agnostophobia. *Neural Information Processing Systems (NeurIPS)*, 2018.

Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: new metrics for Continual Learning. *Neural Information Processing Systems (NeurIPS), Continual Learning Workshop*, 2018.

Timothy J. Draelos, Nadine E. Miner, Christopher C. Lamb, Jonathan A. Cox, Craig M. Vineyard, Kristofor D. Carlson, William M. Severa, Conrad D. James, and James B. Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. *International Joint Conference on Neural Networks (IJCNN)*, pages 526–533, 2017.

Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided Continual Learning with Bayesian Neural Networks. *International Conference on Learning Representations (ICLR)*, 2020.

Sebastian Farquhar and Yarin Gal. A Unifying Bayesian View of Continual Learning. *Neural Information Processing Systems (NeurIPS) Bayesian Deep Learning Workshop*, 2018a.

Sebastian Farquhar and Yarin Gal. Towards Robust Evaluations of Continual Learning. *International Conference on Machine Learning (ICML), Lifelong Learning: A Reinforcement Learning Approach Workshop*, 2018b.

Geli Fei, Shuai Wang, and Bing Liu. Learning cumulatively to become more knowledgeable. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1565–1574, 2016.

Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):594–611, 2006.

Qianyu Feng, Guoliang Kang, Hehe Fan, and Yi Yang. Attract or Distract: Exploit the Margin of Open Set. *International Conference on Computer Vision (ICCV)*, 2019.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. PathNet: Evolution Channels Gradient Descent in Super Neural Networks. *arXiv preprint arXiv:1701.08734*, 2017.

Michael Fink. Object classification from a single example utilizing class relevance metrics. *Neural Information Processing Systems (NeurIPS)*, 2005.

Robert M. French. Semi-distributed Representations and Catastrophic Forgetting in Connectionist Networks. *Connection Science*, 4(3-4):365–377, 1992.

Robert M. French. Pseudo-recurrent Connectionist Networks: An Approach to the 'Sensitivity-Stability' Dilemma. *Connection Science*, 9(4):353–380, 1997.

Yoav Freund and Robert E. Schapire. A decision theoretic generalisation of online learning and an application to boosting. *Computer and System Sciences*, 55(1):119–139, 1997.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation : Representing Model Uncertainty in Deep Learning. *International Conference on Machine Learning (ICML)*, 48, 2015.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. *International Conference on Machine Learning (ICML)*, 2017.

Yonatan Geifman and Ran El-Yaniv. Deep Active Learning with a Neural Architecture Search. *Neural Information Processing Systems (NeurIPS)*, 2019.

Robert Geirhos, Claudio Michaelis, Felix A. Wichmann, Patricia Rubisch, Matthias Bethge, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations (ICLR)*, 2019.

Alexander Gepperth and Cem Karaoguz. A Bio-Inspired Incremental Learning Architecture for Applied Perceptual Problems. *Cognitive Computation*, 8(5):924–934, 2016.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(C):293–306, 1985.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Neural Information Processing Systems (NeurIPS)*, 2014.

Charles G. Gross. Neurogenesis in the adult brain: Death of a dogma. *Nature Reviews Neuroscience*, 1(1):67–73, 2000.

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's Agree to Agree: Neural Networks Share Classification Order on Real Datasets. *International Conference on Learning Representations (ICLR)*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *International Conference on Computer Vision (ICCV)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Donald O. Hebb. *The Organization of Behavior; A Neuropsychological Theory.* John Wiley & Sons, Chapman & Hall, 1949.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR)*, 2017.

Tom M. Heskes and Bert Kappen. On-line learning processes in artificial neural networks. *Mathematical Foundations of Neural Networks*, 51(C):199–233, 1993.

Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *NeurIPS Deep Learning Workshop*, 2014.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

Matthew D. Hoffman and Matthew J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. *Neural Information Processing Systems (NeurIPS), Advances in Approximate Bayesian Inference Workshop*, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial Examples are not Bugs, they are Features. *Neural Information Processing Systems (NeurIPS)*, 2019.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning (ICML)*, 2015.

David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Khurram Javed and Faisal Shafait. Revisiting Distillation and Incremental Classifier Learning. *Asian Conference on Computer Vision (ACCV)*, 2018.

Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Ronald Kemker and Christopher Kanan. FearNet: Brain-inspired model for incremental learning. *International Conference on Learning Representations (ICLR)*, 2018.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring Catastrophic Forgetting in Neural Networks. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *British Machine Vision Conference (BMVC)*, 2017.

Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations (ICLR)*, 2013.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 114(13):3521–3526, 2017.

Ksenia Konyushkova, Sznitman Raphael, and Pascal Fua. Learning active learning from data. *Neural Information Processing Systems (NeurIPS)*, 2017.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Toronto, 2009.

Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), 2019.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples. *International Conference on Learning Representations (ICLR)*, 2018a.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. *Neural Information Processing Systems (NeurIPS)*, 2018b.

Sang Woo Lee, Jin Hwa Kim, Jaehyun Jun, Jung Woo Ha, and Byoung Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Neural Information Processing Systems (NeurIPS)*, pages 4653–4663, 2017.

Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative Models from the perspective of Continual Learning. *International Joint Conference on Neural Networks (IJCNN)*, 2019.

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, 1994.

Fayin Li and Harry Wechsler. Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1686–1697, 2005.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to Grow: A Continual Structure Learning Framework for Overcoming Catastrophic Forgetting. *International Conference on Machine Learning (ICML)*, 2019a.

Xin Li and Yuhong Guo. Adaptive active learning for image classification. *Computer Vision and Pattern Recognition (CVPR)*, pages 859–866, 2013.

Yingzhen Li, John Bradshaw, and Yash Sharma. Are Generative Classifiers More Robust to Adversarial Attacks? *International Conference on Machine Learning (ICML)*, 2019b.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *European Conference on Computer Vision (ECCV)*, 2016.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the Reliability of Out-of-distribution Image Detection in Neural Networks. *International Conference on Learning Representations (ICLR)*, 2018.

Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the Unexpected via Image Resynthesis. *International Conference on Computer Vision (ICCV)*, 2019.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. *Neural Information Processing Systems (NeurIPS)*, 2017.

David J. C. MacKay. Information-Based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604, 1992.

Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. In *European Conference on Computer Vision (ECCV)*, 2018.

Ofer Matan, Richard Kiang, C. E. Stenard, Bernhard E. Boser, John Denker, Don Henderson, W. Hubbard, Larry Jackel, and Yann LeCun. Handwritten Character Recognition Using Neural Network Architectures. *4th USPS Advanced Technology Conference*, 2(5):1003–1011, 1990.

Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. *International Conference on Machine Learning (ICML)*, pages 7744–7754, 2019.

Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

Andrew K. McCallum and Kamal Nigam. Employing EM and Pool-Based Active Learning for Text Classification. *International Conference on Machine Learning (ICML)*, 1998.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks : The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165, 1989.

Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sunderhauf. Dropout Sampling for Robust Object Detection in Open-Set Conditions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249, 2018.

Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

Martin Mundt, Sagnik Majumder, Iuliia Pliushch, Yong Won Hong, and Visvanathan Ramesh. Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition. *arXiv preprint arXiv:1905.12019*, 2019a.

Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers? *International Conference on Computer Vision (ICCV), First Workshop on Statistical Deep Learning for Computer Vision (SDL-CV)*, 2019b.

