Understanding how visual information is represented in humans and machines

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

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

von
Kshitij Dwivedi
aus Kanpur, India

Frankfurt am Main 2022
(D30)

vom Fachbereich Informatik und Mathematik der
Johann Wolfgang Goethe - Universität als Dissertation angenommen.

Dekan (Dean): Prof. Dr. Martin Möller

Gutachter (Supervisor):
Prof. Dr. Gemma Roig
Prof. Dr. Jochen Triesch

Datum der Disputation: 2022.09.27

Publications: This cumulative dissertation is based on the following manuscripts:

- Dwivedi, K., Roig, G. (2019). *Representation similarity analysis for efficient task taxonomy and transfer learning.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12387-12396).

- Dwivedi, K., Huang, J., Cichy, R. M., Roig, G. (2020, August). *Duality diagram similarity: a generic framework for initialization selection in task transfer learning.* In European Conference on Computer Vision (pp. 497-513). Springer, Cham.

- Dwivedi, K., Bonner, M. F., Cichy, R. M., Roig, G. (2021). *Unveiling functions of the visual cortex using task-specific deep neural networks.* PLoS computational biology, 17(8), e1009267.

- Dwivedi, K., Cichy, R. M., Roig, G. (2021). *Unraveling representations in scene-selective brain regions using scene-parsing deep neural networks.* Journal of cognitive neuroscience, 33(10), 2032-2043.

- Dwivedi, K., Roig, G., Kembhavi, A., Mottaghi R. *What do navigation agents learn about their environment?* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10276-10285).

A description of how these works are related and corresponding scientific contributions are provided in the Introduction section.

In addition, I was also a co-author in the following publications while I was pursuing my doctorate. These publications are not incorporated in the thesis.

- Huang, J., Dwivedi, K., Roig, G. (2019). *Deep anchored convolutional neural networks.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

- Cichy, R. M., Roig, G., Andonian, A., Dwivedi, K., Lahner, B., Lascelles, A., ... Oliva, A. (2019). *The algonauts project: A platform for communication between the sciences of biological and artificial intelligence.* arXiv preprint arXiv:1905.05675.

- Cichy, R. M., Dwivedi, K., Lahner, B., Lascelles, A., Iamshchinina, P., Graumann, M., ... Oliva, A. (2021). *The Algonauts Project 2021 Challenge: How the Human Brain Makes Sense of a World in Motion.* arXiv preprint arXiv:2104.13714.

- Graumann, M., Ciuffi, C., Dwivedi, K., Roig, G., Cichy, R. M. (2022). *The spatiotemporal neural dynamics of object location representations in the human brain.* Nature Human Behaviour, 1-16.

- Gifford A. T., Dwivedi, K., Roig, G., Cichy, R. M. (2022). *A large and rich EEG dataset for modeling human visual object recognition* . bioRxiv.

Co-advised Thesis: I also had the privilege of co-advising the following theses while pursuing my doctorate degree:

- Bersch, Domenic. *Towards a general library for deep learning models to understand the architecture of the visual cortex*

- Le Hong, Quang Anh. *Influence of Training Dataset Resemblance to Stimulus Set on Prediction Accuracy of Brain Activity*

- Pietschmann, Daniel and Vorpahl, Yannic. *Many-to-One Task similarity and its relationship with task transferability*

Contents

Deutsche zusammenfassung - German Summary

Im menschlichen Gehirn wird das auf der Netzhaut eintreffende Licht in sinnvolle Darstellungen umgewandelt, die es uns ermöglichen, mit der Welt zu interagieren. In ähnlicher Weise werden die RGB-Pixelwerte von einem tiefen neuronalen Netz (DNN) in sinnvolle Darstellungen umgewandelt, die für die Lösung einer Computer-Vision-Aufgabe relevant sind, für die es trainiert wurde. In meiner Forschung möchte ich daher Erkenntnisse darüber gewinnen, wie visuelle Informationen im menschlichen visuellen Kortex und in DNNs, die trainiert wurden visuelle Aufgaben zu lösen, dargestellt werden.

Die Hauptidee im ersten Teil der Arbeit besteht darin, die Repräsentationen sowohl des menschlichen visuellen Kortex als auch der DNNs zu untersuchen, indem DNNs verglichen werden, die für verschiedene Aufgaben trainiert wurden. Um dies zu erreichen vergleichen wir eine Hirnregion oder eine Schicht eines DNNs mit den aufgabenspezifischen Repräsentationen mehrerer DNNs, die für unterschiedliche Aufgaben trainiert wurden. Der Vergleich informiert uns über die Repräsentation, die für die Lösung der Computer-Vision-Aufgabe relevant ist und die der Repräsentation der Gehirnregion/des Ziel-DNNs am nächsten kommt.

Kapitel 1: Verständnis der Repräsentationen im menschlichen visuellen Kortex

Im ersten Kapitel konzentriere ich mich auf das Verständnis der Repräsentation verschiedener Regionen im visuellen Kortex. Wir untersuchen zunächst, ob unser vorgeschlagener Ansatz Einblicke in die Repräsentation einer Hirnregion liefert, die mit früheren Untersuchungen dieser Hirnregion übereinstimmen. Nachdem wir den

Ansatz validiert haben, können wir ihn anwenden, um die Repräsentationen von weniger untersuchten Hirnregionen zu verstehen und einige Einblicke in die funktionellen Aufgaben dieser Hirnregionen zu gewinnen. Daher validieren wir im ersten Teil von Kapitel 1 unseren Ansatz in den gut untersuchten szenenselektiven Regionen Occipital Place Area (OPA) und Parahippocampal Place Area (PPA). Im zweiten Teil des Kapitels wenden wir unseren Ansatz auf mehrere Regionen des visuellen Kortex an und geben somit Einblicke in deren Repräsentationen.

Sondierung Selektive Regionen der Szene

Szenenselektive Regionen sind Regionen im Gehirn, die im Vergleich zu Bildern aus anderen Kategorien und verschlüsselten Bildern eine hohe Reaktion auf Szenenbilder zeigen. In einer Neuroimaging-Studie wurde gezeigt, dass OPA, eine der szenenselektiven Regionen, an der Vorhersage von den Regionen in einem Innenraum beteiligt ist, die für die Navigation relevant sind (Navigational affordance). Um diese "nagivational affordances' ausfinding zu machen, sind räumliche Informationen darüber, wo sich die Hindernisse befinden und wo der Ausgang in der Szene liegt, entscheidend. Daher war unsere Hypothese, dass die Repräsentation in OPA näher an einem Computermodell liegen sollte, das darauf trainiert ist, Szenen in verschiedene Komponenten (Hindernisse, Boden, Wand usw.) zu zerlegen, als an einem Modell, das darauf trainiert ist, die Kategorie der Szene zu identifizieren. Um unsere Hypothese zu evaluieren, haben wir Modelle für das Parsing und die Klassifizierung von Szenen ausgewählt und ihre Darstellung mit der Darstellung von OPA verglichen. Um die Verallgemeinerbarkeit unserer Ergebnisse zu gewährleisten, verwenden wir drei Architekturen sowohl für die Szenenanalyse als auch für die Szenenklassifikation. Wir fanden heraus, dass die Modelle zur Szenenanalyse bei allen drei Architekturen die Reaktionen der szenenselektiven Region OPA besser vorhersagten. Die Ergebnisse der Studie bestätigen unsere Hypothese und damit die Umsetzbarkeit des vorgeschlagenen Ansatzes zum Verständnis der Repräsentationen der Gehirnregionen im visuellen Kortex.

Untersuchung des gesamten visuellen Kortex

Nachdem wir den vorgeschlagenen Ansatz im vorangegangenen Teil validiert haben, erweitern wir in diesem Teil die Menge der betrachteten Modelle und Gehirnregionen. Um sicherzustellen, dass der Unterschied in der Repräsentationsähnlichkeit zwischen einer bestimmten Hirnregion und einem DNN nur auf die Aufgabe zurückzuführen ist, war unser Kriterium für die Modellauswahl, dass alle Modelle auf demselben Datensatz trainiert werden (kein Einfluss der Trainingsdaten) und eine identische Architektur haben sollten (kein Einfluss der Architektur). Daher wählten wir einen großen Satz von Modellen aus, die auf dem Taskonomy-Datensatz trainiert wurden und die für eine Vielzahl von Aufgaben trainiert wurden, von einfachen 2D-Aufgaben bis hin zu Aufgaben, die ein dreidimensionales Verständnis der Szene und semantisches Wissen über die Szene erfordern. Für die Hirnregionen wählen wir den gesamten visuellen Kortex aus und unterteilen ihn mithilfe eines probabilistischen anatomischen Atlasses in Regionen. Die Untersuchung in diesem Abschnitt ermöglichte es uns, die Repräsentationen aller Regionen im visuellen Kortex im Hinblick auf Computer-Vision-Aufgaben zu verstehen.

Kapitel 2: Verständnis von Repräsentationen in tiefen neuronalen Netzen

Im zweiten Kapitel wenden wir den vorgeschlagenen Ansatz an, um die von einem DNN gelernten Darstellungen zu verstehen, die genutzt werden um eine bestimmte Computer-Vision-Aufgaben zu erfüllen. Ein DNN besteht in der Regel aus mehreren Schichten, wobei jede Schicht eine Berechnung durchführt, bis die letzte Schicht eine Vorhersage für eine bestimmte Aufgabe durchführt. Das Training für verschiedene Aufgaben kann zu sehr unterschiedlichen Repräsentationen führen. Daher untersuchen wir im ersten Teil dieses Kapitels, in welchem Stadium sich die Repräsentation in DNNs, die für verschiedene Aufgaben trainiert wurden, zu unterscheiden beginnt. Wir untersuchen weiter, ob die DNNs, die auf ähnliche Aufgaben trainiert wurden, zu ähnlichen Repräsentationen führen und ob sie auf unterschiedliche Aufgaben trainiert wurden, zu noch unterschiedlicheren Repräsentationen führen. Im zweiten Kapitel untersuchen wir die Auswirkungen verschiedener Merkmalsnormal-

isierungen auf die Repräsentationsähnlichkeit und führen ein neues Maß ein, das verschiedene vorgeschlagene Ähnlichkeitsmaße vereinheitlicht. Des Weiteren untersuchen wir DNNs, die auf hochrangige semantische Aufgaben trainiert wurden, um zu verstehen, wie sich die Repräsentationen unterscheiden, wenn wir von frühen Schichten zu tieferen Schichten übergehen.

Repräsentative Ähnlichkeit zur Bewertung der Ähnlichkeit von Aufgaben

Wir wählten dieselbe Gruppe von DNNs aus, die im vorherigen Kapitel verwendet wurde und die auf dem Taskonomy-Datensatz für eine Reihe von 2D-, 3D- und semantischen Aufgaben trainiert wurden. Anschließend verglichen wir bei einem DNN, das für eine bestimmte Aufgabe trainiert wurde, die Darstellung mehrerer Schichten mit den entsprechenden Schichten in anderen DNNs. Anhand dieser Analyse wollten wir herausfinden, wo in der Netzwerkarchitektur aufgabenspezifische Repräsentationen auftauchen. Wir fanden heraus, dass die Aufgabenspezifität zunimmt, wenn wir tiefer in die DNN-Architektur eindringen und sich ähnliche Aufgaben in Gruppen zusammenschließen. Wir fanden heraus, dass die Gruppierung, die wir anhand der Ähnlichkeit der Repräsentation gefunden haben, in hohem Maße mit der Gruppierung auf der Grundlage des Transferlernens korreliert, was eine interessante Anwendung des Ansatzes zur Modellauswahl beim Transferlernen darstellt. Wir evaluieren die Beziehung zwischen Transferlernen und repräsentativer Ähnlichkeit anhand von 20 Aufgaben aus dem Taskonomy-Datensatz und semantischen Segmentierungsaufgaben aus dem Pascal VOC-Datensatz. Wir bewerten auch den Einfluss der Modellarchitektur und der Anzahl der Bilder auf die Beziehung zwischen repräsentativer Ähnlichkeit und Transferlernen.

Dualitätsdiagramm Ähnlichkeit

Während meiner Arbeit an den vorangegangenen Projekten wurden neue Maße zum Vergleich von DNN-Darstellungen eingeführt. In dieser Arbeit haben wir die Gemeinsamkeiten der verschiedenen Maße identifiziert und die verschiedenen Maße in einem einzigen Rahmen vereint, der als Dualitätsdiagrammähnlichkeit bezeichnet wird. Diese Arbeit eröffnet neue Möglichkeiten für die Entwicklung besserer Ähn-

lichkeitsmaße zum Verständnis von DNN-Repräsentationen. Wir zeigen eine viel höhere Korrelation mit Transfer-Lernen als bisherige State-of-the-Art-Maße und erweitern sie auf das Verständnis schichtweiser Repräsentationen von Modellen, die auf dem Imagenet- und Places-Datensatz unter Verwendung verschiedener Aufgaben trainiert wurden, und demonstrieren ihre Anwendbarkeit auf Transfer-Lernen.

In den beiden vorangegangenen Kapiteln haben wir die aufgabenspezifischen DNN-Repräsentationen verwendet, um die Repräsentationen im menschlichen visuellen Kortex und anderen DNNs zu verstehen. Wir waren in der Lage, unsere Ergebnisse in Bezug auf passive Computer-Vision-Aufgaben wie Kantenerkennung, semantische Segmentierung, Tiefenabschätzung usw. zu interpretieren. Dieser Ansatz hat zwei Einschränkungen:

- Die DNNs/Menschen setzen sich nicht aktiv mit der Umgebung auseinander, was dazu führt, dass nur einige wenige Regionen im Gehirn aktiv sind und nur einige einfache Aufgaben, die für einen Vergleich mit dem menschlichen Gehirn relevant sein könnten.
- Wir waren nicht in der Lage, die Repräsentationen auf menschlich interpretierbare Konzepte abzubilden.

Um der ersten Einschränkung zu begegnen, betrachten wir DNNs, die in der virtuellen Umgebung AI2Thor trainiert wurden und sich aktiv mit der Umgebung auseinandersetzen, um eine komplexe Aufgabe zu erfüllen. Um die zweite Einschränkung zu beheben, entwickeln wir einen neuen Ansatz, der frei verfügbare Annotationen in AI2Thor ausnutzt, um Neuronen auf menschlich interpretierbare Konzepte abzubilden.

Kapitel 3: Assoziierung künstlicher Neuronen mit Konzepten

Im letzten Kapitel stellen wir eine neue Methode Interpretability System for Embodied agEnts (iSEE) vor, die einzelne Neuronen von künstlichen Navigationsagenten, die in einer simulierten virtuellen Umgebung AI2Thor[1] trainiert wurden, auf menschlich interpretierbare Konzepte abbildet. Wir konzentrieren uns auf einfache

Basis-DNNs, die für die Ausführung von Objektziel und Punktziel Navigationsaufgaben trainiert wurden. Wir trainieren ein interpretierbares Modell (Gradient Boosted Tree), um Konzepte wie Hindernisse und die Entfernung zum Ziel aus den versteckten Einheiten des DNNs vorherzusagen. Anschließend wenden wir eine globale Erklärungsmethode namens SHAP[2] an, um herauszufinden, welche Einheiten für die Vorhersage des jeweiligen Konzepts relevant waren. Um die Kausalität unserer Ergebnisse zu bewerten, haben wir die Einheiten aus dem DNN entfernt und dann die Leistung bei der ursprünglichen Aufgabe bewertet.

In dieser Arbeit stellen wir die Fortschritte beim Verständnis von Repräsentationen im menschlichen visuellen Kortex (Kapitel 1) und in tiefen neuronalen Netzen (Kapitel 2 und 3) vor. In Kapitel 1 und 2 haben wir eine gut etablierte Methode namens Repräsentationsähnlichkeitsanalyse (RSA) angewandt, um neue Erkenntnisse über die Repräsentationen des Gehirns und der Ziel-DNNs zu gewinnen, indem wir ihre Repräsentationen mit DNNs verglichen haben, die für eine breite Palette von Computer-Vision-Aufgaben trainiert wurden. Die Ergebnisse in Kapitel 1 und 2 zeigen Einsichten in Bezug auf Computer-Vision-Aufgaben. In Kapitel 3 haben wir eine neue Methode entwickelt, um die Repräsentationen in den DNNs in Form von menschlich interpretierbaren Konzepten zu interpretieren, indem wir die frei verfügbare Grundwahrheit in Simulationsmaschinen nutzen

Introduction

The human brain transforms the incoming signals from different sensory organs into representations relevant for interacting with the environment. Thus, a crucial question for neuroscientists is to understand how sensory information is represented in the human brain, enabling intelligent behavior. Seeking the answer to the above question can benefit medical applications (e.g., neural prosthesis) and help gain better insights into designing artificial agents with human-like capabilities.

Similarly, artificial neural networks solving challenging problems transform the input signals into representations relevant to solving that particular task. Deep neural networks (DNNs) have made tremendous progress in almost every field of science including vision [3], speech [4], natural language processing [5], medicine [6], biology [7], nuclear fusion [8] and many others. To make DNNs more fair, transparent, interpretable, and acceptable to a wider community, it is crucial to understand what changes during the training to make a DNN's representation relevant for solving a task. We investigate the representations in DNNs because of two reasons: first, they are currently the best predicting models of human brain activity [9] and secondly, DNNs allow fast and extensive testing of methods developed to interpret the brain's representation thus encouraging the development of better interpretability methods [10, 11, 12, 13]. Therefore, in this work, our goal is to understand how sensory information is represented in the human brain and artificial neural networks.

Understanding representations related to all sensory signals in the human brain and wide ranges of inputs in the artificial neural networks is an overwhelming goal that requires developing concrete methodology in one specific modality and then expanding the framework to other modalities. Therefore in this work, we restrict our research to the visual part of the human brain, referred to as the human visual cortex, and artificial neural networks designed to solve challenging visual tasks.

The following section briefly reviews previous works investigating how visual information is represented in the human visual cortex and puts it in context with our

proposed approach.

Human visual cortex

The use of non-invasive functional imaging of the human brain, especially functional magnetic resonance imaging (fMRI), has led to a detailed understanding of how visual information is represented in different regions of the brain. To account for how different brain regions together transform the incoming light into meaningful representations for humans, functional specialization theory was proposed. Functional specialization suggests that different specialized neural pathways exist to represent different visual scene aspects. Different functional specialization leads to different representations of the visual scene. Therefore, we focus on understanding representations in different brain regions in this work.

Functional specialization in human visual cortex

In humans, there are two commonly used methods to reveal visual cortical regions using fMRI: retinotopy and functional specialization [14]. Retinotopy [15, 16] exploits the topographic mapping of the visual input from the retina to neurons in the visual cortex. To reveal the topographic mapping in the visual cortex, subjects are asked to fixate at a point, and visual stimuli are presented at selected locations. In a traveling wave, subjects view a high contrast flickering stimulus that rotates around the center (to find the angular selectivity of the cortical region) or expands through the visual field (to find the preferred eccentricity). In population receptive field (pRF) modeling, subjects view a traversing bar. Then a parametric model with parameters corresponding to a hypothetical receptive field and stimulus information is fitted to predict neural responses. Using these methods, researchers have identified several regions (V1, V2, V3, V4/V8, and V3a) that are arranged as parallel, mirror-symmetric bands on the unfolded cortex. Retinotopic methods however are not suitable for investigating where in the brain abstract concepts like object identity, actions, and 3D scene structure are represented.

Functionally specialized regions are identified by designing experiments focusing on different aspects of visual information such as motion, depth, color, shape, navigational affordances, and category selectivity. For example, there is evidence for

different regions showing selective responses to objects [17], faces [18], places [19], body parts [20], and reachable spaces [21]. Demonstrating the response selectivity to a specific category in a region does not necessarily mean that it is due to semantically meaningful features related to that category. The response selectivity could also be due to the presence of visual features that are present more often in one type of category than others. For example, recently, Vinken et al. [22] showed that face selectivity in macaque IT does not reflect a semantic code but a preference for visual features that are present more in faces than in non-faces. Also, several regions have not shown such functional selectivity, and there are debates over the functional definition of several regions which show such functional selectivity. Therefore, here we propose a new approach to finding representations of different brain regions using deep neural networks.

Deep neural networks for predicting visual cortex responses

In the past decade, deep neural networks have been the state-of-the-art approach to computational modeling of the brain. In the seminal work by Yamins and Dicarlo [23], they showed that deep neural networks predict macaque neural responses significantly better than other computational models. In a concurrent work by Razavi and Kriegeskorte [24], they showed similar results with both human and macaque responses in the inferior temporal (IT) cortex. These works led to a series of subsequent works [25, 26, 27, 28] demonstrating similar hierarchies in the DNNs and ventral visual pathway using human fMRI data. Guclu et al. [29] showed that an action recognition model can be used to model the dorsal pathway of the visual cortex. Action recognition DNN was used to predict dorsal stream responses to natural movies and revealed a correspondence between representations of DNN layers and regions in the dorsal stream. Inspired by Guclu et al. [29], Richard et al. [30] also used a DNN trained on action recognition to model the visual cortex. More recently, Mineault et al. [31] used a DNN trained in a self-supervised manner to predict self-motion parameters to model dorsal stream. They also showed that prediction of self-motion parameters leads to better accountability of dorsal stream responses than an action recognition DNN. In Bakhtiari et al. [32] they used a single DNN with two pathways trained using self-supervision to model both ventral and dorsal streams. The progress in using deep neural networks has been more focused

on finding the models that best predict neural responses in a given region. Several challenges have also been organized to bring researchers from the deep learning community to apply their expertise in predicting neural activity. Algonauts challenge 2019 [33] was organized to encourage researchers to develop models that predicted human fMRI and MEG responses to still images. Algonauts challenge 2021 [34] was organized to find the best models that predict human fMRI responses to short video clips. Brain-score 2022 [9] was organized to find the best models that predict macaque neurons' responses in different brain regions. A common theme of all the above works was finding the computational model that best predicts brain activity.

In this work, we identify a different potential of DNNs in neuroscience. We observed in earlier works comparing DNNs with neural responses that DNNs trained in object recognition predict neural responses in the inferior temporal cortex (IT), an area that is related to object categorization [35], better than randomly initialized DNNs. Similarly, DNNs related to functions of the dorsal stream (action recognition, self-motion estimation, dual-stream models) showed better predictivity of responses in the dorsal regions. Based on the above findings, we argue that DNNs trained on a particular task predict responses of a given brain region better than other DNNs since the task of the selected DNN and brain region are related and therefore have similar representations. Based on this argument, we hypothesize that if we compare DNNs trained on different computer vision tasks, we can use it to find how visual information is represented in a given brain region.

We are not the first ones to use DNNs to gain insight into visual cortex representations. The access to weights and gradients of deep neural networks allow interesting applications such as finding preferred stimulus for individual neurons. Bashivan et al. [11] first trained a linear model to predict macaque neuron responses in V4 from a DNN. Then, they optimized the input image to DNN to maximize the neuron's predicted activity resulting in preferred images for that neuron. A similar algorithm was also used to suppress a neuron's activity. Ponce et al. [12] used a genetic algorithm to generate an image that maximized macaque neuron responses in the IT region. Similar works in fMRI (Murty et al. [36], and Gu et al. [37]) found the optimal stimuli for category-selective brain regions. Seeliger et al. [10] proposed Neural Information Flow (NIF) that maps individual layers of a DNN to individual brain regions by training the DNN to predict neural responses in different regions. Then by interpreting what each layer has learned, we can gain insights into repre-

sentations of corresponding brain regions. More recently, Khosla et al. [13] trained DNNs to predict fMRI responses from scratch and by generating preferred images showed that category-selective regions are highly sensitive to visual patterns specific to their respective categories. Although finding preferred stimuli provides interesting insights into visual cortex representations, it is not always trivial to associate the features in the preferred stimulus with a human interpretable concept or a functional property. Therefore, in the first part of the thesis (Chapter 1 for visual cortex and Chapter 2 for DNNs), we find functions of the brain regions and DNNs in terms of computer vision tasks (or functions) and in the later part, develop a new method to associate artificial neurons with human interpretable concepts (Chapter 3).

In the previous section, we reviewed classical techniques to find how visual information is represented in different brain regions and discussed how DNNs can be a new promising tool to achieve what is not possible with classical methods. In the following section, we briefly review the use of DNNs in computer vision.

Deep neural networks for computer vision

In computer vision, DNNs have made significant improvements in several tasks such as image recognition [38], object detection [39, 40], semantic segmentation [39, 41], depth estimation [42], edge detection [43] among many others. In the following paragraphs we briefly review some of these tasks that are relevant to the present thesis.

Image recognition

In an image recognition task, we are provided an RGB image as the input, and the model is expected to identify which objects are present in the image. For instance, given an image of a dog and a cat sitting on a couch, the model should output high probabilities for the dog, cat, and couch categories and low probabilities for other categories, e.g., a table or a bed. To benchmark the image recognition performance, the Imagenet [38] dataset was introduced in 2010, which contains 1.4 million images from 1,000 object classes. The large scale of the dataset allowed for unprecedented opportunities to develop new models of object recognition. From the first computer vision DNN trained using data of this scale (Alexnet [3]), the performance on the

image recognition task (top1-accuracy) has improved from 63.3% to 90.94% [44] thus leading to great progress in the computer vision field. The DNNs trained on Imagenet are not only useful for image recognition but the representations learned on Imagenet have also shown to be relevant to many downstream tasks such as object detection [40], semantic segmentation [39], depth estimation [45], action recognition [46] and even explaining the neural responses in humans [33] and monkeys [47].

Semantic Segmentation

In the semantic segmentation task, we are provided an RGB image, and the model is expected to predict the category label of each pixel in the image. Here, the task requires both classification and localization, i.e., what categories are present and where in the scene are they present. There are multiple datasets such as MS-COCO [39], Pascal VOC [41], and ADE20k [48] to benchmark the progress of semantic segmentation. Standard models used in the semantic segmentation task have an encoder-decoder type of architecture. The encoder architecture usually is derived from DNNs trained on object recognition on the Imagenet dataset. The encoder transforms the pixel-level information of the image into semantic information. The encoder output is generally low resolution, and therefore decoders are designed to map semantic information back to pixel locations.

Depth estimation

In the depth estimation task, we are provided an RGB image, and the model is expected to predict the depth value of each pixel, i.e., how far a given pixel is from the camera. The depth estimation task requires a 3-Dimensional understanding of the scene. A standard dataset to benchmark the progress for depth estimation is NYUv2 [42]. The DNN architectures for depth estimation, similar to semantic segmentation, have an encoder-decoder architecture.

Edge Detection

In the edge detection task, we are provided an RGB image, and the model is expected to predict the magnitude and direction of edges in an image. Edge detection requires

a low-level understanding of the scene and has gained less attention from the deep learning community than other semantically relevant tasks.

Object/Point Goal Navigation

In the later part of the thesis (Chapter 3), we focus on tasks that involve embodied learning where the model is required to interact with the environment to perform the task successfully. We consider two navigation tasks where the model (artificial agent) is placed in a random location in a room in the virtual environment called AI2Thor [1]. In the object goal navigation (Objectnav) task, the agent is given a goal object, and the task is to navigate in the room to reach closer to the target object. In the point goal navigation (Pointnav) task, the agent is given a target coordinate in the room, and the task is to navigate in the room to reach closer to the target coordinate. Objectnav task requires both semantic and 3D scene understanding, while to perform Pointnav task, semantic information is not that relevant. The DNNs used in the navigation task usually have a visual encoder that is derived from DNNs trained on Imagenet, and then visual information and goal information is combined and fed into recurrent layers to take into account the agent's previous actions. The agents are trained using reinforcement learning, where the agent is rewarded if the agent reaches a goal.

All of the above tasks require different aspects of visual scene understanding, and therefore the DNNs solving these tasks must be learning different representations. In this work, we are interested in discovering how training on different tasks leads to different representations in DNNs with similar architecture (Chapters 2 and 3).

There have been several approaches to understanding the representations in DNNs: representational similarity analysis [49], feature interpretation [50], and explaining the model's decisions on individual examples [51, 52]. In this work, we mainly focus on representational similarity analysis, which is a widely accepted method to interpret representations in both neuroscience and the deep learning community. In the later part, we take inspiration from feature interpretation methods and explainability approaches to develop a new method that can associate artificial neurons to human interpretable concepts.

Representational similarity analysis

Representational similarity analysis generally involves the comparison of two feature spaces. These feature spaces could be fMRI-DNN, fMRI-magnetoencephalography (MEG), DNN-DNN, fMRI-Electroencephalography (EEG), and many other possible combinations depending on the questions a researcher is interested in answering. In this work, we primarily focus on the fMRI-DNN and DNN-DNN combination. DNN-DNN comparison using representational similarity analysis can reveal interesting insights about DNN architecture, learning, etc. DNN-fMRI combination can reveal which DNN best predicts fMRI responses in a given brain region.

In representational similarity analysis, we first extract features of a selected set of data points for both the feature spaces. Then, we map the selected data points in both the feature spaces. This mapping creates two graphs representing relationships between individual data points in the feature spaces we are interested in comparing. The main idea is that if two feature spaces are similar, their corresponding relationship graphs should also be similar. To compare two graphs, we calculate the pairwise distances between individual data points in each feature space resulting in two pairwise dissimilarity matrices for each feature space. Then, by comparing the pairwise dissimilarity matrices, we evaluate how similar are the two feature spaces of interest.

The idea behind representational similarity analysis dates back to the 1970s when Escoufier et al. [53] proposed the RV coefficient to quantify similarities between two feature spaces. In neuroscience, Kriegeskorte et al. [54] introduced Representational Similarity Analysis (RSA) to connect different branches of neuroscience (computational modeling, human fMRI, EEG, MEG, monkey cell recordings, and monkey fMRI) together. In deep learning, Kornblith et al. [55] introduced Centered Kernel Alignment (CKA), a measure inspired by RV coefficient that is capable of determining correspondences between layers of DNNs trained from different random initializations.

Representational similarity analysis in Neuroscience

In neuroscience, representational similarity analysis has been applied to reveal several interesting insights about the human brain. Before the introduction of large-

scale datasets in neuroimaging, researchers used carefully selected images (in orders of 100) to distinguish key properties in the visual scene. Taking samples of images from different categories (faces, places, objects, body parts) or supercategories (animate, inanimate) can help identify regions in the brain where the visual representation can distinguish between categories. Similarly, one can use a sample of images and collect human behavioral data and use it to identify where in the brain the visual representation is closer to human behavioral responses. In Groen et al. [56], they collected behavioral similarity judgments and compared them with human fMRI responses in the visual cortex to show that there is limited correspondence between localized fMRI responses and behavioral similarity judgments. In Bonner and Epstein [57], they collected human behavioral responses related to navigational affordances and compared them to fMRI responses in scene-selective regions showing that occipital place area (OPA) encodes navigational affordances. One can also compare features from a computational model to find whether a computational model has a representation similar to a given brain region. Tsantani et al. [58] created different image computable and perceived property models to show that face-selective regions occipital face area (OFA) and fusiform face area (FFA) encode distinct face identity information.

Representational similarity analysis in Deep Learning

In deep learning, representational similarity analysis based methods have been used in the literature to find out how representation changes in the DNN across different layers during training [59]. Another research direction is to use representational similarity analysis to investigate what impact different initialization seeds have on the final representation of DNNs trained on object classification task on Imagenet [55, 60]. Similarly, these analyses have also been used to find differences in different DNN architectures. In Nguyen et al. [61], they applied representational similarity analysis to compare the representations of wide and deep DNNs. Recently Raghu et al. [62] compared the representations learned by recently introduced vision transformers [63] with the standard convolutional architectures. In multimodal learning, CLIP [64] model uses representational similarity analysis to train a model using text and image pairs by maximizing similarities between relationship graphs of text and image embeddings.

In the first part of the thesis, our main idea is to probe the representations of both the human visual cortex and DNNs by comparing DNNs trained on different tasks. Given a brain region or a layer of a DNN, we compare it to task-specific layers' representation of multiple DNNs trained to perform different tasks. The comparison informs us of representations relevant to solving the question of which computer vision task is closest to the brain region's/target DNNs representation.

Chapter 1: Understanding representations in the human visual cortex

In the first chapter, I focus on understanding the representation of different regions in the visual cortex. We first investigate if our proposed approach provides insights into a brain region's representation that converges with previous investigations of that brain region. Having validated the approach, we can then apply it to understanding representations of under-investigated brain regions and provide some insights into the functional roles of those brain regions. Therefore in the first part of chapter 1, we validate our approach in well-investigated scene-selective regions Occipital Place Area (OPA) and Parahippocampal Place Area (PPA). In the second part of the chapter, we apply our approach to multiple regions of the visual cortex, providing insights into their representations.

Probing Scene selective regions

Scene selective regions are regions in the brain that show a high response to scene images as compared to images from other categories and scrambled images. In a neuroimaging study [57], it was shown that OPA, one of the scene-selective regions, is involved in the prediction of regions in an indoor space that are relevant to navigation (navigational affordance). To find the navigational affordance, spatial information about where the obstacles are and where the exit is located in the scene is crucial. Therefore, our hypothesis was that representation in OPA should be closer to a computational model that is trained to parse a scene into different components (obstacles, floor, wall, etc.) compared to a model that is trained to identify the category of the scene. To evaluate our hypothesis, we selected scene parsing and scene

classification models and compared their representation with OPA's representation. To ensure the generalizability of our findings, we used three architectures for both scene parsing and scene classification tasks. Our findings confirmed our hypothesis, therefore validating the feasibility of the proposed approach in understanding the representations of the brain regions in the visual cortex.

Probing the entire visual cortex

Having validated the proposed approach in the previous section, in this section, we increase the number of models and brain regions considered. To ensure that the difference in representational similarity between a given brain region and a DNN is only due to the task, our criteria for model selection was that all the models should be trained on the same dataset (no influence of training data) and have identical architecture (no influence of architecture). Therefore, we selected a large set of models trained on the Taskonomy dataset that were trained to perform a diversity of tasks ranging from low-level 2D tasks to tasks that required 3-dimensional scene understanding and semantic knowledge about the scene. For the brain regions, we selected the entire visual cortex and subdivided it into regions using a probabilistic anatomical atlas [65]. The investigation in this section allowed us to understand representations of all regions in the visual cortex in terms of computer vision tasks.

Chapter 2: Understanding representations in a deep neural network.

In the second chapter, we apply the proposed approach to understand the representations learned by a DNN to perform a given computer vision task. A DNN usually consists of multiple layers, each layer performing a computation leading to the final layer that performs prediction for a given task. Training on different tasks could lead to very different representations. Therefore in the first part of this chapter, we investigate at which stage the representation in DNNs trained on different tasks starts to differ. We further investigate if the DNNs trained on similar tasks lead to similar representations and on dissimilar tasks lead to more dissimilar representations. In the second chapter, we investigate the impact of different feature normalizations on

representational similarity and introduce a new measure that unifies different proposed similarity measures. We further probe DNNs trained on high-level semantic tasks to understand how representations differ as we go from early layers to deeper layers.

Representational similarity for assessing task similarity

We selected the same set of DNNs used in the previous chapter that were trained on the Taskonomy dataset on a diverse range of 2D, 3D and semantic tasks. Then, given a DNN trained on a particular task, we compared the representation of multiple layers to corresponding layers in other DNNs. From this analysis, we aimed to reveal where in the network architecture task-specific representation start appearing. We found that task specificity increases as we go deeper into the DNN architecture, and similar tasks start to cluster in groups. We found that the grouping using representational similarity was highly correlated with grouping based on transfer learning, thus creating an exciting application of the approach to model selection in transfer learning. We evaluate the relationship between transfer learning and representational similarity on 20 tasks from the Taskonomy dataset and the semantic segmentation task from the Pascal VOC dataset. We also evaluate the influence of model architecture and the number of images on the relationship between representational similarity and transfer learning.