Roberto Munro. *Human-in-the-Loop Machine Learning*. Manning Publications, Manning Early Access Program, 2020.

Eric Nalisnick, Akihiro Matsukawa, Yee W. Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? *International Conference on Learning Representations (ICLR)*, 2019.

Ross Naylor. Known knowns, known unknowns and unknown unknowns: A 2010 update on carotid artery disease. *Surgeon*, 8(2):79–86, 2010.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational Continual Learning. *International Conference on Learning Representations (ICLR)*, 2018.

Hieu T. Nguyen and Arnold Smeulders. Active learning using pre-clustering. *International Conference on Machine Learning (ICML)*, 2004.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *Computer Vision and Pattern Recognition (CVPR)*, 2014. ISSN 10636919.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. *Neural Information Processing Systems (NeurIPS)*, 2019.

Poojan Oza and Vishal M. Patel. C2AE: Class Conditioned Auto-Encoder for Open-set Recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Sinno J. Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10), 2010.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113: 54–71, 2019.

Dongmin Park, Seokil Hong, Bohyung Han, and Kyoung Mu Lee. Continual Learning by Asymmetric Loss Approximation with Single-Side Overestimation. *International Conference on Computer Vision (ICCV)*, 2019.

Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: One-class Novelty Detection Using GANs with Constrained Latent Representations. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

B. Pfülb and A. Gepperth. A Comprehensive, Application-Oriented Study of Catastrophic Forgetting in DNNs. *International Conference on Learning Representations (ICLR)*, 2019.

Lorien Y. Pratt. Discriminability-Based Transfer between Neural Networks. *Neural Information Processing Systems (NeurIPS)*, 1993.

Lorien Y. Pratt, Jack Mostow, and Candance A. Kamm. Direct Transfer of Learned Information Among Neural Networks. *AAAI Conference on Artificial Intelligence (AAAI)*, 1991.

Amal Rannen, Rahaf Aljundi, Matthew B. Blaschko, and Tinne Tuytelaars. Encoder Based Lifelong Learning. *International Conference on Computer Vision (ICCV)*, 2017.

Roger Ratcliff. Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological Review*, 97(2):285–308, 1990.

Sylvestre A. Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Anthony Robins. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience Replay for Continual Learning. *Neural Information Processing Systems (NeurIPS)*, 2018.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 441–448, 2001.

Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv preprint arXiv: 1706.05098*, 2017.

Donald Rumsfeld. U.S. Department of Defense news briefing addressing unknown unknowns, 2002. URL `https://archive.defense.gov/Transcripts/Transcript.aspx?TranscriptID=2636`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*, 2016.

David Saad, editor. *On-line learning in neural networks*. Cambridge University Press, New York, NY, USA, 1999.

Doyen Sahoo, Quang Pham, Jing Lu, and Steven C. H. Hoi. Online deep learning: Learning deep neural networks on the fly. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2660–2666, 2018.

Robert E. Schapire. The Strength of Weak Learnability. *Machine Learning*, 5(2):197–227, 1990. ISSN 15730565.

Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7):1757–1772, 2013.

Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. Probability Models For Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *International Conference on Learning Representations (ICLR)*, 2018.

Joan Serra, Dídac Suris, Marius Mirón, and Alexandras Karatzoglou. Overcoming Catastrophic forgetting with hard attention to the task. *International Conference on Machine Learning (ICML)*, 2018.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1070–1079, 2008.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 287–294, 1992.

Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(4):623–656, 1948.

Hanul Shin, Jung K. Lee, Jaehong J. Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. *Neural Information Processing Systems (NeurIPS)*, 2017.

Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep Active Learning: Unified and Principled Method for Query and Training. *AISTATS*, 2020.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *International Conference on Computer Vision (ICCV)*, 2019.

Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. Towards Training Recurrent Neural Networks for Lifelong Learning. *Neural Computation*, 2019.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMRL)*, 15:1929–1958, 2014.

S. C. Suddarth and Y. L. Kergosien. Rule-injection hints as a means of improving network performance and learning time. *Neural Networks. EURASIP 1990. Lecture Notes in Computer Science*, 412, 1990.

Sebastian Thrun. Is Learning The n-th Thing Any Easier Than Learning The First? *Advances in Neural Information Processing Systems*, 1996a.

Sebastian Thrun. *Explanation-Based Neural Network Learning - A Lifelong Learning Approach*. Springer US, 1996b.

Simon Tong and Daphne Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research (JMLR)*, 2001.

Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. Bayesian generative active deep learning. *International Conference on Machine Learning (ICML)*, 2019.

Ivor W. Tsang, James T. Kwok, and Pak Ming Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

Ya. Z. Tsypkin. *Adaptation and Learning in Automatic Systems*. Academic Press, New York, 1971.

Krishna C. Vadodaria and Sebastian Jessberger. Functional neurogenesis in the adult hippocampus: Then and now. *Frontiers in Neuroscience*, 8(8 MAR):1–3, 2014.

Gido M. van de Ven and Andreas S. Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.

Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, Berlin, Heidelberg, 1982.

Yaqing Wang, Quanming Yao, James Kwok, and Lionel M. Ni. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Computing Surveys*, 2020.

Karl Weiss, Taghi M. Khoshgoftaar, and Ding Ding Wang. A survey of transfer learning. *Journal of Big Data*, 3(1), 2016.

Max Welling. Herding dynamical weights to learn. *International Conference On Machine Learning (ICML)*, pages 1121–1128, 2009.

Jason Weston, Ronan Collobert, Fabian Sinz, Léon Bottou, and Vladimir Vapnik. Inference with the Universum. *International Conference on Machine Learning (ICML)*, 2006.

Chenshen Wu, Luis Herranz, Xialei Liu, Yaxing Wang, Joost van de Weijer, and Bogdan Raducanu. Memory Replay GANs: learning to generate images from new categories without forgetting. *Neural Information Processing Systems (NeurIPS)*, 2018.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental Learning Using Conditional Adversarial Networks. *International Conference on Computer Vision (ICCV)*, 2019.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or Signal: The Role of Image Backgrounds in Object Recognition. *ArXiv preprint arXiv: 2006.09994*, 2020.

Ju Xu and Zhanxing Zhu. Reinforced continual learning. *Neural Information Processing Systems (NeurIPS)*, 2018.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung J. Hwang. Lifelong Learning with Dynamically Expandable Networks. *International Conference on Learning Representations (ICLR)*, 2018.

Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-Reconstruction Learning for Open-Set Recognition. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Neural Information Processing Systems (NeurIPS)*, 2014.

Qing Yu and Kiyoharu Aizawa. Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy. *International Conference on Computer Vision (ICCV)*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. *British Machine Vision Conference (BMVC)*, 2016.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. *International Conference on Machine Learning (ICML)*, 70:3987–3995, 2017.

Mengyao Zhai, Lei Chen, Fred Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong GAN: Continual Learning for Conditional Image Generation. *International Conference on Computer Vision (ICCV)*, 2019.

Tong Zhang and Frank J. Oles. A Probability Analysis on the Value of Unlabelled Data for Classification Problems. *International Conference on Machine Learning (ICML)*, 2000.

Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online Incremental Feature Learning with Denoising Autoencoders. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 22:1453–1461, 2012.

Jia-Jie Zhu and José Bento. Generative Adversarial Active Learning. *NeurIPS workshop on Teaching Machines, Robots, and Humans*, 2017.

Martin Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. *International Conference On Machine Learning (ICML)*, 2003.