Duality Diagram Similarity

While I was working on the previous projects, new measures [55] were introduced to compare DNN representations. In this work, we identified the commonalities in different measures and unified different measures into a single framework referred to as duality diagram similarity. This work opens up new possibilities for creating better similarity measures to understand DNN representations. After demonstrating a much higher correlation with transfer learning than previous state-of-the-art measures, we extended it to understanding layer-wise representations of models trained on the Imagenet and Places dataset using different tasks and demonstrated its applicability to transfer learning.

In the previous two chapters, we used the task-specific DNN representations to

understand the representations in the human visual cortex and other DNNs. We were able to interpret our findings in terms of passive computer vision tasks such as edge detection, semantic segmentation, depth estimation, etc. This approach has two limitations: 1. The DNNs/humans do not actively engage with the environment leading to only a few active regions in the brain and only a few simple tasks that could be relevant for comparison with the human brain. 2. We could not map the representations to human interpretable concepts. To address the first limitation, we consider DNNs trained in virtual environment AI2Thor that actively engage with the environment to perform a complex task. To address the second limitation, we develop a new approach that exploits freely available annotations in AI2Thor to map artificial neurons to human interpretable concepts. Several works in the feature interpretability field have attempted to map individual neurons or groups of neurons to interpretable concepts. In another line of research, people have attempted to explain model decisions in terms of human interpretable concepts. We briefly review these two research directions here that motivated us to develop a new method to associate artificial neurons to concepts.

Feature interpretability

The idea to map neurons to a concept goes back to seminal work from Hubel and Wiesel [66] in the 1950s where they measured the activity of a cat's neuron with respect to different orientations and found that it was selective to certain orientations. The image that maximizes a neuron or group of neurons' activity is known as the preferred image. Similar techniques in neuroscience have led to findings such as grandmother neuron [67], neurons selective to celebrities like Halle Berry [68], and category-selective regions [18, 19, 21, 17]. Inspired by these works in neuroscience, deep learning researchers started using similar techniques to find what an artificial neuron encodes.

A straightforward approach to finding preferred images for artificial neurons is to feed a large number of images through the network and identify which images lead to maximal activation of a given neuron [69, 70]. A computational challenge in this approach is that for probing a single neuron, one might need to feed a large number of images. Another challenge is that it is quite possible that maximally

activating images for this neuron might not be in the probe dataset. Finally, the preferred images may not be explicitly informative about what common features of the preferred images are causing the neuron to activate. To address this, Nguyen et al. [71] proposed synthesizing the image using gradients of the DNN that maximizes a given neuron's activity. The synthesized images, however, are not interpretable for all the neurons. Therefore, in subsequent work, Nguyen et al. [72] added a natural image prior using a generative model that ensured the synthesized images looked like natural images. While the method of preferred images provides some qualitative information about what a neuron encodes, it is not trivial how to quantify that association.

To quantify the association of a neuron with concepts, Zhou et al. [73] used the receptive field of the neurons to segment the most activating images and then asked humans to annotate the concept using the segmented images. Bau et al. [74] quantified the concept-to-neuron association by comparing the segmentation maps generated by neurons' spatial activation with ground truth segmentation annotations from the semantic segmentation dataset. More recently, Hernandez et al. [75] used natural language descriptions to label the neurons. Net2Vec [76] extended the approach of [74] to find out whether a single neuron or a group of neurons together encode a given concept. They trained a linear model to predict the presence of a concept from a group of neurons and then used the weights of the linear model to identify which neurons were relevant for predicting the concept's presence. However, interpretability using weights of a linear model can assign weights to noisy inputs when the mapping is non-linear, as shown in Lundberg et al. [2].

All of the above methods require human annotations to quantify the association of a concept with a neuron. In Chapter 3, we note the potential of virtual simulators such as AI2Thor [1], Habitat [77] that have extensive annotations available for free and hence are suitable for the development of the new generation of interpretability frameworks. Further, we do not assume linearity in mapping between neurons and concepts, thus making our proposed approach generic.

Explainability

Explainability research aims to explain why a model is making particular decisions. Initial research in explainability of computer vision models focused on finding which pixels were relevant for making a class prediction [78, 79, 80, 81, 82] using DNN gradients. A common idea behind these methods is to propagate the gradient from the output back to the input to produce a heatmap that indicates which pixels were relevant for the model's output. Another popular approach is additive feature attribution [83, 84, 85] where the impact of adding an input feature in the model's prediction is used to quantify its relevance. A limitation of these methods is that the explainability is on pixel level and not on human interpretable features. To address this limitation, Kim et al. [86] introduced concept vectors instead of raw pixels to explain model predictions. However, this method required additional annotations for concepts, and therefore Ghorbani et al. [87] proposed a method to automatically extract visual concepts and then use those to explain model predictions.

In this work, we notice the potential of explainability methods in interpreting representations learned by the model's hidden units (neurons). The above methods are generally used to explain which input features were most relevant for a model's prediction. What if the input to the model is hidden units and the output is a human interpretable concept? Then using explainability methods, we can find out which hidden units were relevant in the prediction of that concept and hence are encoding that concept.

However, most of the works we discussed above provide local explanations, i.e., given an input image and model predictions, it identifies which features of the input image were relevant for this particular prediction. To identify which hidden units encode a concept, the explainability should be global, i.e., over multiple samples. Therefore, we use a global explainability SHAP [2] method in this work for the following reasons: (a) it provides a unique solution with three desirable properties: local accuracy, missingness and consistency [84], (b) it unifies several model agnostic [83, 85] and tree-based explanation methods [88], and (c) it provides explanations on both local (single example) and global (dataset) levels.

Chapter 3: Associating artificial neurons to concepts

In the last chapter, we present a new method **I**nterpretability **S**ystem for **E**mbodied ag**E**nts (iSEE) that maps individual neurons of artificial navigation agents trained in simulated virtual environment AI2Thor [1] to human interpretable concepts. We focus on simple baseline DNNs trained to perform Objectnav and Pointnav tasks. We train an interpretable model (Gradient Boosted Tree) to predict concepts like obstacles and target visibility from the hidden units of the DNN. Then, we apply a global explainability method called SHAP [2] to find out which units were relevant for predicting the given concept. To evaluate the causality in our findings, we removed the units from the DNN and then evaluated the performance of the ablated models on the original task.

Summary

In this thesis, we present the progress towards understanding how visual information is represented in the human visual cortex (Chapter 1) and deep neural networks (Chapters 2 and 3). In Chapters 1 and 2, we applied a well-established method called representational similarity analysis (RSA) to reveal new insights into the brain (Figure0.1b) and target DNN (Figure0.1c) representations by comparing their representations with DNNs trained on a wide range of computer vision tasks (Figure0.1a). The findings in Chapters 1 and 2 reveal insights in terms of computer vision tasks. In Chapter 3, we developed a new method to interpret the representations in the DNNs in terms of human interpretable concepts (Figure0.1d) by exploiting the freely available groundtruth in simulation engines.

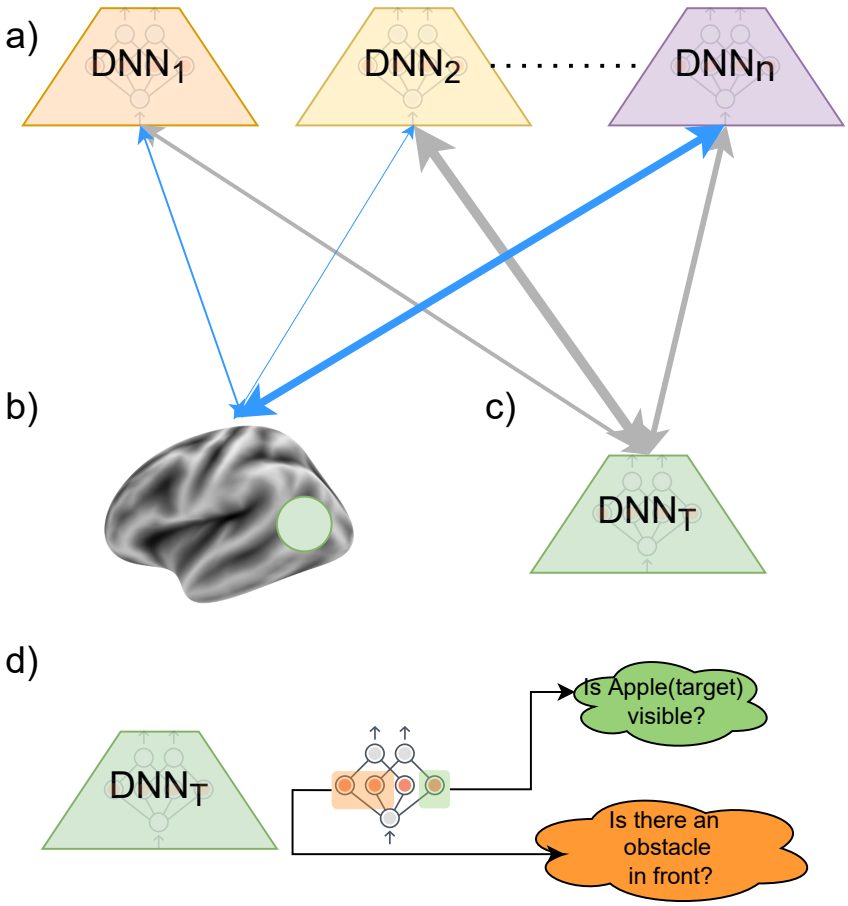

Figure 0.1: **Overview:** a) Given a set of DNNs trained on n tasks, b) In Chapter 1, we compare representations of n DNNs to a brain region's representations to reveal insights about brain representation in terms of n tasks. c) In Chapter 2, we compare representations of n DNNs to a target DNN's representation to reveal insights about this DNN's representations in terms of n tasks. d) In Chapter 3, we develop a new method to find out where in the hidden layers of a target DNN are the concepts (like target visibility, obstacle detection) encoded.

1 Understanding representations in the human visual cortex

1.1 Unraveling representations in scene-selective brain regions using scene-parsing deep neural networks

Published in final edited form as:
J Cogn Neurosci. 2021 September 01; 33(10): 2032–2043. doi:10.1162/jocn_a_01624.

Unraveling Representations in Scene-selective Brain Regions Using Scene Parsing Deep Neural Networks

Kshitij Dwivedi[1,2,*], **Radoslaw Martin Cichy**[1,†,*], **Gemma Roig**[2,†,*]

[1]Department of Education and Psychology, Free Universität Berlin, Germany

[2]Department of Computer Science, Goethe University, Frankfurt am Main, Germany

Abstract

Visual scene perception is mediated by a set of cortical regions that respond preferentially to images of scenes, including the occipital place area (OPA) and parahippocampal place area (PPA). However, the differential contribution of OPA and PPA to scene perception remains an open research question. In this study, we take a deep neural network (DNN)-based computational approach to investigate the differences in OPA and PPA function. In a first step we search for a computational model that predicts fMRI responses to scenes in OPA and PPA well. We find that DNNs trained to predict scene components (e.g., wall, ceiling, floor) explain higher variance uniquely in OPA and PPA than a DNN trained to predict scene category (e.g., bathroom, kitchen, office). This result is robust across several DNN architectures. On this basis, we then determine whether particular scene components predicted by DNNs differentially account for unique variance in OPA and PPA. We find that variance in OPA responses uniquely explained by the navigation-related floor component is higher compared to the variance explained by the wall and ceiling components. In contrast, PPA responses are better explained by the combination of wall and floor, that is scene components that together contain the structure and texture of the scene. This differential sensitivity to scene components suggests differential functions of OPA and PPA in scene processing. Moreover, our results further highlight the potential of the proposed computational approach as a general tool in the investigation of the neural basis of human scene perception.

Keywords

Occipital Place Area; Parahippocampal Place Area; Deep Neural Networks; Functional Magnetic Resonance Imaging; Navigational Affordance model; Representational Similarity Analysis; Scene parsing

[*]To whom correspondence should be addressed: roig@cs.uni-frankfurt.edu; rmcichy@zedat.fu-berlin.de; kshitijdwivedi93@gmail.com.
[†]jointly directed work

Conflict of interest. All authors declare that they have no conflicts of interest.

Europe PMC Funders Author Manuscripts

1 Introduction

Visual scene understanding is a fundamental cognitive ability that enables humans to interact with the components and objects present within the scene. Within the blink of an eye [Potter 1975, Thorpe et al. 1996, Li et al. 2007, Greene and Oliva, 2009] we know what type of scene we are in (e.g. kitchen, or outdoors), as well as its spatial layout and the objects contained in it.

Research on the neural basis of scene understanding has revealed a set of cortical regions with a preferential response to images of scenes over images of objects. These regions are the parahippocampal place area (PPA) [Epstein and Kanwisher, 1998], occipital place area (OPA) [Dilks et al., 2013, Hasson et al., 2003], and retrosplenial cortex (RSC) [O'Craven and Kanwisher, 2000]. To investigate the distinct function each of these place regions has, subsequent research has begun to tease apart their commonalities and differences in activation profile and representational content [Epstein and Kanwisher, 1998, Hasson et al., 2003, Dilks et al., 2013, Bonner and Epstein, 2017, Silson et al., 2015, O'Craven and Kanwisher, 2000]. However, a complete picture of how scene-selective regions together orchestrate visual scene understanding is still missing.

To gain further insights, a promising, but relatively less explored approach is computational modelling of brain activity. Recently, large advances have been made in modeling activity in visual cortex using deep neural networks (DNNs) trained on object categorization tasks [Krizhevsky et al., 2012] in both human and non-human primates [Yamins et al., 2014, Khaligh-Razavi and Kriegeskorte, 2014, Cichy et al., 2016]. Inspired by this success, researchers have also begun to use DNNs trained on scene categorization to investigate scene-selective cortex [Cichy et al., 2017, Bonner and Epstein, 2018, Groen et al., 2018].

In this process two issues have emerged that need to be addressed. First, while DNN trained on categorization tasks currently do best in predicting activity in scene-selective cortical regions, they do not account for all explainable variance. One particularly promising direction is the exploration of models trained on tasks different from categorization that might more closely resemble the brain region's functionality, and thus predict brain activity better [Yamins et al., 2014, Cichy and Kaiser, 2019]. Second, it remains unclear what is the nature of the representations in the DNNs that gives them their predictive power. Thus, additional effort is needed to clarify what these representations are.

To address the above issues, we investigated neural activity in the scene-selective cortex using DNNs trained on scene parsing instead of categorization. A scene parsing task requires the DNN to predict the location and category of each scene component in the image. While the scene-categorization task requires only recognizing the scene category, the scene parsing task requires deeper scene understanding involving categorization as well as a grasp of the spatial organization of components and objects within the scene. In order to help interact with different objects and navigate within the scene, scene-selective brain regions should also encode the spatial organization of components within the scene. Therefore, we hypothesize that the scene parsing task is closer to the task the brain has to solve, and a

Europe PMC Funders Author Manuscripts

DNN trained on scene parsing will predict brain activity better than a DNN trained on scene categorization.

To evaluate our hypothesis, we compared the power of DNNs trained on scene parsing versus categorization to predict activity in scene-selective cortical regions. For this we used an existing fMRI set of brain responses elicited by viewing scene images [Bonner and Epstein, 2017] and applied representational similarity analysis (RSA) to compare brain responses with DNNs. We found that scene parsing DNNs explain significantly more variance in brain responses uniquely in scene-selective regions than scene-classification DNNs.

We next investigated what representations in the DNNs trained on scene parsing gave the model its predictive power. For this we queried the DNN's representations of different scene components, considering components that were present in all stimulus images: wall, floor, and ceiling. We showed that different scene components predict responses in OPA and PPA differently: floor explained more variance in OPA than wall and ceiling, while wall explained more variance in PPA than floor and ceiling. Importantly, results were consistent across three different DNN architectures, showing the generalizability of our claims across architectures.

In sum, our results reveal differential representational content in scene-selective regions OPA and PPA, and highlight DNNs trained on scene parsing as a promising model class for modelling human visual cortex with well interpretable output.

2 Materials and Methods

2.1 fMRI data

We used fMRI data from a previously published study by Bonner and Epstein [2017] where all experimental details can be found, as well as instructions on how to download the data. The fMRI data were collected from 16 participants on a Siemens 3.0T Prisma scanner with a 64-channel head coil. The participants were presented with images of indoor environments. The images were presented for 1.5s on the screen followed by a 2.5s interstimulus interval. The images presented in the experiment were from a stimulus set of 50 color images depicting indoor environments. During the fMRI scan, participants were asked to fixate on a cross all the time and press a button if the image presented to them was a bathroom. The task required participants to attend to each image and categorize it. Voxel-wise (voxel size = 2 × 2 × 2 mm) responses to each image during each scan run were extracted using a standard linear model.

We here focus on two scene-selective regions of interest (ROIs): PPA and OPA. PPA and OPA were identified from separate functional localizer scans using a contrast of brain responses to scenes larger than to objects and additional anatomical constraints. For both ROIs and all the subjects, each voxel's responses in a given ROI were z-scored across images in a given run and then averaged across runs. The responses to a particular image were further z-scored across voxels.

Europe PMC Funders Author Manuscripts

2.2 Behavioral data

We used scene-related behavioral data representing navigational affordances assessed on the same stimulus set as used for recording the fMRI data described above [Bonner and Epstein, 2017]. To represent navigational affordances, a behavioral experiment was conducted in which 11 participants (different from the participants in the fMRI experiment) indicated the path to walk through each image of the indoor environment used in the fMRI study using a computer mouse. The probabilistic maps of paths for each image were created, followed by a histogram construction of navigational probability in one-degree angular bins radiating from the bottom center of the image. These histograms represent a probabilistic map of potential navigation routes from the viewer's perspective. The resultant histogram is referred to as the Navigational Affordance Model (NAM).

2.3 DNN Models

We selected DNNs optimized on two different scene-related tasks: scene classification and scene parsing. We describe both types of models in detail below.

Scene-classification models—For solving a scene-classification task, a DNN model is optimized to predict the probabilities of the input image belonging to a particular scene-category. For comparison with neural and behavioral data, we considered DNNs pre-trained on the scene-classification task on the Places-365 dataset [Zhou et al., 2017]. Places-365 is a large scale scene-classification dataset consisting of 1.8 million training images from 365 scene categories. We selected multiple scene-classification DNN architectures to investigate if our results generalize across different architectures. For this purpose, we considered 3 standard architectures: Alexnet [Krizhevsky et al., 2012], Resnet-18 [He et al., 2016], and Resnet-50 [He et al., 2016] and downloaded pre-trained models from: https://github.com/CSAILVision/places365.

Alexnet consists of 5 convolutional layers (conv1-conv5) followed by 3 fully connected layers (fc6, fc7, and fc8). Both Resnet-18 and Resnet-50 consist of a convolutional layer followed by four residual blocks (block1 - block4) each consisting of several convolutional layers with skip connections leading to a final classification layer (fc). Resnet-18 consists of 18 layers and Resnet-50 consists of 50 layers in total and they differ in the number of layers within each block.

Scene parsing models—We used scene parsing models trained on ADE20k scene parsing dataset [Zhou et al., 2016]. The ADE20k dataset (publicly available at http://groups.csail.mit.edu/vision/datasets/ADE20K/) is a densely annotated dataset consisting of 25k images of complex everyday scenes with pixel-level annotations of objects (chair, bed, bag, lamp, etc.) and components (wall, floor, sky, ceiling, water, etc.). The images were annotated using the LabelMe interface [Russel et. al 2008] by a single expert human annotator.

For the first set of experiments, where we compare the predictive power of scene parsing models to scene-classification models for explaining the neuronal responses, we design scene parsing models such that their encoder architecture is taken from scene-classification

Europe PMC Funders Author Manuscripts

models while their decoder architecture is task-specific. The encoder of the scene parsing models consists of the convolutional part (conv1-conv5 of Alexnet, and block1-block4 of Resnet18 and Resnet50) of scene classification models. The decoder of scene parsing models is adapted to the scene parsing task following the architecture proposed by Zhao et al. [2017]. It consists of a Pyramid Pooling module with deep supervision [Zhao et al., 2017] (d1), followed by a layer (d2) that predicts several spatial maps, one spatial map per scene component predicted, that represent the probability of the presence of that component at a given spatial location. The encoder weights of scene parsing models are initialized with the weights learned on the scene-classification task and decoder weights are initialized randomly. The scene parsing DNNs are then trained on ADE20k training data using a per-pixel cross-entropy loss which measures the performance of the classifier at each pixel whether the correct component is assigned the highest probability or not. The above procedure ensures that gain/drop in explaining neural/behavioral responses could only be due to additional supervision on the scene parsing task.

The aforementioned scene parsing DNNs are well suited for a direct comparison with the scene categorization DNNs as they have the same encoder architecture and were initialized with weights learned on the scene-categorization task. However, they are not comparable to state-of-the-art models in terms of accuracy on the scene parsing task. Since our aim is to reveal differences in representations of the scene areas in the brain by comparing scene components, for the second set of experiments, it is crucial to select components detected with DNNs from the literature that achieve the highest accuracy in scene parsing. For this reason, we selected 3 state-of-the-art models on the scene parsing task namely Resnet101-PPM [Zhou et al., 2016], UperNet101 [Xiao et al., 2018], and HR-Netv2 [Sun et al., 2019]. All the 3 state-of-the-art models were trained on the ADE20k dataset. We selected multiple models to investigate if the results we obtain are consistent across different models. Resnet101-PPM consists of a dilated version of the Resnet101 model (a deeper version of Resnet50 that consists of a total of 101 layers) trained on Imagenet as the encoder and a Pyramid Pooling module with deep supervision [Zhao et al., 2017] as the decoder. Due to the small receptive field in the feature maps, scene-parsing DNNs fail to correctly segment larger objects/components. The Pyramid Pooling module [Zhao et al., 2017] tackles this issue by fusing the feature maps that have different receptive field sizes to merge high spatial resolution information with low spatial resolution information for a better local and global level scene understanding. Upernet101 [Xiao et al., 2018] is based on the Feature Pyramid Network by Lin et al. [2017] that uses multi-level feature representations via a top-down architecture to fuse high-level semantic features with mid and low-level using lateral connections. Upernet101 also has a Pyramid Pooling Module before the top-down architecture to overcome the small receptive-field issue. HRNetv2 [Sun et al., 2019] relies on the importance of high-resolution feature maps for pixel labeling maps by maintaining high-resolution feature representations throughout the architecture and by merging information from both high and low resolution convolutions in parallel overcomes the small receptive field issue mentioned above. We downloaded all above mentioned models from https://github.com/CSAILVision/semantic-segmentation-pytorch.

To reveal performance differences between different models on the scene parsing task, we compared the performance of state-of-the-art models with the scene parsing models used for

Europe PMC Funders Author Manuscripts

comparison with scene-classification DNNs (see above, Alexnet, Resnet-18, Resnet-50). For the comparison, we calculated the mean intersection over union (mIoU) score of detecting all components for all the images from the ADE20k validation dataset. The IoU score is calculated by dividing the intersection between a predicted and corresponding ground truth pixel-level segmentation mask by their union. IoU is a standard metric to evaluate the overlap of a predicted and corresponding pixel-level mask of a particular component. Mean IoU is calculated by taking the mean of IoU scores across all images in the validation dataset for all components.

As illustrated in Figure 1a, a scene parsing model decomposes an image into its constituent components. This decomposition allows investigating which scene components are more relevant to explaining the representations in scene-selective brain regions. We first identified which scene components are present in all the images from the stimulus set used for obtaining fMRI responses. To achieve this, we feedforwarded all the 50 images in the stimulus set of the fMRI dataset through the models and checked the presence of all the components in the image. Since the DNN has been trained on an image dataset that is different from the set of stimuli used for the fMRI data, not all scene components predicted by the DNN appear in the stimulus set. In this particular set, we found that wall, floor, and ceiling were core scene components present in all images.

A scene parsing model outputs a spatial probability map for each component. To scale the spatial probability maps corresponding to different components in the same range, we normalized the spatial probability map for each component independently such that each pixel value lies in the range [0, 255]. We show the extracted normalized scene components corresponding to the wall, floor, and ceiling components for an example stimulus in Figure 1b.

2.4 Representational Similarity Analysis (RSA)

We applied representational similarity analysis (RSA; [Kriegeskorte et al., 2008] to compare DNN activations and scene components with neural and behavioral responses. RSA enables relating signals from different source spaces (such as here behavior, neural responses, DNN activation) by abstracting signals from separate source spaces into a common similarity space. For this, in each source space condition-specific responses are compared to each other for dissimilarity (e.g., by calculating Euclidean distances between signals) and the values are aggregated in so-called representational dissimilarity matrices (RDMs) indexed in rows and columns by the conditions compared. RDMs thus summarize the representational geometry of the source space signals. Different from source space signals themselves RDMs from different sources spaces are directly comparable to each other for similarity and thus can relate signals from different spaces. We describe the construction of RDMs for different modalities and the procedure by which they were compared in detail below.

fMRI ROI RDMs—First, for each ROI (OPA, and PPA), individual subject RDMs were constructed using Euclidean distances between the voxel response patterns for all pairwise comparisons of images. Then, subject-averaged RDMs were constructed by calculating the mean across all individual subject RDMs. We downloaded the subject averaged RDMs

Europe PMC Funders Author Manuscripts

of OPA and PPA from the link (https://figshare.com/s/5ff0a04c2872e1e1f416) provided in Bonner and Epstein [2018].

Navigational affordance model (NAM) RDMs—NAM RDMs were constructed using Euclidean distances between the navigational affordance histograms for all pairwise comparisons of images. We downloaded the NAM RDM from (https://figshare.com/s/5ff0a04c2872e1e1f416).

DNN RDMs—For all the DNNs we investigated in this work, we constructed the RDM for a particular layer using 1-ρ, where ρ is the Pearson's correlation coefficient, as the distance between layer activations for all pairwise comparisons of images. For scene classification DNN RDMs, we created one RDM for each of the 5 convolutional layers (conv1-conv5) and for the 3 fully connected layers (fc6,fc7, and fc8) for Alexnet, and the last layer of each block (block1 - block4) and the final classification layer (fc) of Resnet-18/Resnet-50 to compare with neural/behavioral RDMs. For scene parsing DNN RDMs, we created one RDM for each of the 5 convolutional layers (conv1-conv5) and for the 2 decoder layers (d1 and d2) for Alexnet, and the last layer of each block (block1 - block4) and 2 decoder layers (d1 and d2) of Resnet-18/Resnet-50 to compare with neural/behavioral RDMs.

Scene component RDMs—For each of the scene components investigated we constructed RDM for it using 1-ρ as the distance between normalized spatial probability maps of that scene component, based on all pairwise comparisons of images.

Comparing DNN and scene component RDMs with behavioral and neural RDMs—In this work, we pose two questions: first, whether scene parsing models can better explain scene-selective neural responses and navigational affordance behavioral responses better than scene-classification models, and second, whether the scene-components detected by scene parsing models reveal differences in representations of scene-selective ROIs.

To investigate the first question, we calculated the Spearman's correlation between the RDMs of different layers of a scene-classification DNN with a particular behavioral/neural RDM and selected the layer RDM that showed the highest correlation with the behavioral/neural RDM. We used the selected layer RDM as the representative RDM for that architecture. We repeated the same procedure to select the representative RDM from a scene parsing model. As a baseline for comparison, we also considered a randomly initialized model and selected the representative RDM from it.

We first found that deeper layers of both scene-classification and scene-parsing models showed a higher correlation with neural responses than earlier layers. A possible explanation behind the observed trend could be that deeper layers are more task-specific while early layers learn low-level visual features irrespective of tasks and therefore do not represent task-specific information in the model. Moreover, the highest correlation with PPA and OPA was found in deeper layers of the network further supporting this idea. We report the layer used to select the representative RDMs for each model in Table 1. To compare which model RDM (scene parsing/scene-classification/random) explains behavioral/neural RDM

better, we compared the correlation values of all three RDMs with behavioral/neural RDM (illustrated in Figure 1c for scene classification vs scene parsing).

To investigate whether the scene-components detected by scene parsing models reveal differences in representations of scene-selective ROIs, we computed the correlation between a scene component RDM and a neural RDM and compared which scene component explains better a particular ROI.

2.5 Variance Partitioning

While in its basic formulation RSA provides insights about the degree of association between a DNN RDM and a behavioral/neural RDM, it does not provide a full picture of how multiple DNN RDMs together explain the behavioral/neural RDM. Therefore, we applied a variance partitioning analysis that determines the unique and shared contribution of individual DNN RDMs in explaining the behavioral/neural RDM when considered in conjunction with the other DNN RDMs. Further, variance partitioning allows selection of multiple layers from a single model to explain the variance in neural and behavioral RDM.

We illustrate the variance partitioning analysis in Figure 1d. We assigned a behavioral/neural RDM as the dependent variable (referred to as predictand). We then assigned two model (DNN/scene component) RDMs as the independent variables (referred to as predictors). Then, we performed three multiple regression analyses: one with both independent variables as predictors, and two with individual independent variables as the predictors. Then, by comparing the explained variance (r^2) of a model used alone with the explained variance when it was used with other models, the amount of unique and shared variance between different predictors can be inferred (Figure 1d). In the case of three independent variables, we performed seven multiple regression analyses: one with all 3 independent variables as predictors, three with different combinations of 2 independent variables as predictors, and three with individual independent variables as the predictors.

To compare scene parsing and scene-classification models with a randomly initialized model as the baseline, the predictors were the respective DNN RDMs and predictands were the behavioral and neural RDMs. We performed variance partitioning analysis first using the selected representative RDMs (Table 1) for each model using RSA. In a second analysis, we relax the criteria of representing a model by single layer RDM and use multiple layer RDMs together to represent the model. We selected all the layer RDMs (Table 1) from each model (scene-classification/scene-parsing/random) and used them as predictors for variance partitioning.

To compare different scene components, the predictors were the respective scene component RDMs and predictands were the neural RDMs of scene-selective ROIs and behavioral RDM.

2.6 Statistical Testing

We applied nonparametric statistical tests to assess the statistical significance in a similar manner to a previous related study [Bonner and Epstein, 2018]. We assessed the significance of the correlation between neural/behavioral responses with a DNN through a permutation

test by permuting the conditions randomly 5000 times in either the neural ROI RDM or the DNN RDM.

From the distribution obtained using these permutations, we calculated p-values as one-sided percentiles. We calculated the standard errors of these correlations by randomly resampling the conditions in the RDMs for 5000 iterations. We used re-sampling without replacement by subsampling 90% (45 out of 50 conditions) of the conditions in the RDMs. We used an equivalent procedure for testing the statistical significance of the correlation difference and unique variance difference between the two models. The statistical outcomes were corrected for multiple comparisons using false detection rate (FDR) correction with a threshold equal to 0.05.

3 Results

3.1 Are scene parsing models suitable to account for scene-selective brain responses and scene-related behavior?

We investigated the potential of DNNs trained on scene parsing to predict scene-related human brain activity focusing the analysis on scene-selective regions OPA and PPA. To put the result into context we compared the predictive power of DNNs trained on scene parsing to DNNs trained on scene classification, which are currently the default choice in investigating scene-related brain responses and behavior [Bonner and Epstein, 2018, Groen et al., 2018, Cichy et al., 2017] and against a randomly initialized DNN as baseline. To ensure that the results can be attributed to differences in the task rather than being specific to particular network architecture, we investigated three different network architectures: Alexnet, Resnet18, and Resnet50.

We applied representational similarity analysis (RSA) to relate DNN models (scene classification,scene parsing, and random) with the brain responses in OPA and PPA (Figure 2a). We found that DNNs trained on scene parsing significantly predicted brain activity in all investigated regions ($p = 0.0001$ for Alexnet, $p = 0.0001$ for Resnet18, and $p = 0.0001$ for Resnet50). This shows that they are suitable candidate models for the investigation of brain function. We further found that DNNs trained on scene parsing explain as much or more variance in scene-selective regions than DNNs trained on scene-categorization. We note both scene parsing and scene classification DNNs explain significantly higher variance in scene-selective regions than a randomly initialized DNN across different architectures ($p<0.001$ for all the comparisons).

If scene parsing models are suitable models for predicting responses in scene-selective brain regions, and these regions underlie scene understanding, the models should predict scene-related behavior, too. We considered navigational affordance behavior operationalized as the angular histogram of navigational trajectories that participants indicated for the stimulus set. Paralleling the results on brain function, the investigation of behavior showed that DNNs trained on scene parsing predicted behavior significantly ($p = 0.0005$ for Alexnet, $p = 0.0005$ for Resnet18, and $p = 0.0005$ for Resnet50), and also significantly better than DNNs trained on scene-classification ($p = 0.01$ for Alexnet, $p = 0.0005$ for Resnet18, and $p = 0.03$ for Resnet50). Similar to results on brain function, we note that both scene parsing and

scene classification DNNs explain significantly higher variance in behavior than a randomly initialized DNN across different architectures.

While the RSA results above provided insights about the degree of association between a DNN RDM and behavioral/neural RDM, it cannot tell how multiple DNN RDMs together predict the behavioral/neural RDM. For this more complete picture, we conducted variance partitioning to reveal the unique variance of neural/behavioral RDMs explained by scene-classification and scene parsing DNN RDMs (Figure 2b). We observe from Figure 2b that scene parsing DNNs explain more variance uniquely (OPA: $p = 0.0003$ for Alexnet, $p = 0.0002$ for Resnet18, and $p = 0.0002$ for Resnet50 ; PPA: $p = 0.0002$ for Alexnet, $p = 0.0003$ for Resnet18, and $p = 0.0002$ for Resnet50) than scene-classification DNNs for both scene-selective ROIs. We further observe from Venn diagrams in Figure 2b that for scene-selective neural RDMs most of the variance explained is shared between scene-classification and scene parsing DNNs across all three architectures. The results suggest that scene parsing DNNs might be a better choice for investigating scene-selective neural responses than scene-classification DNNs.

We observe for behavior that the scene-classification DNNs explain nearly no unique variance, while on the other hand scene parsing DNNs explain significantly higher unique variance ($p = 0.001$ for Alexnet, $p = 0.0002$ for Resnet18, and $p = 0.0005$ for Resnet50) across all three architectures (see Figure 2b for unique variance and Venn Diagrams illustrating both unique and shared variances). The results suggest that since the scene parsing task takes into account the spatial arrangement of constituent components in the scene, a scene parsing DNN explains behavioral affordance assessments better than a scene classification DNN.

In both the above analysis, we selected the RDM of the layer of a model that showed highest RSA correlation with a neural/behavioral RDM as the representative RDM for the model but this brings the question how well a particular layer represents a model as a whole. To answer this question, we selected multiple layer RDMs from each model (scene-classification/ scene parsing/ random) and applied variance partitioning to find out how much of the variance in neural/behavior RDM is explained uniquely by a given model represented by multiple layer RDMs. We report the results in Figure 2c and observe a similar trend as observed in Figure 2b using layer RDMs that showed maximum correlation. The results suggest that layers that showed maximum correlation with a neural/behavior RDM were deeper in the network and therefore have more task-specific representation as opposed to earlier layers of the network. The variance explained by earlier layers contributes mostly to the shared variance of neural/behavioral RDMs explained by all models together.

Together, these results establish DNNs trained on scene parsing tasks as a promising model class for investigating scene-selective cortical regions in the human brain and for navigational behavior related to the spatial organization of scene components.