Discussion and Outlook

Summary

The three main chapters of the thesis, comprised of six manuscripts and one section containing a qualitative application demonstration, have seen the introduction of multiple techniques that each in their own address various aspects of the machine learning workflow.

Broadly speaking, the first chapter has been focused on questions concerning selection of adequate representational capacity in a deep neural network, its hierarchy of individual operations and how to adapt an architecture over time for prospective continual learning. Three works have proposed a mechanism to repeatedly expand the amount of neural network features as required during training (Mundt et al., 2017), have empirically investigated common design rule of thumbs for the neural network topology with respect to its feature distribution across layers (Mundt et al., 2018b), and have explored neural architecture meta-learning in the context of a newly proposed dataset for the task of detecting and classifying defects in concrete bridges (Mundt et al., 2019a). From a perspective of a practical machine learning engineer, these works have called to attention that deep learning requires more than a simple "plug and play", that is taking a specific neural network architecture and feeding it with some collected data will not inherently provide the desired solution. Apart from raising awareness, the central contribution of the chapter has thus been to develop and showcase ways of how these architectures can better be selected or even constructed automatically given a specific task, with an application to real world concrete defect classification.

The second chapter has proceeded to investigate complimentary questions of how to continuously train a neural network when the underlying distribution of the training data shifts over time and correspondingly, how to detect data instances that are unlikely to originate from the same distribution as the observed data population. The first two works in this chapter have proposed a data distribution centered variant for extreme value theory based open set recognition in deep neural networks. They have highlighted the interdependence of sta-

tistical outlier recognition and the generative rehearsal of instances that are constrained to resemble previously observed data (Mundt et al., 2020b). They have investigated the role of model choice and deep uncertainty approximations for the recognition of unseen unknown data (Mundt et al., 2019b). A final section has qualitatively corroborated the application of the developed techniques to the earlier introduced concrete defect detection task, in order to provide an additional practical application context. In essence, the central contribution can be summarized in simplified terms. The works introduce and characterize a common approach to address the well-known issue of a deep neural network forgetting what it has previously learned when training is continued on very different concepts, and at the same time overcome the long-standing challenge of a deep neural network producing completely wrong predictions on unknown data. For the practical machine learning engineer, this implies that there no longer needs to be a drastic restriction of what a user of a neural network based machine learning system is allowed to show to the system. Examples that do not belong to the trained task or deviate substantially from the training data will simply be flagged as such, providing a strong indication for the user not to trust the system and set these examples aside for a different form of processing. For instance, if a neural network has been trained to distinguish dogs and cats, we no longer have to worry about the system assigning a dog or cat label to images of owls or horses, and instead will be notified that they belong to neither category. Once they are set aside, these images of owls and horses can then be used for further training of the neural network to add two additional categories to the classifier, without the necessity of continuously reminding it about dogs and cats.

The third and last chapter has concluded the thesis with a manuscript that provides the overall integrated perspective (Mundt et al., 2020a). The presented viewpoint is approached from a conducted broader literature review and linked to prior lessons which have seemingly been forgotten in the deep learning era. It has not only highlighted the synergies of a consolidated view, but also extensively argued about the threats and pitfalls in the absence of the latter. Based on the author's earlier manuscripts and their respective insights, a unified framework has been presented in order to condense questions regarding selection of data instances to include for future continuous training steps, the role and importance of task order, mitigation of catastrophic forgetting in continuous training and general robustness when training and deploying deep neural networks in an unconstrained real world beyond static benchmarks. This perspective has been empirically validated through multiple quantitative experiments in a framework's realization in deep generative neural networks. These experiments raise the machine learning practitioner's awareness about the required systems perspective in order to succeed beyond a particular type of benchmark, and have contributed one practical realization towards a symbiosis of separately treated machine learning aspects in a single deep neural network approach.

Conclusively, the proposed methods and obtained insights can all be regarded as advances towards more robust real-world applicability of deep neural networks. Even though there are still many components that require substantially more grounded understanding, such as the precise nature of the formed neural network representations in a multi-layer encoder or the respective cascade of non-linear transformations that translate learned generative factors into actual data instances, the developed techniques fill some of the current gaps in pure blackbox deep learning usage. As a consequence, various future directions are conceivable. Some of them reside on the rather immediate time scale, whereas other imaginable paths are likely to require significantly more time or the emergence of a disruptive breakthrough. In the subsequent final thesis sections some of these prospects are briefly sketched. At first, this includes natural and imminent extensions of the presented individual works and some of their remaining challenges that still need to be resolved. This is then followed by the author's personal conjectures on potential advancement in a grander frame of reference.

Short-term Prospects

Each of the works presented in the thesis comes with its respective short-term outlook. Here, these outlooks are first extended on a detailed level and then followed by the larger overarching long-term research questions and open challenges. The sub-sequent paragraphs thus roughly follow the outline of the thesis' chapters, revisit their individual works and suggest further interconnection and extensions.

The chronologically earliest introduced work of neural network capacity expansion (Mundt et al., 2017) is accompanied by two major open challenges. The first one is an empirical investigation into the stability and practical reliability of the proposed technique. The second one is the development or embedding into a grounded theory of the deep neural network learning trajectory. Whereas the latter remains largely open and requires a potential deep dive into the complicated high dimensional loss landscapes, an examination of the former aspect has already been initiated in a co-advised thesis of Wendland (2020). In the respective analysis it becomes apparent that the success of the neural network expansion technique is reliant on inherent stochastic gradient descent regularization mechanisms and a decoupling of layers. Recall that the expansion itself is initiated whenever all features in a layer are observed to change with respect to their random initialization. A safe use of the technique with individually treated layers is thus enabled by techniques which ensure a steady gradient's magnitude in backpropagation. The expansion technique is then not prone to limitless capacity increase as a result of experiencing undesired effects leading to gradient explosion or vanishing. Two corresponding techniques: batch normalization (Ioffe

and Szegedy, 2015) and layer specific weight initialization to preserve activation magnitudes across layers (He et al., 2015), have been identified as key requirements. Empirically, as long as one of them is present, capacity expansion seems to be bounded and converge. In the absence of both mechanisms the capacity expansion technique is observed to be unstable and continue infinitely, which is simply due to lack of activation or gradient normalization yielding erratic weight changes. This is demonstrated by removing activation normalization according to batch statistics and employing weight initialization that does not normalize the rectified linear unit activations, such as the initialization previously proposed by Glorot and Bengio (2010). In consequence of this observation, the investigation is then extended to the question of online capacity addition. The previously assumed requirement of complete re-initialization of weights after an expansion step is thus lifted. Empirically, this seems to be possible without any major drawbacks, although further investigation on how the newly added capacity needs to formally be initialized remains open, i.e. the weight distribution can in principle drift during training with respect to the initial distribution from which it was drawn. The original work's computational limitation of weight re-initialization after expansion steps thus seems to be surmountable. One emerging central observation is that an online procedure yields multiple different types of architectures with the same final accuracy, even on rather simple and controlled tasks such as separating samples from multiple uni-modal Gaussian distributions with clearly distinct means. This opens up prospective future theoretical analysis with respect to the lottery ticket hypothesis (Frankle and Carbin, 2019), which articulates that the training effectiveness is similar to a lottery. Even if the same distribution is chosen for weight initialization, the empirical effectiveness with respect to training speed and amount of required parameters seems to be heavily reliant on the exact combination of initialized weights. It is believed that such an effect is observed in the investigated capacity expansion technique, where multiple points of convergence can be seen with similar accuracy, yet consistently differing amount of overall parameters. This further underlines the necessity of tying future work to recently developed theoretical advances.