3.2 State-of-the-art scene parsing models for investigating scene components represented in the human brain

Models trained on the scene parsing task offer the possibility to selectively investigate which of the scene components (such as wall, ceiling or floor) they encode. But, what is the most suitable scene parsing model to compare to the brain? In the model comparison above our choice was guided by making models as similar to each other as possible in complexity to rule out that observed differences in accounting for brain activity are simply due to differences in model complexity. However, for in-depth investigation of scene-selective areas using scene-components it is crucial to choose models that detect the scene-components with as high accuracy as possible. Therefore, we compared the performance of different scene parsing models qualitatively and quantitatively to select the most accurate ones to compare with the responses of scene-selective brain areas.

For performance comparison on the scene parsing task, we chose the three models used above (Alexnet, Resnet18, and Resnet50), plus three state-of-the-art models of scene parsing: HRNetv2, Upernet101, and Resnet101-PPM. The state-of-the-art models achieve high performance by merging low-resolution feature maps with high-resolution feature maps to generate results in high spatial resolution. We illustrate their parsing performance qualitatively by examining their output on an example image (Figure 3). We observe that the scene parsing output generated by Resnet50 had smooth and less precise boundaries of components while Resnet101-PPM, Upernet101, and HRNetv2 detected components accurately with precise boundaries in their outputs.

To quantitatively compare model performance, we evaluated the performance of all models on the ADE20k validation dataset. For (mIoU) score of detecting all components for all the images from the ADE20k validation dataset. We report mIoU scores of individual components that were present in all images in the stimulus set: wall, ceiling, and floor. The results are reported in Table 2. They indicate that state-of-the-art models beat the complexity-matched models by a margin of 12% accuracy. Therefore, for in-depth investigation of representations in scene-selective brain areas we used the top 3 models, i.e., HRNetv2, Upernet101, and Resnet101-PPM.

3.3 Scene parsing networks reveal a differential contribution of wall, floor and ceiling components to representations in scene-selective regions

We investigated whether the scene components detected by a scene parsing DNN reveal a difference in the representational content of scene-selective ROIs. We focused on the three scene components - wall, floor, and ceiling - that were present in all the images of the stimulus set and compared them with scene-selective ROIs OPA and PPA and behavioral model NAM.

We first report the RSA results (Figure 4a) of comparing a scene component RDM with OPA, PPA and NAM for three state-of-the-art architectures HRNetv2, Upernet101, and Resnet101- PPM. We found that the correlation of the OPA RDM with the floor RDM was significantly higher than of the wall ($p = 0.02$ for HRNetv2, $p = 0.05$ for Upernet101, $p = 0.0002$ for Resnet101- PPM) and ceiling ($p = 0.01$ for HRNetv2, $p = 0.01$ for Upernet101,

p = 0.0002 for Resnet101- PPM) RDMs, and the correlation of the PPA RDM with the wall (p = 0.006 for HRNetv2, p = 0.001 for Upernet101, p = 0.01 for Resnet101- PPM) and floor (p = 0.006 for HRNetv2, p = 0.003 for Upernet101, p = 0.001 for Resnet101- PPM) RDMs was significantly higher than with the ceiling RDM. NAM which represents the navigational paths in the scenes showed the highest correlation with floor RDM which was significantly higher than the correlation with wall (p = 0.0002 for HRNetv2, p = 0.0002 for Upernet101, p = 0.0002 for Resnet101- PPM) and ceiling (p = 0.0002 for HRNetv2, p = 0.0002 for Upernet101, 0.0002 for Resnet101- PPM) RDM. The above results held consistently across all investigated models. Together, this suggests that OPA and PPA have differential representational content with respect to scene components.

To tease out how much variance in OPA and PPA is explained by individual scene components, we apply variance partitioning to find the unique and shared variance of OPA and PPA RDMs explained by different scene component RDMs. We report the variance partitioning results showing unique variance explained by each component along with Venn diagram illustrating both unique and shared variances in Figure 4b. We observed that in the case of OPA, the floor RDM explains significantly higher variance of OPA RDM uniquely compared to wall (p = 0.0002 for HRNetv2, p = 0.002 for Upernet101, p = 0.0008 for Resnet101- PPM) and ceiling (p = 0.0002 for HRNetv2, p = 0.002 for Upernet101, p = 0.0008 for Resnet101- PPM) RDMs. For PPA, the wall RDM explains significantly higher variance of PPA RDM uniquely compared to the floor (p = 0.001 for HRNetv2, p = 0.006 for Upernet101, p = 0.015 for Resnet101- PPM) and ceiling (p = 0.0005 for HRNetv2, p = 0.002 for Upernet101, p = 0.007 for Resnet101- PPM) RDMs. And for NAM, the floor RDM explains significantly higher variance as compared to the wall (p = 0.0002 for HRNetv2, p = 0.0002 for Upernet101, p = 0.0002 for Resnet101- PPM) and ceiling (p = 0.0002 for HRNetv2, p = 0.05 for Upernet101, p = 0.0002 for Resnet101- PPM) RDMs. Consistent with the RSA results above, this result reinforces the differences between OPA and PPA in the representation of scene components.

4 Discussion

In this study, we investigated the potential of scene parsing DNNs in predicting neural responses in scene-selective brain regions. We found that scene parsing DNNs predicted responses in scene-selective ROIs OPA and PPA better than scene-classification DNNs. We further showed that scene components detected by scene parsing DNNs revealed differences in representational content of OPA and PPA.

Previous work using DNNs to predict neural responses has emphasized the importance of the task for which the DNNs were optimized for [Yamins and DiCarlo, 2016, Khaligh-Razavi and Kriegeskorte, 2014, Richards et al., 2019]. We argue that the higher unique variance of scene-selective neural responses explained by scene parsing DNNs over scene-classification DNNs is due to such a difference in tasks. The scene classification task aims at identifying the category of the scene irrespective of the spatial organization of different components and objects in the scene. In contrast, the scene parsing task requires pixelwise labeling of the whole image and thus a more comprehensive understanding of the scene in terms of how different objects and components are spatially organized

within a given scene. Higher variance of the scene selective neural responses explained by the scene-parsing DNNs that encode spatial structure suggests that scene-selective neural responses also encode spatial structure of the scene entailing the position of different objects and components.This view is supported further by evidence from neuroimaging literature [Kravitz et al., 2011, Park et al., 2011] showing that scene-selective regions represent the spatial layout of scenes. The information about the spatial structure of the scene might be required by the brain to plan interaction within the scene, such as navigating to a target, reaching objects or performing visual search.

Our in-depth analysis using scene components revealed differential representations in OPA and PPA. We observed that OPA had a significantly higher correlation with floor than ceiling and wall. A possible explanation for the observed difference could be due to OPA's involvement in detecting navigational affordances [Bonner and Epstein, 2017], for which the floor plays a major role, and could explain the high sensitivity of OPA to stimulation in the lower visual field [Silson et al., 2015]. In contrast, we found that PPA shows a significantly higher correlation with wall as well as floor compared to ceiling. This could explain why PPA has sensitivity to the upper visual field [Silson et al., 2015]. A plausible explanation could be that detecting the wall is relevant to identifying the type of room, its texture [Henriksson et al., 2019, Park and Park, 2017] and landmarks [Troiani et al., 2012].

Previous work has already aimed at determining the nature of OPA representations by computational modelling [Bonner and Epstein, 2018] on the same fMRI dataset that was used in our study. For this, the authors determined for a DNN trained on scene categorization which individual DNN units most correlated with NAM and OPA and visualized those units using receptive field mapping and segmentation from Zhou et al. [2014]. The units extracted corresponded mostly to uninterrupted portions of floor and wall or the junctions between floor and wall. The results align with our findings that floor components explain OPA responses uniquely while the wall units could be attributed to shared variance explained by floor and wall components in our study. However, arguably segmentation maps extracted using the receptive field mapping method are less interpretable as they cannot be directly assigned to meaningful entities without additional human annotations [Zhou et al. 2014] or by comparing with ground-truth segmentation maps of meaningful entities [Bau et al. 2017]. Further, assigning a segmentation map of a unit obtained using receptive field mapping to a meaningful entity using ground-truth segmentation maps leads to less accurate segmentation maps [Bau et al. 2017] compared to the segmentation from a scene parsing DNN [Xiao et al. 2018] trained to segment components using ground truth segmentation maps. Thus, we believe our approach using scene-components generated by a scene parsing DNN to be particularly well suited to reveal the representational content of scene-selective brain regions.

In our analysis investigating navigational affordance behavioral responses, we found that scene-parsing DNNs explained significantly higher variance of behavioral RDM than the scene classification DNN. A plausible explanation for this finding is that determining navigational affordances requires spatial understanding of the scene, including floor and obstacle detection. Scene parsing tasks explicitly require such spatial understanding, whereas scene classification tasks can be performed without explicitly detecting the spatial

organization of objects and components in the scene. Further, the layers that show highest correlation with NAM in scene classification models are later convolutional layers which preserve spatial information of the image suggesting that spatial information is required to explain NAM. In the comparison of NAM with scene components we found that NAM is best explained by the floor component, while other components (wall, ceiling) explained insignificant unique variance. The above finding further reinforces our argument that scene-parsing DNNs explain NAM responses better due to the task requirement of finding spatial layout of objects and components. While in this study, we focused on showing the advantage of scene-parsing DNNs over scene-classification DNNs in explaining scene-selective neural and navigational affordance-related behavioral responses, our results do not rule out scene-classification DNNs as useful models to explain semantic behavioral responses related to scene category or semantic similarity.

A limitation of our study is that the differences revealed between OPA and PPA is based on the analysis of only 3 scene components. This is due to the limitations of the stimulus set, which consistently had only 3 components that were present in all 50 images. Future work should exploit the full richness of scene components provided by DNNs trained on scene parsing. For this a stimulus set would have to be designed that contains many components in all images of the stimulus set. Another possible direction would be to use stimuli with annotations and use these annotations directly to compare with the fMRI responses. The advantage of using a scene-parsing DNN over human annotations is that once the DNN is trained on scene-parsing, the components can be extracted for a new set of stimuli with zero cost as opposed to annotations, where human effort is required to annotate every new stimulus set.

It is crucial to point out that our findings could be influenced by the task participants performed while inside the scanner. The participants were required to identify whether the presented image was a bathroom or not. Most of the studies do not look into the influence of tasks in the fMRI studies and the opinion on whether the difference in tasks results in different representations is divided. For instance, some studies [Duncan, 2010; Woolgar et al., 2011] suggest that the parietal and prefrontal cortex are involved in representing task context while the scene processing is attributed to the occipitotemporal cortex. Recently, some studies [Harel et al., 2014; Erez and Duncan, 2015; Lowe et al., 2016; Bracci et al., 2017; Bugatus et al., 2017; Vaziri-Pashkam and Xu, 2017; Hebart et al. 2018] have shown evidence of task influence in the occipitotemporal cortex. Therefore, a promising future direction of research might be to find out whether our findings are replicated or not on another fMRI study where participants performed a different task.

To summarize, our findings provided evidence supporting the use of DNNs trained on the scene parsing task as a promising tool to predict and understand activity in the visual brain. We believe that this approach has the potential to be applied widely, providing interpretable results that give insights into how the human visual cortex represents the visual world.

Acknowledgements

We thank Agnessa Karapetian and Greta Häberle for their valuable comments on the manuscript. G.R. thanks the support of the Alfons and Gertrud Kassel Foundation. R.M.C. is supported by Deutsche Forschungsgemeinschaft (DFG) grants (CI241/1-1, CI241/3-1) and the European Research Council Starting Grant (ERC-2018-StG 803370).

References

David, Bau; Zhou, Bolei; Khosla, Aditya; Oliva, Aude; Torralba, Antonio. Network dissection: Quantifying interpretability of deep visual representations; Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. 6541–6549.

Bonner, Michael F; Epstein, Russell A. Coding of navigational affordances in the human visual system. Proceedings of the National Academy of Sciences. 2017; 114 (18) :4793–4798.

Bonner, Michael F; Epstein, Russell A. Computational mechanisms underlying cortical responses to the affordance properties of visual scenes. PLoS computational biology. 2018; 14 (4) e1006111 [PubMed: 29684011]

Martin, Cichy Radoslaw; Khosla, Aditya; Pantazis, Dimitrios; Torralba, Antonio; Oliva, Aude. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Scientific reports. 2016; 6 27755 [PubMed: 27282108]

Martin, Cichy Radoslaw; Khosla, Aditya; Pantazis, Dimitrios; Oliva, Aude. Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. NeuroImage. 2017; 153 :346–358. [PubMed: 27039703]

Martin, Cichy Radoslaw; Kaiser, Daniel. Deep neural networks as scientific models. Trends in cognitive sciences. 2019; 4 :305–317. DOI: 10.1016/j.tics.2019.01.009

Dilks, Daniel D; Julian, Joshua B; Paunov, Alexander M; Kanwisher, Nancy. The occipital place area is causally and selectively involved in scene perception. Journal of Neuroscience. 2013; 33 (4) :1331–1336. [PubMed: 23345209]

Russell, Epstein; Kanwisher, Nancy. A cortical representation of the local visual environment. Nature. 1998; 392 (6676) :598. [PubMed: 9560155]

Greene, Michelle R; Oliva, Aude. The briefest of glances: The time course of natural scene understanding. Psychological Science. 2009; 20 (4) :464–472. [PubMed: 19399976]

Groen, Iris IA; Greene, Michelle R; Baldassano, Christopher; Fei-Fei, Li; Beck, Diane M; Baker, Chris I. Distinct contributions of functional and deep neural network features to representational similarity of scenes in human brain and behavior. eLife. 2018; 7 e32962 [PubMed: 29513219]

Uri, Hasson; Harel, Michal; Levy, Ifat; Malach, Rafael. Large-scale mirror-symmetry organization of human occipito-temporal object areas. Neuron. 2003; 37 (6) :1027–1041. [PubMed: 12670430]

Kaiming, He; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian. Deep residual learning for image recognition; Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. 770–778.

Hebart, Martin N; Bankson, Brett B; Harel, Assaf; Baker, Chris I; Cichy, Radoslaw M. The representational dynamics of task and object processing in humans. eLife. 2018; 7 e32816 [PubMed: 29384473]

Linda, Henriksson; Mur, Marieke; Kriegeskorte, Nikolaus. Rapid invariant encoding of scene layout in human opa Neuron. 2019

Seyed-Mahdi, Khaligh-Razavi; Kriegeskorte, Nikolaus. Deep supervised, but not unsupervised, models may explain ITit cortical representation. PLoS computational biology. 2014; 10 (11) e1003915 [PubMed: 25375136]

Kravitz, Dwight J; Peng, Cynthia S; Baker, Chris I. Real-world scene representations in high-level visual cortex: it's the spaces more than the places. Journal of Neuroscience. 2011; 31 (20) :7322–7333. [PubMed: 21593316]

Nikolaus, Kriegeskorte; Mur, Marieke; Bandettini, Peter A. Representational similarity analysis-connecting the branches of systems neuroscience. Frontiers in systems neuroscience. 2008; 2 :4. [PubMed: 19104670]

Alex, Krizhevsky; Sutskever, Ilya; Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012 :1097–1105.

Fei-Fei, Li; Iyer, Asha; Koch, Christof; Perona, Pietro. What do we perceive in a glance of a real-world scene? Journal of vision. 2007; 7 (1) :10.

Tsung-Yi, Lin; Dollar, Piotr; Girshick, Ross; Kaiming, He; Hariharan, Bharath; Belongie, Serge. Feature pyramid networks for object detection; Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. 2117–2125.

O'Craven, Kathleen M; Kanwisher, Nancy. Mental imagery of faces and places activates corresponding stimulus-specific brain regions. Journal of cognitive neuroscience. 2000; 12 (6) :1013–1023. [PubMed: 11177421]

Jeongho, Park; Park, Soojin. Conjoint representation of texture ensemble and location in the parahippocampal place area. Journal of neurophysiology. 2017; 117 (4) :1595–1607. [PubMed: 28123006]

Soojin, Park; Brady, Timothy F; Greene, Michelle R; Oliva, Aude. Disentangling scene content from spatial boundary: complementary roles for the parahippocampal place area and lateral occipital complex in representing real-world scenes. Journal of Neuroscience. 2011; 31 (4) :1333–1340. [PubMed: 21273418]

Potter, Mary C. Meaning in visual search. Science. 1975; 187 (4180) :965–966. [PubMed: 1145183]

Richards, Blake A; Lillicrap, Timothy P; Beaudoin, Philippe; Bengio, Yoshua; Bogacz, Rafal; Christensen, Amelia; Clopath, Claudia; Costa, RuiPonte; de Berker, Archy; Ganguli, Surya; , et al. A deep learning framework for neuroscience. Nature neuroscience. 2019; 22 (11) :1761–1770. [PubMed: 31659335]

Russell, Bryan C; Torralba, Antonio; Murphy, Kevin P; Freeman, William T. LabelMe: a database and web-based tool for image annotation. International journal of computer vision. 2008; 77 (1-3) :157–173.

Harry, Silson Edward; Chan, Annie Wai-Yiu; Reynolds, Richard Craig; Kravitz, Dwight Jacob; Baker, Chris Ian. A retinotopic basis for the division of high-level scene processing between lateral and ventral human occipitotemporal cortex. Journal of Neuroscience. 2015; 35 (34) :11921–11935. [PubMed: 26311774]

Ke, Sun; Zhao, Yang; Jiang, Borui; Cheng, Tianheng; Xiao, Bin; Liu, Dong; Yadong, Mu; Wang, Xinggang; Liu, Wenyu; Wang, Jingdong. High-resolution representations for labeling pixels and regions. arXiv preprint. 2019 arXiv:1904.04514

Simon, Thorpe; Fize, Denis; Marlot, Catherine. Speed of processing in the human visual system. Nature. 1996; 381 (6582) :520–522. [PubMed: 8632824]

Vanessa, Troiani; Stigliani, Anthony; Smith, Mary E; Epstein, Russell A. Multiple object properties drive scene-selective regions. Cerebral cortex. 2012; 24 (4) :883–897. [PubMed: 23211209]

Tete, Xiao; Liu, Yingcheng; Zhou, Bolei; Jiang, Yuning; Sun, Jian. Unified perceptual parsing for scene understanding; Proceedings of the European Conference on Computer Vision (ECCV); 2018. 418–434.