Another natural short-term extension towards fully dynamic neural architectures is the combination of the presented capacity expansion technique with the other investigated architecture search methods (Mundt et al., 2019a). On the one hand, dynamic capacity expansion of each individual layer can dramatically reduce the search time and search space complexity of current architecture search techniques, which treat each combination of specific layer choice and its exact parameter amount as a unique state. On the other hand, architecture search allows to find a suitable hierarchy of operators, which can lift the current limitation of dynamic capacity expansion techniques to operate solely on pre-determined architecture depth. In this context, the perhaps most promising immediate future direction would be

the application of either dynamic architecture search technique to generative models. In the later works of the thesis (Mundt et al., 2019b, 2020b,a), the necessity for generative deep modelling has clearly emerged. Even if a fully supervised task is desired, knowledge of the training distribution seems to be a core component in identification of the open set and continual learning rehearsal. However, straightforward application of presently known reinforcement learning architecture search techniques to deep generative models is infeasible. Independently of whether a generative adversarial network or a variational autoencoder based model is considered, this is due to the interplay of multiple competing loss terms rendering the formulation of a simple reward signal problematic. As an example, in the explored variational Bayesian approaches a classification accuracy or reconstruction loss is typically regularized by a Kullback-Leibler divergence that encourages the encoded approximate posterior to follow a prior distribution. In a simple attempt to combine the reward through addition of individual loss terms, the architecture search procedure is bound to fail as it can rapidly decrease only one term at the expense of the other. Given the number of parameters of a deep neural network encoder, it is almost trivial to transform complex input into the often used unit Gaussian prior at the cost of maintaining adequate structure of the data's correlation. Conversely, it is equally trivial to rapidly minimize a reconstruction loss by simply creating a pure look-up table at the cost of deviation from the prior distribution. At present, the lack of detailed understanding of the often occurring training instabilities and the desired balance between this rate-distortion-perception tradeoff (Blau and Michaeli, 2019) need to be overcome first. In the co-advised thesis of Majumder (2018), reinforcement learning based architecture search for variational generative models has been investigated through attempts of loss modifications, alas with limited success. In today's literature, this problem is thus generally circumvented by basing the reward on a second model that judges the generated architecture through a supervised proxy objective, i.e. scoring the generated images according to a classifier (Gong et al., 2019), or by refraining from reinforcement learning and adopting techniques from evolutionary computing (Hajewski and Oliveira, 2020).

From a complementary perspective, the above challenges and future possibilities can be significantly strengthened by explicitly incorporating the specific observed data instances into the conducted analysis. In the last work of the thesis (Mundt et al., 2020a), the importance of individual data instances and their ordered curriculum beyond random sampling from the entire population is exemplified. Based on insights from recent works (Hacohen and Weinshall, 2019; Hacohen et al., 2020), it is further speculated that a natural curriculum is inherent to any stochastic gradient descent learning process. Even in a single task across multiple different neural network architectures and fully randomized sampling, the authors reckon that individual data instances are learned at a similar point in time of the

training trajectory. It is imaginable that existence of such an inherent curriculum could be correlated with traditional computer vision and human image difficulty metrics (Spain and Perona, 2008; Pinto et al., 2008; Liu et al., 2011; Hoiem et al., 2012; Russakovsky et al., 2013; Isola et al., 2014; Ionescu et al., 2016), following the intuition that easy to describe data is learned first. This could not only be further analysed with respect to the weight composition, e.g. Li et al. (2016); Wang et al. (2018) who observed that different neural networks also converge to similar representations, but also with respect to information theoretic principles of analysis. A prominent example for the latter could be the information flow throughout the training process and the respective theory on evolving information bottlenecks and their manifestation in periods of learning in neural networks (Tishby and Zaslavsky, 2015; Schwartz-Ziv and Tishby, 2017; Alemi et al., 2017; Saxe et al., 2018; Achille et al., 2019). Such a theory hypothesizes the existence of multiple distinct training phases that first attempt an initial data fit, followed by subsequent compression, before concluding the stochastic gradient descent optimization with diffusion like behavior. It is then naturally tied to questions about the architecture design, such as "do deep neural networks learn shallow learnable examples first?" (Mangalam and Prabhu, 2019) or the question whether residual deep neural networks behave like ensembles of more shallow architectures (Veit et al., 2016). Although generally difficult to corroborate and analyse in practice due to the lacking control of precise dataset composition, 3-D graphics simulators could serve as the required tool to nevertheless investigate above hypotheses.

Lastly, an essential question is how to transfer the developed architecture adaptation, open set and continual learning capabilities to scenarios with overall less human supervision. The corresponding thesis models (Mundt et al., 2019b, 2020b,a) are in principle already based on unsupervised learning techniques, however still require semi-supervision in providing class or dataset labels to form and identify clusters in the generative model's latent space. Plenty of recent works attempt to alleviate such supervision requirements and propose techniques for so called unsupervised disentanglement of generative factors. Works try to distinguish individual generative factors through decomposition and modification of the loss function (Higgins et al., 2017; Burgess et al., 2017; Mathieu et al., 2018) or by encoding the factors into altogether separate dimensions (Achille et al., 2018). However, the amount of required generative factors remains unclear, the amount of clusters that manifest in a suitable latent space without external supervision for an arbitrary dataset is similarly difficult to determine, and the question how these disentangled factor are ultimately associated with the way a human would address a task remains open. In the personal author's view, most of these questions cannot be adequately addressed without also regarding subsequent human interaction with the machine learning system. A conjecture of the open challenges of the latter is provided in the next and final thesis section.

Long-term Open Challenges

Pursuing most of the above formulated prospects on dynamic architectures, pinpointing failure modes of deep neural networks in robust application and continual adaptation, disentangled representations, or unsupervised learning, is unavoidably linked to a higher-level viewpoint and questions revolving around our desiderata as humans deploying machine learning systems. Conclusively, and in summary of the thesis' main message, it is not only a question of how a specific deep learning algorithm can be advanced to improve run-times, memory requirements or prediction accuracies. In contrast, depending on the precise applications, their potential safety or decision explainability prerequisites, the wanted human machine interaction, a variety of increasingly philosophical questions become progressively more relevant. To illustrate this from a perspective of the work in this thesis, let us revisit the final proposed integrated workflow (Mundt et al., 2020a). Whereas the necessity to combine architecture modification with active data queries, equip the system with open set recognition and continual learning capability, consider the importance of the training dataset composition and presented order is made clear in one imaginable natural deep framework, such a system is now subject to a variety of mechanism choices. Inasmuch as the explicit statistical choices and proposed mechanisms of this thesis certainly present one imaginable deep neural network system that advances the current blackbox state of the art, a variety of other techniques towards the same system goal might also be sensible. The underlying choices result in a myriad of conceivable combinations for a system's assembly and an ensuing multitude of required system comparisons beyond the validation of individual components in isolation on singled-out benchmarks. This is difficult because it requires extensive effort and is thus tremendously challenging to compare on a purely empirical basis. On the one hand, it is impractical to construct many large scale systems that rely on various combinations of suitable mechanisms. On the other hand, it is not necessarily sufficiently scientifically conclusive to propose a working system, beyond an observation that the engineering effort yields a system that currently has no competitor. It can thus be challenging to predict whether the built system pursues a generally promising direction. This highlights the necessity of both increasing our understanding of the deep neural network internals from a grounded theoretical perspective in addition to the currently dominating engineering efforts, and the inclusion of a more general debate on fundamental machine learning goals.

At the same time, our scientific community is currently facing the grand challenge of being confronted with an every growing plethora of parallel research threads. As outlined multiple times throughout the thesis, often these threads are disconnected from each other:

continual learning is separated from recognition of unknown data instances, neural architectures are viewed as almost agnostic to the specifically pursued task, data is treated from a perspective of fixed benchmark sets. Some of these threads continue to pursue the direction of constantly increasing accuracies with growing dataset sizes, which has been empirically demonstrated to scale (un)reasonably well with ever growing representational capacity (Sun et al., 2017) or in empirical observations of a double-descent phenomenon in deep learning, where the conventional wisdom of early stopping and the assumed bias variance trade-off (Nakkiran et al., 2020) no longer seems to apply. A central focus could thus be to find the precise source of deep neural networks' power on large datasets, or how to leverage deep neural networks even when small amounts of (labelled) data are accessible. Alternatively, works that try to decompose neural networks and impose disentanglement on their representations can be advanced (Higgins et al., 2017; Burgess et al., 2017; Mathieu et al., 2018; Achille et al., 2018). Plenty of other concurrent threads certainly exist. In either way, their coexistence highlights that it is time to go beyond simple benchmarking, i.e. building ever greater datasets and models for an abundant amount of different tasks, but to consider the entire systems perspective in connection with desiderata outside of contrived test set performance. Such a systems perspective could be explored through a symbiosis of traditional modelling techniques and modern data-driven learning systems. If we were to e.g. lift the constraint of hierarchically distributed and entangled representations in deep neural networks without loss in representational complexity, many of the current challenges such as catastrophic forgetting would be inherently simpler to solve.

In the end, however, we need to ask ourselves the question of what kind of advances to our body of scientific knowledge we desire, how we as humans can explain what we have crafted, why it fits our needs or whether we even need to explain every application in detail. In deep neural networks it is often said that knowledge is transferred in continual or transfer learning. Here, the word knowledge is tantamount to the learned representations. As such, in deep catastrophic interference it is generally stated that knowledge is forgotten because parameters are overwritten. We rarely seem to account for other forms of knowledge however: the question of how to include effective priors into deep learning, how our modelling assumptions are transferred and propagated through time, how quickly we can produce or reproduce a solution, how our design process evolves and our cultural knowledge is incorporated. We similarly seldomly seem to ask ourselves whether the "interpretability" or "disentanglement" we so greatly seem to seek is what is actually desirable in the practical world. Is a model consisting of purely linear operations interpretable (Lou et al., 2013)? And if so, should we switch our efforts to mathematically more intuitive generative flow models that impose functional invertability on every operation in a deep cascade at the cost of massive compute (see Kobyzev et al. (2020) for a review of deep flow models)? Are we

satisfied with deep hybrid models, that are in parts composed of traditional computer vision mechanisms and allow the inclusion of controlled elements at some levels of the hierarchy (Nushi and Horvitz, 2018; Shanahan et al., 2020; Gould et al., 2019)? We seem to hardly ever question the datasets we compose. How much of deep learning's success and failure can be attributed to our own human and social biases in the constructed benchmarks and the resulting undesired side effects? Traditional modelling of course suffers from similar challenges, alas it can be more clear whether a specific bias has been included or when a particular assumption is violated.

Ultimately, it may be that we have to balance the questions of "what can deep learning do for us?" and "what can we do for deep learning?" in practice. This is to say, we shouldn't pursue deep learning in an application for the sake of applying deep learning, and on the contrary, we should not immediately dismiss deep learning because certain aspects are not sufficiently explored yet. In this thesis, deep neural networks have been leveraged because of their immense representational power, yet advances have been made and challenges surpassed to partially lift their blackbox nature with respect to architecture design and using statistical modelling tools to enable continual learning and robust application. As is so often the case, the key between purely data-driven power and rigorous modelling lies in an appropriate balance. A balance which can only be achieved by constantly reminding oneself of the overarching systems perspective, even if we can only progress through the exploration of parts one step at a time. It is the author's sincere hope that such a systems perspective will enjoy an increased amount of adoption in future deep learning research.

Bibliography

Alessandro Achille, Tom Eccles, Loic Matthey, Christopher P. Burgess, Nick Watters, Alexander Lerchner, and Irina Higgins. Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies. *Neural Information Processing Systems (NeurIPS)*, 2018.

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical Learning Periods in Deep Networks. *International Conference on Learning Representations (ICLR)*, 2019.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep Variational Information Bottleneck. *International Conference on Learning Representations (ICLR)*, 2017.

Jose M. Alvarez and Mathieu Salzmann. Learning the Number of Neurons in Deep Networks. *Neural Information Processing Systems (NeurIPS)*, 2016.

Vincent Andrearczyk and Paul F. Whelan. Using filter banks in Convolutional Neural Networks for texture classification. *Pattern Recognition Letters*, 84:63–69, 2016.

Timur Ash. Dynamic Node Creation in Backpropagation Networks. *Connection Science*, 1(4):365–375, 1989.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences of the United States of America*, 113(27):7353–7360, 2016.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research (JMLR)*, 20:1–25, 2019.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Analyzing Classifiers: Fisher Vectors and Deep Neural Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing Neural Network Architectures using Reinforcement Learning. *International Conference on Learning Representations (ICLR)*, 2016.

Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences of the United States of America*, 110(45):18327–18332, 2013.

Matthias Bauer and Andriy Mnih. Resampled Priors for Variational Autoencoders. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 89, 2019.

Soeren Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. *arXiv preprint arXiv: 1807.03418*, 2018.

Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the Materials in Context Database. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Abhijit Bendale and Terrance E. Boult. Towards Open World Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Abhijit Bendale and Terrance E. Boult. Towards Open Set Deep Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17 (1), 1985.

Thomas O. Binford and Tod S. Levitt. Quasi-invariants: theory and exploitation. In *Proceedings of DARPA Image Understanding Workshop*, pages 819–829, 1993.

Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. *International Conference on Machine Learning (ICML)*, 2019.

Terrance E. Boult, Steve Cruz, Akshay R. Dhamija, Manuel Gunther, James Henrydoss, and Walter J. Scheirer. Learning and the Unknown : Surveying Steps Toward Open World Recognition. *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Joan Bruna and Stephane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1872–1886, 2013.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-VAE. *Neural Information Processing Systems (NeurIPS), Workshop on Learning Disentangled Representations*, 2017.

Charles F. Cadieu, Ha Hong, Daniel L.K. Yamins, Nicolas Pinto, Diego Ardila, Ethan A. Solomon, Najib J. Majaj, and James J. DiCarlo. Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *PLoS Computational Biology*, 10(12), 2014.

Mao Sen Cao and Qing Wen Ren. Fractal Behavior of Concrete Crack and Its Application to Damage Assessment. *Key Engineering Materials*, 312:325–332, 2006.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational Lossy Autoencoder. *International Conference on Learning Representations (ICLR)*, 2017.

Roland T. Chin and Charles R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1), 1986.

Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep convolutional filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep Learning for Classical Japanese Literature. *Neural Information Processing Systems (NeurIPS), Workshop on Machine Learning for Creativity and Design*, 2018.