Yamins, Daniel LK; Hong, Ha; Cadieu, Charles F; Solomon, Ethan A; Seibert, Darren; DiCarlo, James J. Performance-optimized hierarchical models predict neural responses in higher vi- sual cortex. Proceedings of the National Academy of Sciences. 2014; 111 (23) :8619–8624.

Yamins, Daniel LK; DiCarlo, James J. Using goal-driven deep learning models to understand sensory cortex. Nature neuroscience. 2016; 19 (3) :356. [PubMed: 26906502]

Hengshuang, Zhao; Shi, Jianping; Xiaojuan, Qi; Wang, Xiaogang; Jia, Jiaya. Pyramid scene parsing network; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. 2881–2890.

Bolei, Zhou; Khosla, Aditya; Lapedriza, Agata; Oliva, Aude; Torralba, Antonio. Object detectors emerge in deep scene cnns. arXiv preprint. 2014 arXiv:1412.6856

Bolei, Zhou; Lapedriza, Agata; Khosla, Aditya; Oliva, Aude; Torralba, Antonio. Places: A 10 million image database for scene recognition; IEEE transactions on pattern analysis and machine intelligence; 1452–1464.

Bolei, Zhou; Zhao, Hang; Puig, Xavier; Fidler, Sanja; Barriuso, Adela; Torralba, Antonio. Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision. 127 (3) :302–321.

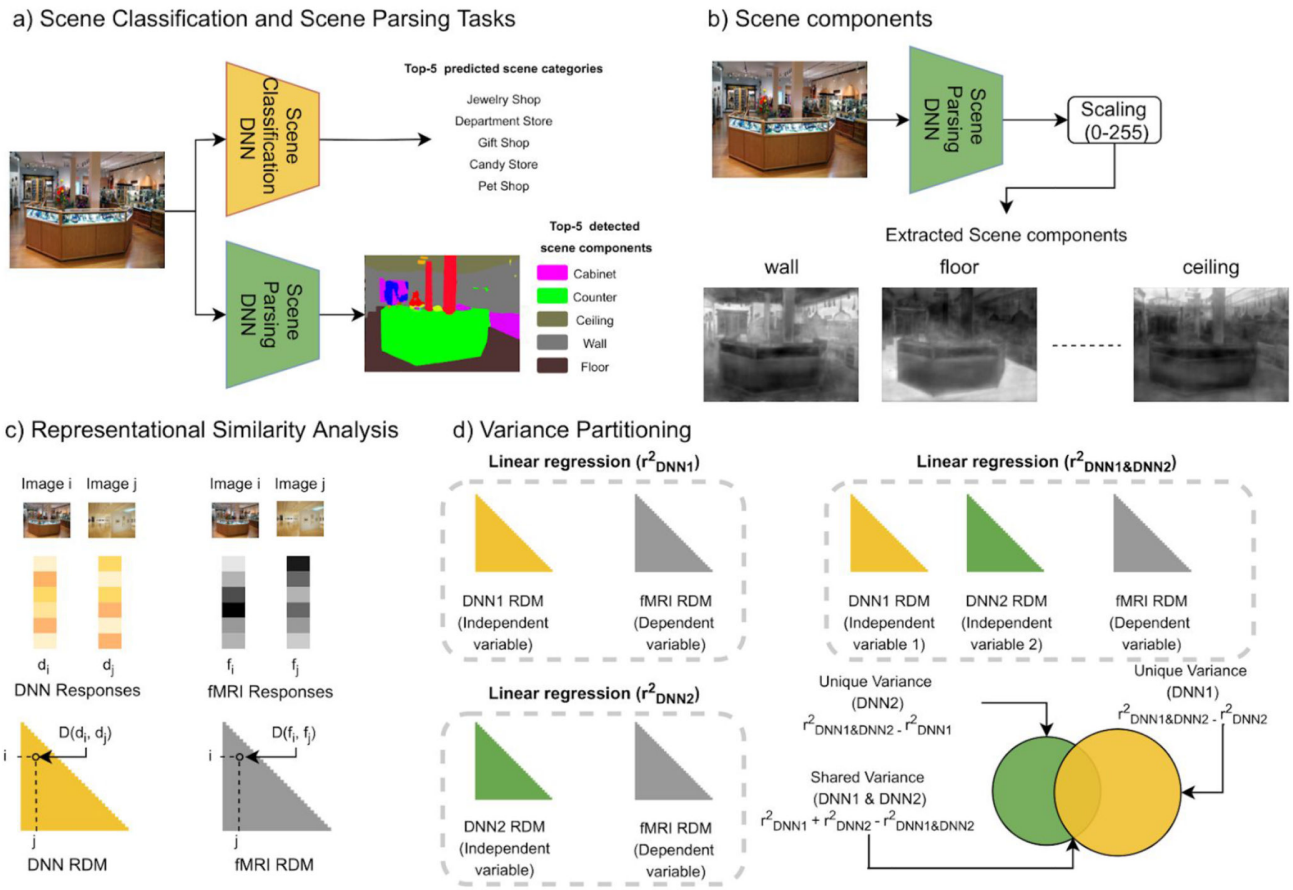

Figure 1. Outline of our approach.

a) In the scene classification task, the model outputs the probability of an image belonging to a particular class. In the scene parsing task, the model outputs a spatial map for each component. The pixel value of the spatial map corresponding to a component represents the probability of that pixel belonging to that component. b) We use DNNs trained on scene parsing to extract responses corresponding to individual scene components. c) RSA: We first compute RDMs for a DNN model and a brain ROI by computing pairwise distance (D) between DNN (d_i,d_j)/fMRI(f_i,f_j) responses corresponding to each pair (i,j) of images in the stimulus set. We next compute the correlation of a DNN RDM with fMRI RDM to determine the similarity between the brain and the DNN. d) Variance Partitioning: We conduct three multiple linear regressions with DNN RDMs as the independent variables and fMRI RDM as the dependent variable to estimate unique and shared variance of fMRI RDM explained by DNN RDMs.

Europe PMC Funders Author Manuscripts

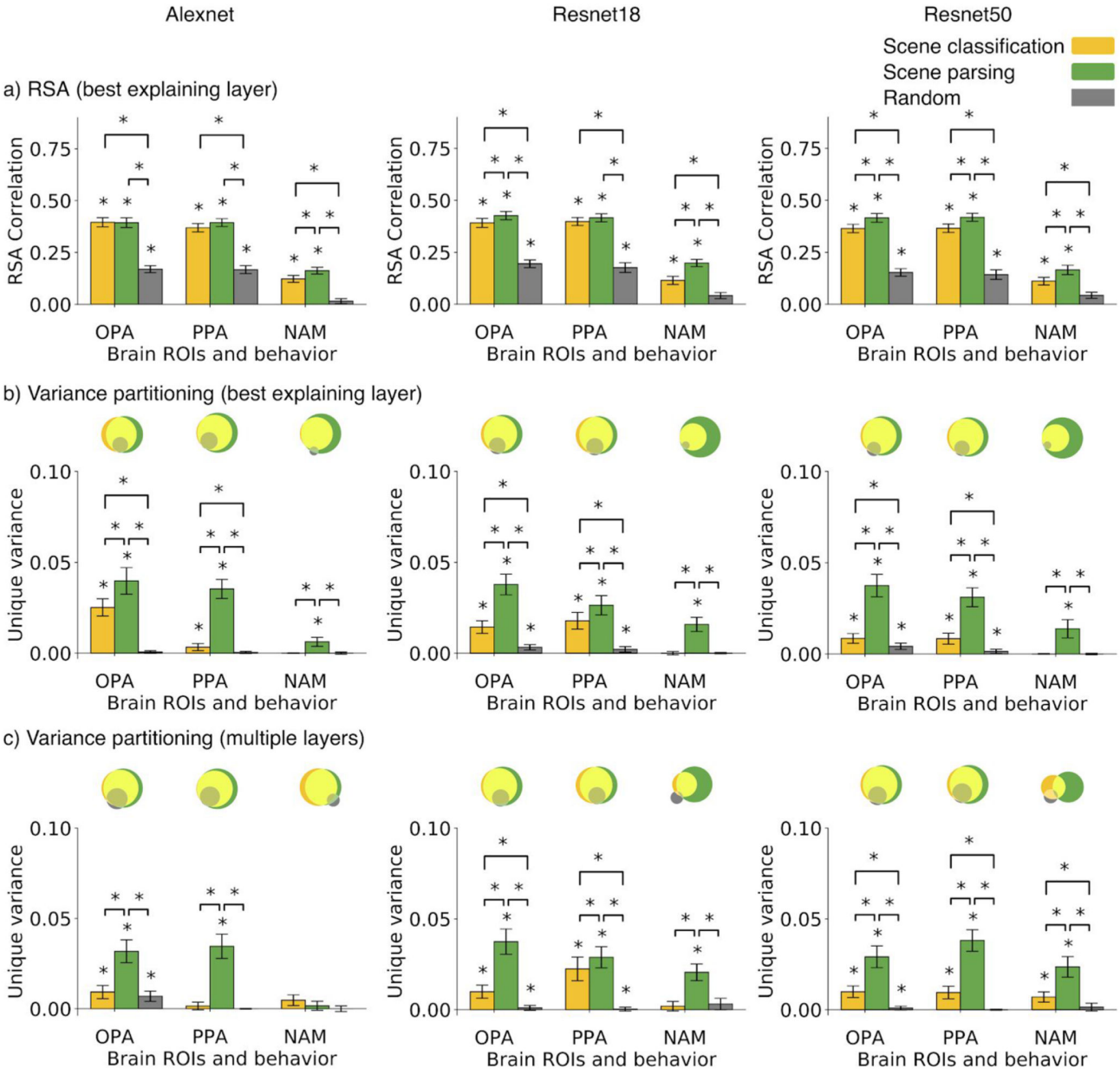

Figure 2. Model comparison in accounting for OPA and PPA as well as behavior.

a) RSA of scene-selective areas PPA, OPA, and behavioral model NAM with scene parsing, and scene-classification, and random models (best-explaining layer), and b) variance of scene-selective areas PPA, OPA, and behavioral model NAM explained uniquely by scene parsing, and scene-classification, and random models (best-explaining layer) for the architecture Alexnet (left), Resnet18 (middle), Resnet50 (right). Venn diagram on top of each bar plot illustrates the unique and shared variance of ROIs and behavior explained by multiple models together. c) Variance of scene-selective areas PPA, OPA, and behavioral model NAM explained uniquely by scene parsing, scene-classification and random models (multiple layers) for the architecture Alexnet (left), Resnet18 (middle), Resnet50 (right).

Venn diagram on top of each bar plot illustrates the unique and shared variance of ROIs and behavior explained by multiple models together. The asterisk at the top indicates the significance ($p<0.05$) calculated by permuting the conditions 5000 times.

Europe PMC Funders Author Manuscripts

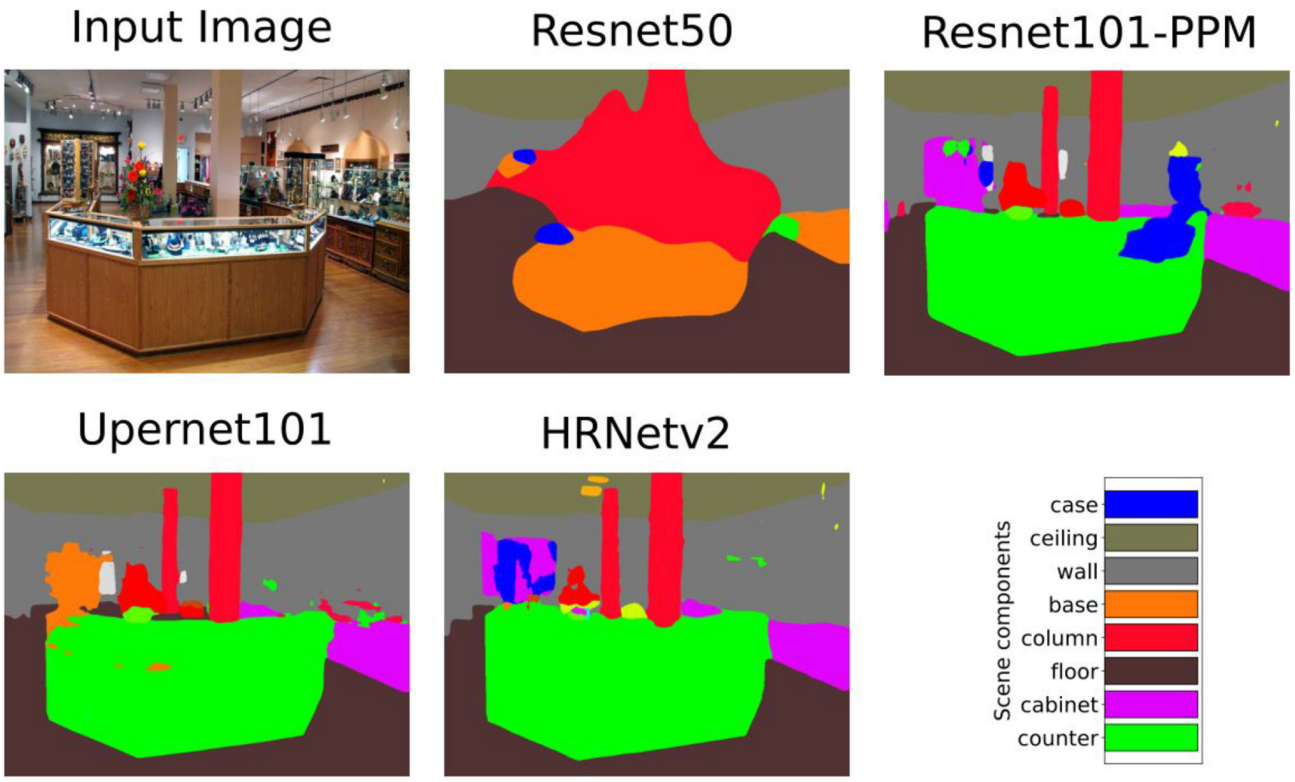

Figure 3. Qualitative comparison of scene parsing output for different models.
Input image (top left) and corresponding scene parsing output of the different models investigated in this work.

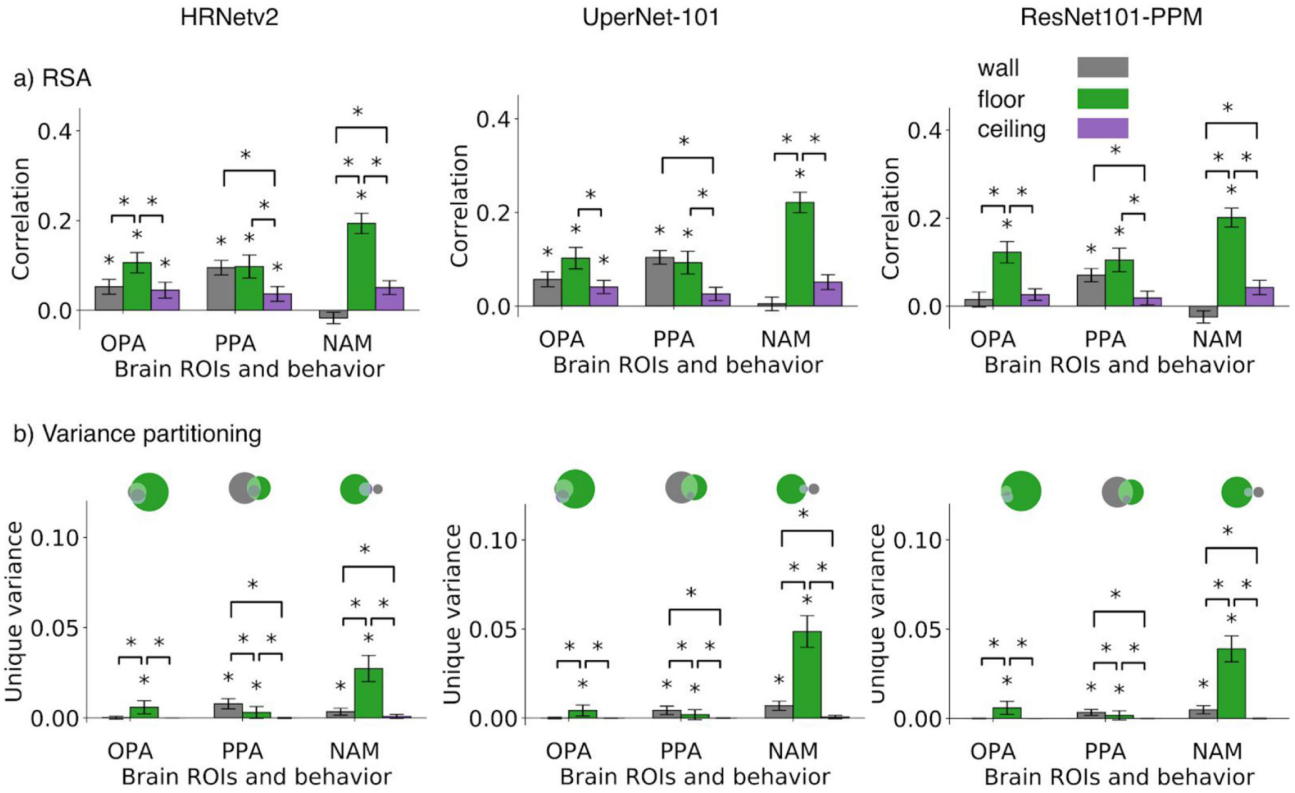

Figure 4. Scene components reveal the differences in representational content of OPA, PPA and behavioral model NAM.

a) Results of RSA for OPA, PPA and NAM with scene components of wall floor and ceiling for 3 state of the art models on scene parsing task HRnetv2 (left), Upernet101 (middle), Resnet101 (right); b) Unique variance accounted for in OPA, PPA and NAM by using components from HRnetv2 (left), Upernet101 (middle), Resnet101 (right) models. Venn diagram on top of each bar plot illustrates the unique and shared variance of ROIs and behavior explained by multiple components together. The asterisk at the top indicates the significance ($p<0.05$) calculated by permuting the conditions 5000 times, FDR corrected with a threshold equal to 0.05.