Wilson R. L. da Silva and Diogo S. de Lucena. Concrete Cracks Detection Based on Deep Learning Image Classification. *International Conference on Experimental Mechanics (ICEM)*, 2018.

Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, and Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34, 1999.

DARPA and John Launchbury. A DARPA Perspective on Artificial Intelligence, 2017. URL `https://www.darpa.mil/about-us/darpa-perspective-on-ai`.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *arXiv preprint arXiv: 1909.08383*, 2019.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, Université de Montréal, 2009.

Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2014.

Alireza Farhidzadeh, Eehsan Dehghan-Niri, Mohamed A. Moustafa, Salvatore Salamone, and Andrew Whittaker. Damage Assessment of Reinforced Concrete Structures Using Fractal Analysis of Residual Crack Patterns. *Experimental Mechanics*, 53(9):1607–1619, 2013.

Sebastian Farquhar and Yarin Gal. Towards Robust Evaluations of Continual Learning. *International Conference on Machine Learning (ICML), Lifelong Learning: A Reinforcement Learning Approach Workshop*, 2018.

Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *International Conference on Learning Representations (ICLR)*, 2019.

Robert M. French. Semi-distributed Representations and Catastrophic Forgetting in Connectionist Networks. *Connection Science*, 4(3-4):365–377, 1992.

Brian V. Funt and Graham D. Finlayson. Color Constant Color Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(5):522–529, 1995.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation : Representing Model Uncertainty in Deep Learning. *International Conference on Machine Learning (ICML)*, 48, 2015.

Robert Geirhos, Claudio Michaelis, Felix A. Wichmann, Patricia Rubisch, Matthias Bethge, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations (ICLR)*, 2019.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 9:249–256, 2010.

Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. AutoGAN: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3223–3233, 2019.

Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout Networks. *International Conference on Machine Learning (ICML)*, 28(3):1319–1327, 2013.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Neural Information Processing Systems (NeurIPS)*, 2014.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Google Cloud. Google Cloud guides: ML solutions overview, the ML workflow, 2020. URL `https://cloud.google.com/ai-platform/docs/ml-solutions-overview`.

Stephen Gould, Richard Hartley, and Dylan Campbell. Deep Declarative Networks : A New Hope. *arXiv preprint arXiv: 1909.04866*, 2019.

Charles G. Gross. Neurogenesis in the adult brain: Death of a dogma. *Nature Reviews Neuroscience*, 1(1):67–73, 2000.

Ishaan Gulrajani, Kundan Kumar, Ahmed Faruk, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. PixelVAE: a Latent Variable Model for Natural Images. *International Conference on Learning Representations (ICLR)*, 2017.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. *International Conference on Machine Learning (ICML)*, 2019.

Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's Agree to Agree: Neural Networks Share Classification Order on Real Datasets. *International Conference on Learning Representations (ICLR)*, 2020.

Jeff Hajewski and Suely Oliveira. An Evolutionary Approach to Variational Autoencoders. *Annual Computing and Communication Workshop and Conference (CCWC)*, pages 71–77, 2020.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both Weights and Connections for Efficient Neural Networks. *Neural Information Processing Systems (NeurIPS)*, 2015.

Song Han, Huizi Mao, and William J. Dally. Deep Compression - Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *International Conference on Learning Representations (ICLR)*, 2016.

Song Han, Huizi Mao, Enhao Gong, Shijian Tang, William J. Dally, Jeff Pool, John Tran, Bryan Catanzaro, Sharan Narang, Erich Elsen, Peter Vajda, and Manohar Paluri. DSD: Dense-Sparse-Dense Training For Deep Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Li Hao, Asim Kadav, Hanan Samet, Igor Durdanovic, and Hans Peter Graf. Pruning Filfters For Efficient Convnets. In *International Conference on Learning Representations (ICLR)*, 2017.

Robert M. Haralick. Performance Characterization in Computer Vision. *British Machine Vision Conference (BMVC)*, 1992.

Eric Hayman, Barbara Caputo, Mario Fritz, and Jan-Olof Eklundh. On the Significance of Real-World Conditions for Material Classification. In *European Conference on Computer Vision (ECCV)*, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer Vision (ECCV)*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *International Conference on Computer Vision (ICCV)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *International Conference on Learning Representations (ICLR)*, 2019.

Timm Hess. *Training Convolutional Neural Networks on Virtual Examples for Object Classification in the Robocup-Environment*. Bachelor thesis, Goethe University, Frankfurt am Main, 2016.

Timm Hess*, Martin Mundt*, Tobias Weis, and Visvanathan Ramesh (* equal contribution). Large-scale stochastic scene generation and semantic annotation for deep convolutional neural network training in the robocup SPL. *Lecture Notes in Computer Science (LNAI), RoboCup 2017: Robot World Cup XXI.*, 11175:33–44, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *International Conference on Learning Representations (ICLR)*, 2017.

Geoffrey E. Hinton, Christopher K. I. Williams, and Michael Revow. Adaptive Elastic Models for Hand-Printed Character Recognition. *Neural Information Processing Systems (NeurIPS)*, 1992.

Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *Neural Information Processing Systems (NeurIPS), Deep Learning Workshop*, 2014.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Matthew D. Hoffman and Matthew J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. *Neural Information Processing Systems (NeurIPS), Advances in Approximate Bayesian Inference Workshop*, 2016.

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. *European Conference on Computer Vision (ECCV)*, 2012.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Huaibo Huang, Zhihang Li, Ran He, Zhenan Sun, and Tieniu Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. *Neural Information Processing Systems (NeurIPS)*, 2018.

David H. Hubel and Torsten N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160:106–54, 1962.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial Examples are not Bugs, they are Features. *Neural Information Processing Systems (NeurIPS)*, 2019.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning (ICML)*, 2015.

Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. How hard can it be? Estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What makes a photograph memorable? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7): 1469–1482, 2014.

Achref Jaziri. *A deep learning approach for semantic segmentation of concrete material cracks from virtually generated images*. Bachelor thesis, Goethe University, Frankfurt am Main, 2020.

Guoliang Kang, Jun Li, and Dacheng Tao. Shakeout: A New Regularized Deep Neural Network Training Scheme. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1751–1757, 2016.

Osman Semih Kayhan and Jan C. van Gemert. On Translation Invariance in CNNs: Convolutional Layers Can Exploit Absolute Spatial Location. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring Catastrophic Forgetting in Neural Networks. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Hyunjun Kim, Eunjong Ahn, Myoungsu Shin, and Sung-Han Sim. Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning. *Structural Health Monitoring*, 2018.

Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations (ICLR)*, 2013.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-Supervised Learning with Deep Generative Models. *Neural Information Processing Systems (NeurIPS)*, 2014.

Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing Flows: An Introduction and Review of Current Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

Christian Koch, Kristina Georgieva, Varun Kasireddy, Burcu Akinci, and Paul Fieguth. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, 29(2):196–210, 2015.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems (NeurIPS)*, 2012.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building Machines that learn and think like people. *Behav Brain Sci.*, 40(253), 2017.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), 2019.

Anders Boesen Lindbo Larsen, Soren Kaae Sonderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *International Conference on Machine Learning (ICML)*, 2016.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu. Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 464–472, 2015.

Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. *International Conference on Machine Learning (ICML)*, 2016.

Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative Models from the perspective of Continual Learning. *International Joint Conference on Neural Networks (IJCNN)*, 2019.

Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision (IJCV)*, 43(1):29–44, 2001.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent Learning: Do different neural networks learn the same representations? *International Conference on Learning Representations (ICLR)*, 2016.

Yundong Li, Hongguang Li, and Hongren Wang. Pixel-wise crack detection using deep local pattern predictor for robot application. *Sensors*, 18(9), 2018.

Min Lin, Chen Qiang, and Yan Shuicheng. Network In Network. *International Conference on Learning Representations (ICLR)*, 2014.

Zachary C. Lipton. The Mythos of Model Interpretability. *International Conference on Machine Learning (ICML), Workshop on Human Interpretability in Machine Learning*, 2016.

Dingding Liu, Yingen Xiong, Kari Pulli, and Linda Shapiro. Estimating image segmentation difficulty. *Machine Learning and Data Mining in Pattern Recognition*, 6871 LNAI:484–495, 2011.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent With Warm Restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 623–631, 2013.

Ali Maaruf, Michael A. Gennert, and Trevor G Clarkson. Analysis, Generation and Compression of Pavement Distress Images Using Fractals. In A. J. Crilly, R. A. Earnshaw, and H. Jones, editors, *Applications of Fractals and Chaos*, pages 147–169. Springer-Verlag Berlin and Heidelberg GmbH & Co., 1993.

Sagnik Majumder. *Neural Architecture Meta-learning via Reinforcement*. Bachelor thesis, Birla Institute of Technology and Science, Pilani, 2018.

Stéphane Mallat. Group Invariant Scattering. *Communications on Pure and Applied Mathematics*, 65 (10):1331–1398, 2012.

Karttikeya Mangalam and Vinay Prabhu. Do deep neural networks learn shallow learnable examples first? *International Conference on Machine Learning (ICML), Deep Phenomena Workshop*, 2019.

Gary Marcus. Deep Learning: A Critical Appraisal. *arXiv preprint arXiv: 1801.00631*, 2018.

Gary Marcus and Yoshua Bengio. AI Debate: The Best Way Forward for AI, 2019. URL `https://montrealartificialintelligence.com/aidebate/`.

Gary Marcus and Danny Lange. AI: The Next Chapter, Featured Conference Session Debate, 2020. URL https://emtech.technologyreview.com/emtech-digital-2020/home.

Ofer Matan, Richard Kiang, C. E. Stenard, Bernhard E. Boser, John Denker, Don Henderson, W. Hubbard, Larry Jackel, and Yann LeCun. Handwritten Character Recognition Using Neural Network Architectures. *4th USPS Advanced Technology Conference*, 2(5):1003–1011, 1990.

Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling Disentangling. *Neural Information Processing Systems (NeurIPS), Workshop on Bayesian Deep Learning*, 2018.

Emile Mathieu, Tom Rainforth, N. Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. *International Conference on Machine Learning (ICML)*, pages 7744–7754, 2019.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks : The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165, 1989.

Grégoire Montavon, Wojciech Samek, and Klaus Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal*, 73:1–15, 2018.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric Invariant Theory*. Springer-Verlag Berlin Heidelberg, 1994.

Martin Mundt, Tobias Weis, Kishore Konda, and Visvanathan Ramesh. Building effective deep neural network architectures one feature at a time. *arXiv preprint arXiv: 1705.06778*, 2017.

Martin Mundt, Sagnik Majumder, and Visvanathan Ramesh. AEROBI - D3.6 Deliverable: Online Learning, European Union's Horizon 2020 research programme under grant agreement No. 687384. Technical report, Frankfurt Institute for Advanced Studies, Goethe University, Frankfurt am Main, 2018a.

Martin Mundt, Sagnik Majumder, Tobias Weis, and Visvanathan Ramesh. Rethinking Layer-wise Feature Amounts in Convolutional Neural Network Architectures. *Neural Information Processing Systems (NeurIPS), Critiquing and Correcting Trends in Machine Learning Workshop*, 2018b.

Martin Mundt, Sreenivas Murali, Andres Fernandes, Iuliia Pliushch, Sumit Pai, and Visvanathan Ramesh. AEROBI - D3.3 Deliverable: Cognitive Vision System V2, European Union's Horizon 2020 research programme under grant agreement No. 687384. Technical report, Frankfurt Institute for Advanced Studies, Goethe Unviersity, Frankfurt am Main, 2018c.

Martin Mundt, Sreenivas Murali, and Visvanathan Ramesh. AEROBI - D3.1 and D3.2 Deliverable: Cognitive Vision System V1, Online Learning. Technical report, Frankfurt Institute for Advanced Studies, Goethe Unviersity, Frankfurt am Main, 2018d.

Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019a.

Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. Open Set Recognition Through Deep Neural Network Uncertainty: Does Out-of-Distribution Detection Require Generative Classifiers? In *Proceedings of the IEEE Computer Society International Conference on Computer Vision (ICCV), First Workshop on Statistical Deep Learning for Computer Vision (SDL-CV)*, 2019b.

Martin Mundt, Yong Won Hong, Iuliia Pliushch, and Visvanathan Ramesh. A Wholistic View of Continual Learning with Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning. *arXiv preprint arXiv:2009.01797*, 2020a.

Martin Mundt, Sagnik Majumder, Iuliia Pliushch, Yong Won Hong, and Visvanathan Ramesh. Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition. *arXiv preprint arXiv:1905.12019*, 2020b.

Joseph L. Mundy and Andrew Zisserman. *Geometric invariance in computer vision*. MIT Press, US, Cambridge, MA, 1992.

Kenji Nagao and W. Eric L. Grimson. Using Photometric Invariants for 3D Object Recognition. *Computer Vision and Image Understanding*, 71(1):74–93, 1998.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt. *International Conference on Learning Representations (ICLR)*, 2020.

Eric Nalisnick, Akihiro Matsukawa, Yee W. Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? *International Conference on Learning Representations (ICLR)*, 2019.

Srinivasa G. Narasimhan, Visvanathan Ramesh, and Shree K. Nayar. A class of photometric invariants: Separating material from shape and illumination. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1387–1394, 2003.

Shree K. Nayar and Ruud M. Bolle. Reflectance based object recognition. *International Journal of Computer Vision*, 17(3):219–240, 1996.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. *Neural Information Processing Systems (NeurIPS), Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Maria Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification, 2006.

Besmira Nushi and Eric Horvitz. Towards Accountable AI : Hybrid Human-Machine Analyses for Characterizing System Failure. *AAAI Conference on Human Computation and Crowdsourcing*, 2018.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The Building Blocks of Interpretability. *Distill*, 2018.

Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep Learning vs. Traditional Computer Vision. *Advances in Intelligent Systems and Computing*, 943:128–144, 2019.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. *Neural Information Processing Systems (NeurIPS)*, 2019.

Sinno J. Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10), 2010.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113:54–71, 2019.

Dragutin Petkovic. The Need for Accuracy Verification of Machine Vision Algorithms and Systems. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1989.

Benedikt Pfülb and Alexander Gepperth. A Comprehensive, Application-Oriented Study of Catastrophic Forgetting in DNNs. *International Conference on Learning Representations (ICLR)*, 2019.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient Neural Architecture Search via Parameters Sharing. *International Conference on Machine Learning (ICML)*, 2018.

Nicolas Pinto, David D. Cox, and James J. DiCarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1):151–156, 2008.