Europe PMC Funders Author Manuscripts

Table 1

Correlation value and layer information of the layer that showed the highest correlation with a particular brain area or behavior for all the models considered (AlexNet, ResNet18 and ResNet50).

Task	Models	OPA	PPA	NAM
Scene class.	Alexnet	0.395(fc7)	0.369(fc7)	0.122(conv5)
	Resnet18	0.391(fc)	0.397(fc)	0.114(block4)
	Resnet50	0.364(fc)	0.365(fc)	0.111(block4)
Scene parsing	Alexnet	0.393(d2)	0.393(d1)	0.162(conv5)
	Resnet18	0.426(d1)	0.415(d1)	0.198(block4)
	Resnet50	0.415(d1)	0.418(d1)	0.165(d2)
Random	Alexnet	0.169 (fc7)	0.167(fc6)	0.015(fc7)
	Resnet18	0.194(fc)	0.176(fc)	0.041(block4)
	Resnet50	0.152(fc)	0.142(fc)	0.042(block4)

Europe PMC Funders Author Manuscripts

Table 2

Scene parsing performance on ADE20k validation set. The table shows the accuracy of detecting selected components along with overall accuracy for different scene parsing models in decreasing order.

Model	Wall	Ceiling	Floor	Overall accuracy
HRNetv2	0.7538	0.8278	0.7811	0.4320
Upernet101	0.7503	0.8265	0.7772	0.4276
Resnet101-PPM	0.7453	0.8195	0.7659	0.4257
Resnet50	0.6422	0.7356	0.6642	0.3020
Resnet18	0.6223	0.6997	0.6627	0.2741
AlexNet	0.5857	0.6810	0.6105	0.2306

1.2 Unveiling functions of the visual cortex using task-specific deep neural networks

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Unveiling functions of the visual cortex using task-specific deep neural networks

Kshitij Dwivedi[1,2]*, **Michael F. Bonner**[3], **Radoslaw Martin Cichy**[1‡], **Gemma Roig**[2‡]*

1 Department of Education and Psychology, Freie Universität Berlin, Germany, **2** Department of Computer Science, Goethe University, Frankfurt am Main, Germany, **3** Department of Cognitive Science, Johns Hopkins University, Baltimore, Maryland, United States of America

‡ jointly directed work.
* dwivedi@em.uni-frankfurt.de (KD); roig@cs.uni-frankfurt.de (GR)

Citation: Dwivedi K, Bonner MF, Cichy RM, Roig G (2021) Unveiling functions of the visual cortex using task-specific deep neural networks. PLoS Comput Biol 17(8): e1009267. https://doi.org/10.1371/journal.pcbi.1009267

Editor: Ulrik R. Beierholm, Durham University, UNITED KINGDOM

Received: February 9, 2021

Accepted: July 11, 2021

Published: August 13, 2021

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1009267

Data Availability Statement: The authors confirm that all data underlying the findings are fully available without restriction. The data required to reproduce the results is available here: https://osf.

Abstract

The human visual cortex enables visual perception through a cascade of hierarchical computations in cortical regions with distinct functionalities. Here, we introduce an AI-driven approach to discover the functional mapping of the visual cortex. We related human brain responses to scene images measured with functional MRI (fMRI) systematically to a diverse set of deep neural networks (DNNs) optimized to perform different scene perception tasks. We found a structured mapping between DNN tasks and brain regions along the ventral and dorsal visual streams. Low-level visual tasks mapped onto early brain regions, 3-dimensional scene perception tasks mapped onto the dorsal stream, and semantic tasks mapped onto the ventral stream. This mapping was of high fidelity, with more than 60% of the explainable variance in nine key regions being explained. Together, our results provide a novel functional mapping of the human visual cortex and demonstrate the power of the computational approach.

Author summary

Human visual perception is a complex cognitive feat known to be mediated by distinct cortical regions of the brain. However, the exact function of these regions remains unknown, and thus it remains unclear how those regions together orchestrate visual perception. Here, we apply an AI-driven brain mapping approach to reveal visual brain function. This approach integrates multiple artificial deep neural networks trained on a diverse set of functions with functional recordings of the whole human brain. Our results reveal a systematic tiling of visual cortex by mapping regions to particular functions of the deep networks. Together this constitutes a comprehensive account of the functions of the distinct cortical regions of the brain that mediate human visual perception.

1. Introduction

The human visual system transforms incoming light into meaningful representations that underlie perception and guide behavior. This transformation is believed to take place through

io/dj7v2/ The code is available here: https://github.com/cvai-repo/dnn2brain-function.

Funding: G.R. thanks the support of the Alfons and Gertrud Kassel Foundation. R.M.C. is supported by DFG grants (CI241/1-1, CI241/3-1) and the ERC Starting Grant (ERC-2018- StG 803370). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

a cascade of hierarchical processes implemented in a set of brain regions along the so-called ventral and dorsal visual streams [1]. Each of these regions has been stipulated to fulfill a distinct sub-function in enabling perception [2]. However, discovering the exact nature of these functions and providing computational models that implement them has proven challenging. Recently, computational modeling using deep neural networks (DNNs) has emerged as a promising approach to model, and predict neural responses in visual regions [3–7]. These studies have provided a first functional mapping of the visual brain. However, the resulting account of visual cortex functions has remained incomplete. This is so because previous studies either explain the function of a single or few candidate regions by investigating many DNNs or explain many brain regions comparing it to a single DNN trained on one task only (usually object categorization). In contrast, for a systematic and comprehensive picture of human brain function that does justice to the richness of the functions that each of its subcomponents implements, DNNs trained on multiple tasks, i.e., functions, must be related and compared in their predictive power across the whole cortex.

Aiming for this systematic and comprehensive picture for the visual cortex we here relate brain responses across the whole visual brain to a wide set of DNNs, in which each DNN is optimized for a different visual task, and hence, performs a different function.

To reliably reveal the functions of brain regions using DNNs performing different functions, we need to ensure that only function and no other crucial factor differs between the DNNs. The parameters learned by a DNN depend on a few fundamental factors, namely, its architecture, training dataset, learning mechanism, and the function the DNN was optimized for. Therefore, in this study, we select a set of DNNs [8] that have an identical encoder architecture and are trained using the same learning mechanism and the same set of training images. Thus, the parameters learned by the encoder of the selected DNNs differ only due to their different functions.

We generate a functional map of the visual cortex by comparing the fMRI responses to scene images [9] with the activations of multiple DNNs optimized on different tasks [8] related to scene perception, e.g., scene classification, depth estimation, and edge detection. Our key result is that different regions in the brain are better explained by DNNs performing different tasks, suggesting different computational roles in these regions. In particular, we find that early regions of the visual cortex are better explained by DNNs performing low-level vision tasks, such as edge detection. Regions in the dorsal stream are better explained by DNNs performing tasks related to 3-dimensional (3D) scene perception, such as occlusion detection and surface normal prediction. Regions in the ventral stream are best explained by DNNs performing tasks related to semantics, such as scene classification. Importantly, the top-3 best predicting DNNs explain more than 60% of the explainable variance in nine ventral-temporal and dorsal-lateral visual regions, demonstrating the quantitative power and potential of our AI-driven approach for discovering fine-grained functional maps of the human brain.

2. Results

2.1 Functional map of visual cortex using multiple DNNs

Our primary goal is to generate a functional map of the visual brain in terms of the functions each of the regions implements. Our approach is to relate brain responses to activations of DNNs performing different functions. For this, we used an fMRI dataset recorded while human subjects (N = 16) viewed indoor scenes [9] and performed a categorization task; and a set of 18 DNNs [8] optimized to perform 18 different functions related to visual perception (some of the tasks can be visualized here: https://sites.google.com/view/dnn2brainfunction/home#h.u0nqne179ys2) plus an additional DNN with random weights as a baseline. The

different DNNs' functions were associated with indoor scene perception, covering a broad range of tasks from low-level visual tasks, (e.g., edge detection) to 3-dimensional visual perception tasks (e.g., surface normals prediction) to categorical tasks (e.g., scene classification). Each DNN consisted of an encoder-decoder architecture, where the encoder had an identical architecture across tasks, and the decoder varied depending on the task. To ensure that the differences in variance of fMRI responses explained by different DNNs from our set were not due to differences in architecture, we selected the activations from the last two layers of the identical encoder architecture for all DNNs.

The layer selection was based on an analysis finding the most task-specific layers of the encoder (see S1 Text and S2 Fig). Furthermore, all DNNs were optimized using the same set of training images, and the same backpropagation algorithm for learning. Hence, any differences in our findings across DNNs cannot be attributed to the training data statistics, architecture, or learning algorithm, but to the task for which each DNN was optimized.

To compare fMRI responses with DNNs, we first extracted fMRI responses in a spatially delimited portion of the brain for all images in the stimulus set (Fig 1A). This could be either a group of spatially contiguous voxels for searchlight analysis [10–12] or voxels confined to a particular brain region as defined by a brain atlas for a region-of-interest (ROI) analysis. Equivalently, we extracted activations from the encoders of each DNN for the same stimulus set.

We then used Representational Similarity Analysis (RSA) [13] to compare brain activations with DNN activations. RSA defines a similarity space as an abstraction of the incommensurable multivariate spaces of the brain and DNN activation patterns. This similarity space is defined by pairwise distances between the activation patterns of the same source space, either fMRI responses from a brain region or DNN activations, where responses can be directly related. For this, we compared all combinations of stimulus-specific activation patterns in each source space (i.e., DNN activations, fMRI activations). Then, the results for each source space were noted in a two-dimensional matrix, called representational dissimilarity matrices (RDMs). The rows and columns of RDMs represent the conditions compared. To relate fMRI and DNNs in this RDM-based similarity space we performed multiple linear regression predicting fMRI RDM from DNN RDMs of the last two encoder layers. We obtained the adjusted coefficient of determination R^2 (referred to as R^2 in the subsequent text) from the regression to quantify the similarity between the fMRI responses and the DNN (Fig 1B). We performed this analysis for each of the 18 DNNs investigated, which we group into 2D, 3D, or semantic DNNs when those are optimized for 2D, 3D, or semantic tasks, respectively, and an additional DNN with random weights as a baseline. The tasks were categorized into three groups (2D, 3D, and semantic) based on different levels of indoor scene perception and were verified in previous works using transfer performance using one DNN as the initialization to other target tasks [8] and representational similarity between DNNs [14]. We finally used the obtained DNN rankings based on R^2 to identify the DNNs with the highest R^2 for fMRI responses in that brain region (Fig 1C top). To visualize the results, we color-coded the brain region by color indexing the DNN showing the highest R^2 in that brain region (Fig 1C bottom).

To generate a functional map across the whole visual cortex we performed a searchlight analysis [11,12]. In detail, we obtain the R^2-based DNN rankings on the local activation patterns around a given voxel, as described above. We conducted the above analysis for each voxel, resulting in a spatially unbiased functional map.

We observed that different regions of the visual cortex showed the highest similarity with different DNNs. Importantly, the pattern with which different DNNs predicted brain activity best was not random but spatially organized: 2D DNNs (in shades of blue in Fig 1D; interactive map visualization available here: https://sites.google.com/view/dnn2brainfunction/home#h.ub1chq1k42n6) show a higher similarity with early visual regions, 3D DNNs (in shades of

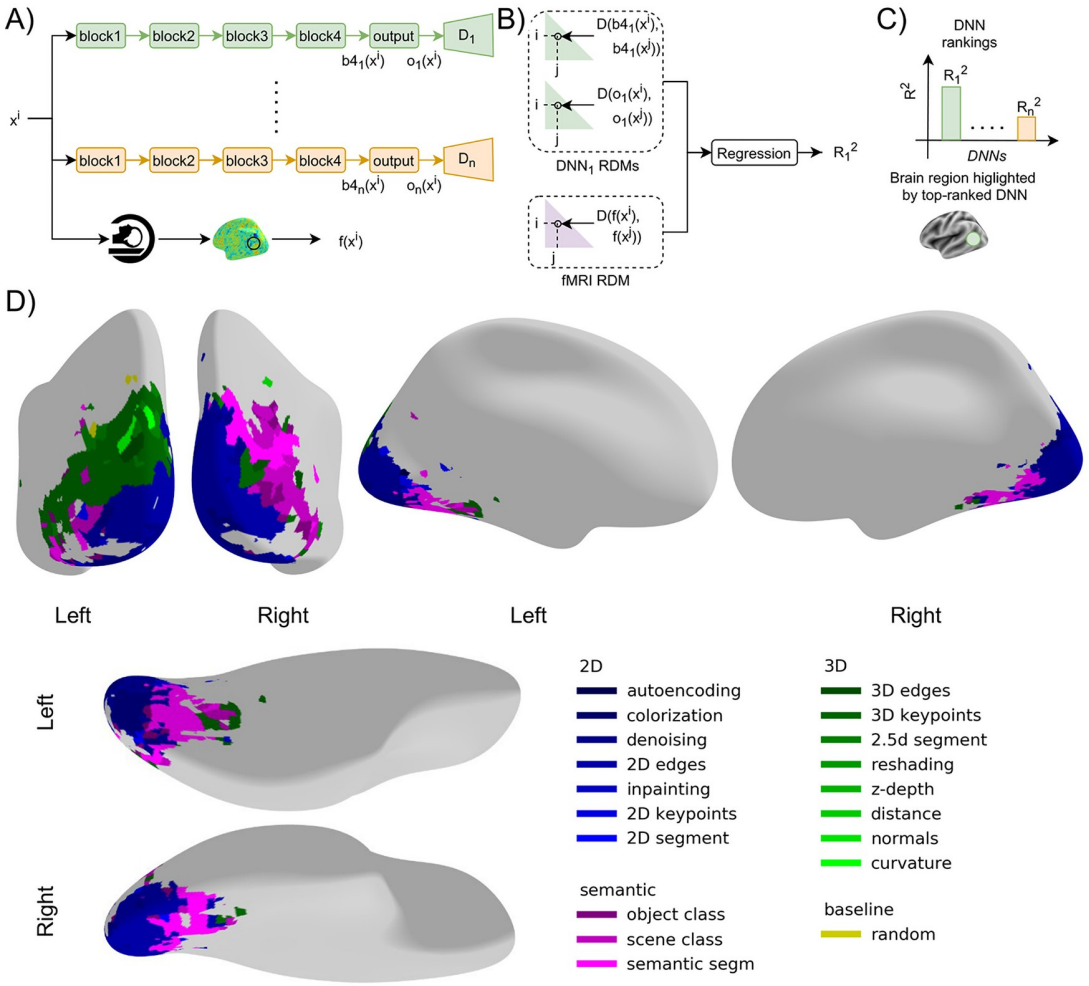

Fig 1. Methods and results of functional mapping of the visual cortex by task-specific DNNs. A) Schema of DNN-fMRI comparison. As a first step, we extracted DNN activations from the last two layers (block 4 and output) of the encoders, denoted as $b4_1(x^i)$, $o_1(x^i)$ for DNN_1 and $b4_n(x^i)$, $o_n(x^i)$ for DNN_n in the figure, from n DNNs and the fMRI response of a region $f(x^i)$ for the i^{th} image x^i in the stimulus set. We repeated the above procedure for all the images in the stimulus set. **B**) We used the extracted activations to compute the RDMs, two for the two DNN layers and one for the brain region. Each RDM contains the pairwise dissimilarities of the DNN activations or brain region activations, respectively. We then used multiple linear regression to obtain an R_1^2 score to quantify the similarity between DNN_1 and the brain region. We repeated the same procedure using other DNNs to obtain corresponding R^2 **C**) We obtained a ranking based on R^2 to identify the DNNs with the highest R^2 for fMRI responses in that brain region. To visualize the results, we color-coded the brain region by the color indexing the DNN showing the highest R^2 in that brain region. **D**) Functional map of the visual brain generated through a spatially unbiased searchlight procedure, comparing 18 DNNs optimized for different tasks and a randomly initialized DNN as a baseline. We show the results for the voxels with significant noise ceiling and R^2 with DNN ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected). An interactive visualization of the functional brain map is available in this weblink (https://sites.google.com/view/dnn2brainfunction/home#h.ub1chq1k42n6).

https://doi.org/10.1371/journal.pcbi.1009267.g001

green) show a higher similarity with dorsal regions, while semantic DNNs (in shades of magenta) show a higher similarity with ventral regions and some dorsal regions.

Together, the results of our AI-driven mapping procedure suggest that early visual regions perform functions related to low-level vision, dorsal regions perform functions related to both 3D and semantic perception, and ventral regions perform functions related to semantic perception.

2.2 Nature and predictive power of the functional map

Using the searchlight results from Fig 1D, we identified the DNN that showed the highest R^2 for each searchlight. This poses two crucial questions that require further investigation for an in-depth understanding of the functions of brain regions. Firstly, does a single DNN prominently predict a region's response (one DNN-to-one region) or a group of DNNs together predict its response (many DNNs-to-one region)? A one-to-one mapping between DNN and a region would suggest a single functional role while a many-to-one mapping would suggest multiple functional roles of the brain region under investigation. Secondly, given that the DNNs considered in this study predict fMRI responses, how well do they predict on a quantitative scale? A high prediction accuracy would suggest that the functional mapping obtained using our analysis is accurate, while a low prediction accuracy would suggest that DNNs considered in this study are not suitable to find the function of that brain region. Although it is possible to answer the above questions for each voxel, for conciseness we consider 25 regions of interest (ROIs) tiling the visual cortex from a brain atlas [15].