Visvanathan Ramesh and Robert M. Haralick. Random perturbation models and performance characterization in computer vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 521–527, 1992a.

Visvanathan Ramesh and Robert M. Haralick. Performance characterization of edge detectors. *Applications of Artificial Intelligence X: Machine Vision and Robotics*, 1708, 1992b.

Roger Ratcliff. Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological Review*, 97(2):285–308, 1990.

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Quoc Le, and Alex Kurakin. Large-Scale Evolution of Image Classifiers. In *International Conference on Machine Learning (ICML)*, 2017.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(6):1137–1149, 2017.

Greg Ridgeway, David Madigan, Thomas Richardson, and John O'Kane. Interpretable Boosted Naïve Bayes Classification. *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 101–104, 1998.

Anthony Robins. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995.

Pau Rodriguez, Jordi González, Guillem Cucurull, Josep M. Gonfaus, and Xavier Roca. Regularizing CNNs With Locally Constrained Decorrelations. In *International Conference on Learning Representations (ICLR)*, 2017.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Olga Russakovsky, Jia Deng, Zhiheng Huang, Alexander C. Berg, and Li Fei-Fei. Detecting avocados to Zucchinis: What have we done, and where are we going? *International Conference on Computer Vision (ICCV)*, 2013.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015.

Andrew M. Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D. Tracey, and David D. Cox. On the Information Bottleneck Theory of Deep Learning. *International Conference on Learning Representations (ICLR)*, 2018.

Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Towards Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7):1757–1772, 2013.

Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. Probability Models For Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014.

Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(5):530–535, 1997.

Ravid Schwartz-Ziv and Naftali Tishby. Opening the black box of Deep Neural Networks via Information. *arXiv preprint arXiv: 1703.00810*, 2017.

Murray Shanahan, Kyriacos Nikiforou, Antonia Creswell, Christos Kaplanis, David Barrett, and Marta Garnelo. An Explicitly Relational Neural Network Architecture. *International Conference on Machine Learning (ICML)*, 2020.

Lavanya Sharan, Ruth Rosenholtz, and Edward H. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision (JOV)*, 9(8), 2009.

Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12): 3434–3445, 2016.

Hanul Shin, Jung K. Lee, Jaehong J. Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. *Neural Information Processing Systems (NeurIPS)*, 2017.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not Just a Black Box: Interpretable Deep Learning by Propagating Activation Differences. *International Conference on Machine Learning (ICML)*, 2016.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations (ICLR) - Workshop Track*, 2014.

Merrielle Spain and Pietro Perona. Some objects are more equal than others: measuring and predicting importance. *European Conference on Computer Vision (ECCV)*, 2008.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net. *International Conference on Learning Representations (ICLR)*, 2015.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMRL)*, 15:1929–1958, 2014.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems (NeurIPS)*, 1999.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014.

Neil A. Thacker, Adrian F. Clark, John L. Barron, J. Ross Beveridge, Patrick Courtney, William R. Crum, Visvanathan Ramesh, and Christine Clark. Performance characterization in computer vision: A guide to best practices. *Computer Vision and Image Understanding*, 109(3):305–334, 2008.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop (ITW)*, 2015.

Jakub M. Tomczak and Max Welling. VAE with a vampprior. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 84, 2018.

Yanghai Tsin, Robert T Collins, and Visvanathan Ramesh. Bayesian Color Constancy for Outdoor Object Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

John Tsotsos, Iuliia Kotseruba, Alexander Andreopoulos, and Yulong Wu. Why does data-driven beat theory-driven computer vision? *International Conference on Computer Vision Workshops (ICCVW)*, 2019.

Jasper R. R. Uijlings, Koen E. A. Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision (IJCV)*, 104:154–171, 2013.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. It Takes (Only) Two : Adversarial Generator-Encoder Networks. *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Krishna C. Vadodaria and Sebastian Jessberger. Functional neurogenesis in the adult hippocampus: Then and now. *Frontiers in Neuroscience*, 8(8 MAR):1–3, 2014.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. *International Conference on Machine Learning (ICML)*, 48:1747–1756, 2016.

Andreas Veit, Michael Wilber, and Serge Belongie. Residual Networks Behave Like Ensembles of Relatively Shallow Networks. *Neural Information Processing Systems (NeurIPS)*, 2016.

Liwei Wang, Lunjia Hu, Jiayuan Gu, Yue Wu, Zhiqiang Hu, Kun He, and John Hopcroft. Towards understanding learning representations: To what extent do different neural networks learn the same representation. *Neural Information Processing Systems (NeurIPS)*, 2018.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

Tobias Weis, Martin Mundt, Patrick Harding, and Visvanathan Ramesh. Anomaly detection for automotive visual signal transition estimation. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems Proceedings (ITSC)*, 2018.

Karl Weiss, Taghi M. Khoshgoftaar, and Ding Ding Wang. A survey of transfer learning. *Journal of Big Data*, 3(1), 2016.

Hannah Wendland. *Feature-Wise Expansion of Neural Network Architectures*. Bachelor thesis, Goethe University, Frankfurt am Main, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv: 1708.07747*, 2017.

Liang Yang, Bing Li, Wei Li, Zhaoming Liu, Guoyong Yang, and Jizhong Xiao. Deep Concrete Inspection Using Unmanned Aerial Vehicle Towards CSSC Database. In *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Neural Information Processing Systems (NeurIPS)*, pages 3320–3328, 2014.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. *British Machine Vision Conference (BMVC)*, 2016.

Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Acknowledgements

I would like to extend my deepest gratitude to my supervisor Prof. Dr. Visvanathan Ramesh. Apart from providing consistent support and guidance on research ideas and applied projects, he has repeatedly surpassed the necessary requirements of an advisor. He has continuously challenged me to question my work, in the most positive of ways. Without his constant encouragement I would not have been able to expand my comfort zone and push the boundaries of my research. I particularly thank him for convincing me to pursue an application driven project and consider the complete system development cycle. Although it certainly has not been easy at all times, most of the identified questions and hypotheses of this thesis would have never been formulated without regarding such a bigger picture.
I would also like to thank Prof. Dr. Gemma Roig for discussions and feedback. I have welcomed and valued the additional perspective.

I thank the European Union's Horizon 2020 research and innovation programme under grant agreement No. 687384 for the funding which I have received while working on the AEROBI (AErial Robotic system for in-depth Bridge Inspection by contact) project. This includes all members of the consortium, who have helped tremendously in providing the necessary application user perspective, requirements and feedback.

My thanks go out to all my former and current colleagues. I appreciate the countless discussions and often wild speculations that have helped me in widening my horizon. I especially thank all my collaborators, who have assisted in strengthening my work and hardening it towards the authored publications. Above all, thank you Dr. Iuliia Pliushch, Sagnik Majumder, Yong Won Hong, Sreenivas Murali, Tobias Weis, Dr. Kishore Konda and Timm Hess.

I am grateful for being given the chance to co-advise multiple theses and the granted teaching opportunities. To all the students of my classes, thank you for your continued interest and feedback. You have helped me reinforce my own knowledge. To the students to whom I could extend my guidance to for their own theses, thank you for your trust Sagnik Majumder, Timm Hess, Hannah Wendland, Sina Dietzel, Achref Jaziri and Nicolas Lupp. There was much I could learn from you.

Finally, I would like to give thanks to my family from the bottom of my heart. To my parents, your continual assistance has been invaluable to me. To Fabian, you have served as my cornerstone in life for the last couple of years.