To determine how accurately DNNs predict fMRI responses, we calculated the lower and upper bound of the noise ceiling for each ROI. We included ROIs (15 out of 25) with a lower noise ceiling above 0.1 and discarded other ROIs due to low signal-to-noise ratio. We show the locations of the investigated ROIs in the visual cortex in Fig 2A.

For each ROI we used RSA to compare fMRI responses (transformed into fMRI RDMs) with activations of all 18 DNNs plus a randomly initialized DNN as a baseline (transformed into DNN RDMs). This yielded one R^2 value for each DNN per region (see S3 Fig). We then selected the top-3 DNNs showing the highest R^2 and performed a variance partitioning analysis [16]. We used the top-3 DNN RDMs as the independent variable and the ROI RDM as the dependent variable to find out how much variance of ROI responses is explained uniquely by each of these DNNs while considered together with the other two DNNs. Using the variance partitioning analysis (method illustrated in S1 Fig) we were able to infer the amount of unique and shared variance between different predictors (DNN RDMs) by comparing the explained variance (R^2) of a DNN used alone with the explained variance when it was used with other DNNs. Variance partitioning analysis (Fig 2B) using the top-3 DNNs revealed the individual DNNs that explained the most variance uniquely for a given ROI along with the unique and shared variance explained by other DNNs. The DNN that detects edges explained significantly higher variance ($p<0.05$, permutation test, FDR corrected across DNNs) in ROIs in early and mid-level visual regions (V1v, V1d, V2v, V2d, V3v, and hV4) uniquely than the other two DNNs, suggesting a function related to edge detection. Semantic segmentation DNN explained significantly higher unique variance in ventral ROIs VO1 and VO2, suggesting a function related to the perceptual grouping of objects. 3D DNNs (3D Keypoints, 2.5D Segmentation, 3D edges, curvature) were best predicting DNNs for dorsal ROIs V3d and V3b suggesting their role in 3D scene understanding. A combination of 3D and semantic DNNs were best predicting DNNs for other ROIs (PHC1, PHC2, LO1, LO2, and V3a). It is crucial to note that if two DNNs from the same task group are in the top-3 best predicting DNNs for an ROI, the unique variance of ROI RDM explained by DNNs in the same group will generally be lower than by DNN not in the group. We have observed that DNNs in the same task group show a higher correlation with each other as compared to DNNs in other task groups [14]. A higher correlation between the DNNs of the same task group leads to an increase in shared variance and reduces the unique variance of the ROI RDM explained by within task group DNNs. For instance, we can observe this in PHC2 (also in PHC1, V3a), where two semantic DNNs explain less unique variance than a 3D DNN. Therefore, in such cases, we restrain from interpreting that one type of DNN is significantly better than others.

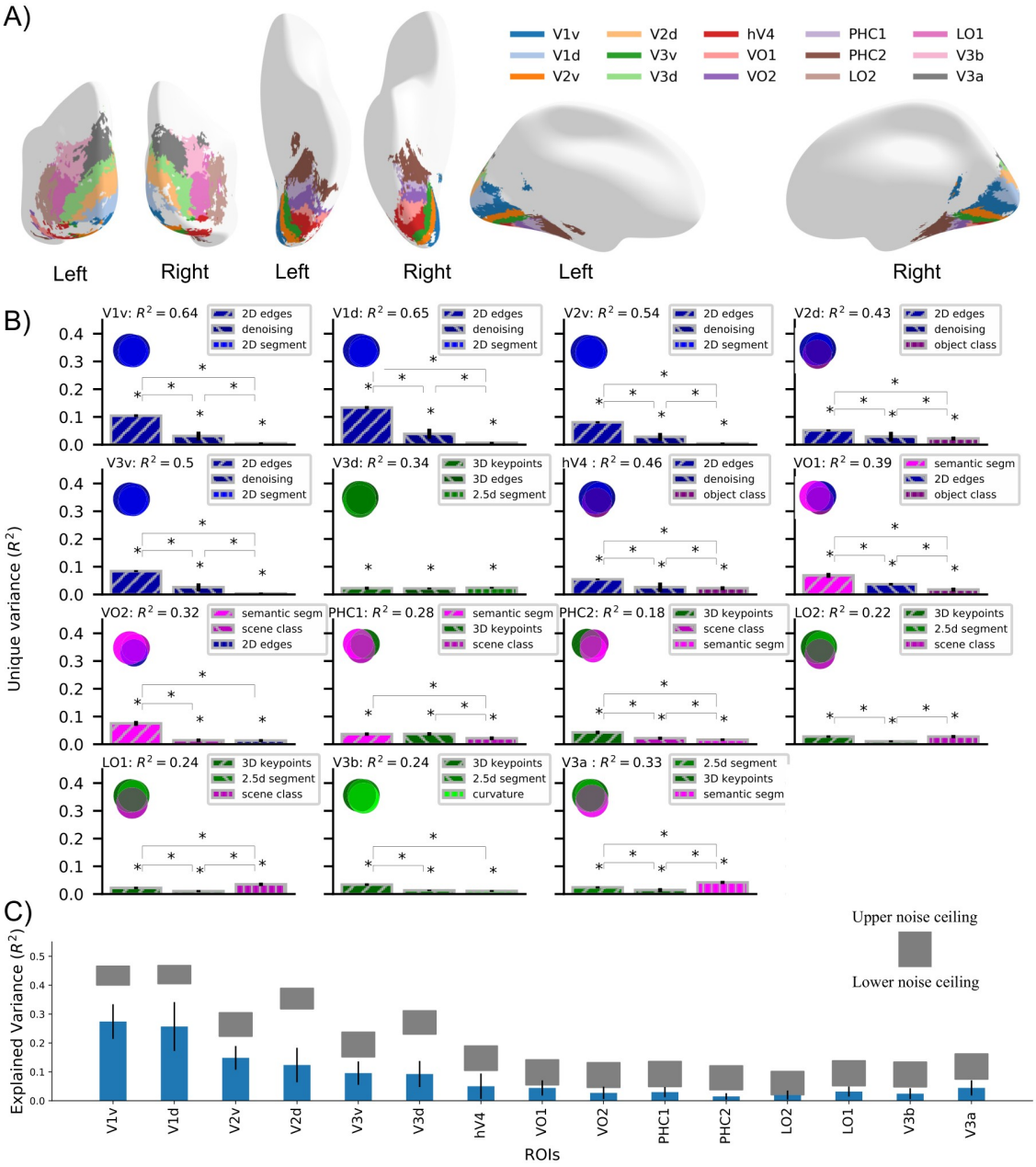

Fig 2. Nature and predictive power of the functional map. A) Cortical overlay showing locations of selected cortical regions from the probabilistic atlas used. **B)** Absolute total variance (R^2) explained in 15 ROIs by using the top-3 DNNs together. The Venn diagram for each ROI illustrates the unique and shared variance of the ROI responses explained by the combination of the top-3 DNNs. The bar plot shows the unique variance of each ROI explained by each of the top-3 DNNs individually. The asterisk denotes the significance of unique variance and the difference in unique variance ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs). The error bars show the standard deviation calculated by bootstrapping 90% of the conditions (10,000 iterations). **C)** Variance of each ROI explained by top-3 best predicting DNNs (cross validated across subjects and conditions) indicated in blue bars compared with lower and upper bound of noise ceiling indicated by shaded gray region. The error bars show the 95% confidence interval calculated across N = 16 subjects. All the R^2 values are statistically significant ($p<0.05$, two-sided t-test, FDR-corrected across ROIs).

https://doi.org/10.1371/journal.pcbi.1009267.g002

Overall, we observed a many-to-one relationship between function and region for multiple regions, i.e., multiple DNNs explained jointly a particular brain region. In early and mid-level regions (V1v, V1d, V2v, V3v) the most predictive functions were related to low-level vision (2D edges, denoising, and 2D segmentation). In dorsal regions V3d and V3b, the most predictive functions were related to 3D scene understanding. In later ventral and dorsal regions (V2d, hV4, VO1, VO2, PHC1, PHC2, LO1, LO2, and V3a) we observed a mixed mapping of 2D, 3D, and semantic functions suggesting multiple functional roles of these ROIs. The predictability of a region's responses by multiple DNNs demonstrates that a visual region in the brain has representations well suited for distinct functions. A plausible conjecture of the above findings is that these regions might be performing a function related to the best predicting DNNs but is not present in the set of DNNs investigated in this study.

To determine the accuracy of the functional mapping of the above ROIs, we calculated the percentage of the explainable variance explained by the top-3 best predicting DNNs. We calculated the explained variance by best predicting DNNs using cross-validation across subjects (N-fold) and conditions (two-fold). As we use multiple models together for multiple linear regression, we need to cross-validate using different sets of RDMs for fitting and evaluating the fit of the regression. Here, we perform cross-validation across subjects by fitting the regression on one-subject-left-out subject-averaged RDMs on half of the images in the stimulus set and evaluating on the left-out single subject RDM on the other half of the images. The above method is a stricter evaluation criterion as compared to the commonly used one without cross-validation (See S5 Fig). We compared the variance explained by the top-3 DNNs with the lower estimate of the noise ceiling which is an estimate of the explainable variance. We found that variance explained in nine ROIs (V1v, V1d, V2v, V3v, VO1, PHC1, LO2, LO1, V3a) is higher than 60% of the lower bound of noise ceiling (Fig 2C, absolute R^2 = 0.085 ± 0.046). In absolute terms, the minimum, median, and maximum cross-validated R^2 values across the 15 ROIs were 0.014 (PHC2), 0.044 (VO1), and 0.27 (V1v) which are comparable to related studies [17] performing evaluation in a similar manner. This shows that the DNNs selected in this study predict fMRI responses well and therefore are suitable for mapping the functions of the investigated ROIs.

In sum, we demonstrated that in many regions of the visual cortex, DNNs trained on different functions predicted activity. This suggests that these ROIs have multiple functional roles. We further showed quantitatively that more than 60% of the explainable variance in nine visual ROIs is explained by the set of DNNs we used, demonstrating that the selected DNNs are well suited to investigate the functional roles of these ROIs.

2.3 Functional map of visual cortex through 2D, 3D, and semantic tasks

In the previous section, we observed a pattern qualitatively suggesting different functional roles of early (2D), dorsal (3D and semantic), and ventral (semantic) regions in the visual cortex. To quantitatively assess this, we investigated the relation of brain responses and DNNs not at the level of single tasks, but task groups (2D, 3D, and semantic), where DNNs belonging to a task group showed a higher correlation with other DNNs in the group than with DNNs in other task groups (see S1 Text).

We averaged the RDMs of DNNs in each task group to obtain aggregate 2D, 3D, and semantic RDMs. Averaging the RDMs based on task groups reduced the number of DNN comparisons from 18 to 3. This allowed us to perform variance partitioning analysis to compare fMRI and DNN RDMs, which would be impractical with 18 single DNNs due to a large number of comparisons and computational complexity. When used in this way, variance

partitioning analysis reveals whether and where in the brain one task group explained brain responses significantly better than other task groups.

We first performed a searchlight analysis to identify where in the cortex one task group explains significantly higher variance uniquely than the other task groups. We selected the grouped DNN RDM that explains the highest variance in a given region uniquely to create a functional map of the task groups in the visual cortex (Fig 3A). Here, due to the reduced number of comparisons, we can clearly observe distinctions where one grouped DNN explains fMRI responses better than the other grouped DNNs ($p<0.05$, permutation test with 10,000

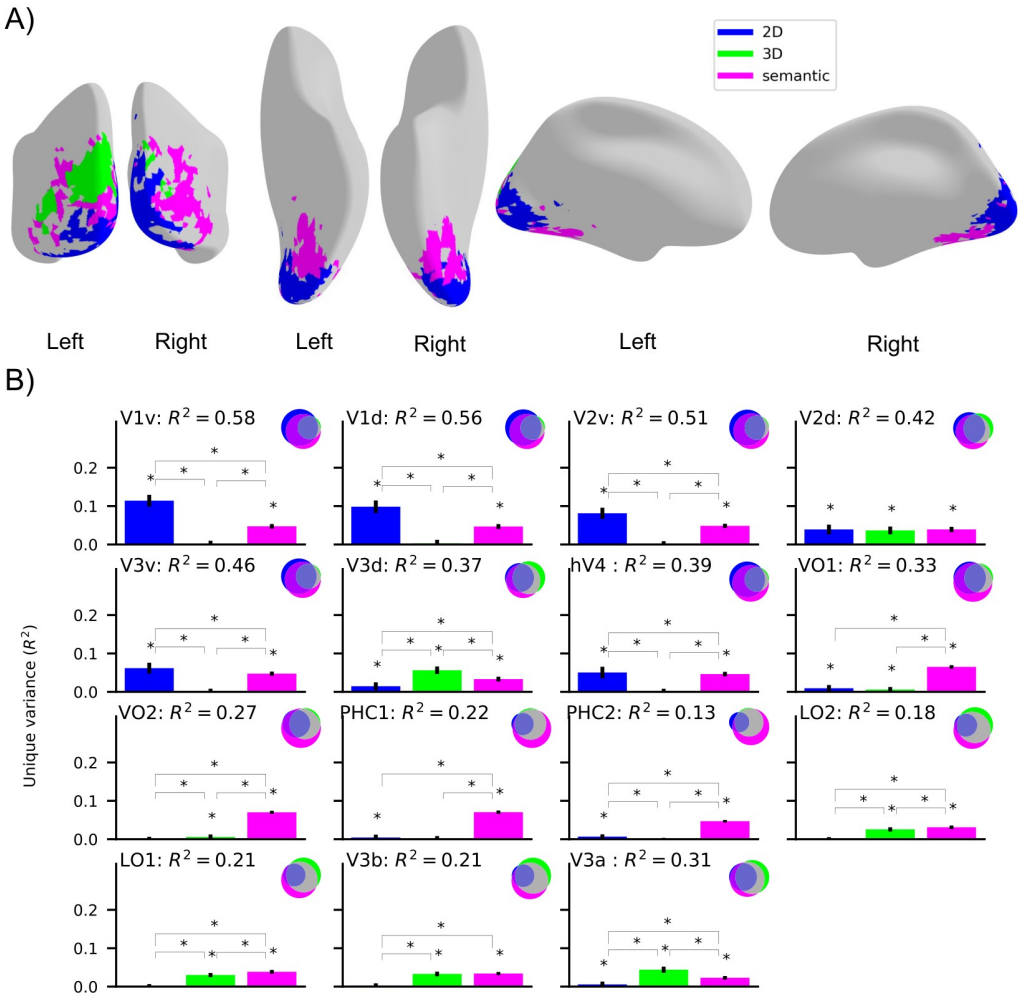

Fig 3. Functional mapping of the visual cortex with respect to 2D, 3D, and semantic tasks. A) Functional map of the visual cortex showing the regions where unique variance explained by one DNN group (2D, 3D, or semantic) is significantly higher than the variance explained by the other two DNN groups ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected). We show the results for the voxels with a significant noise ceiling that show significantly higher unique variance for one DNN group than other two DNN groups ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs and searchlights). The functional brain map can be visualized in this weblink (https://sites.google.com/view/dnn2brainfunction/home#h.xi402x2hr0p3). **B)** Absolute variance (R^2) explained in 15 ROIs by using 3 DNN RDMs averaged across task groups (2D, 3D, or semantic). The Venn diagram for each ROI illustrates the unique and shared variance of the ROI responses explained by the combination of 3 task groups. The bar plot shows the unique variance of each ROI explained by each task group individually. The asterisk denotes whether the unique variance or the difference in unique variance was significant ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs). The error bars show the standard deviation calculated by bootstrapping 90% of the conditions (10,000 iterations).

https://doi.org/10.1371/journal.pcbi.1009267.g003

iterations, FDR corrected across DNNs and searchlights). The resulting functional map (Fig 3A; interactive visualization available in this link: https://sites.google.com/view/dnn2brainfunction/home#h.xi402x2hr0p3) is different from the functional map in Fig 1D in two ways. First, in the functional map here we highlight the searchlight where one DNN group explained significantly higher variance uniquely than the other 2 DNN groups. In the functional map of Fig 1D, we highlighted the DNN that explained the highest variance of a searchlight without performing any statistical analysis whether the selected DNN was significantly better than the second best DNN or not due to the higher number of comparisons. Second, here we compared functions using groups of DNNs (3 functions: 2D, 3D and semantic), whereas in the previous analysis we compared functions using single DNNs (18 functions). The comparison using groups of DNNs allows us to put our findings in context with previous neuroimaging findings that are typically reported at this level.

We observed that the 2D DNN RDM explained responses in the early visual cortex, semantic DNN RDM explained responses in the ventral visual stream, and some parts in the right hemisphere of the dorsal visual stream, and 3D DNN RDM explained responses in the left hemisphere of the dorsal visual stream. The above findings quantitatively reinforce our qualitative findings from the previous section that early visual regions perform functions related to low-level vision, dorsal regions perform functions related to both 3D and semantic perception, and ventral regions perform functions related to semantic perception.

While the map of the brain reveals the most likely function of a given region, to find out whether a region can have multiple functional roles we need to visualize the variance explained by other grouped DNN RDMs along with the best predicting DNN RDM. To achieve that, we performed a variance partitioning analysis using 3 grouped DNN RDMs as the independent variable and 15 ROIs in the ventral-temporal and the dorsal-ventral stream as the dependent variable. The results in Fig 3B show the unique and shared variance explained by group-level DNN RDMs (2D, 3D, and semantic) for all the 15 ROIs.

From Fig 3B we observed that the responses in early ROIs (V1v, V1d, V2v, V3v, hV4) are explained significantly higher ($p<0.05$, permutation test with 10,000 iterations, FDR corrected across DNNs) by 2D DNN RDM uniquely, while responses in later ventral-temporal ROIs (VO1, VO2, PHC1, and PHC2) are explained by semantic DNN RDM uniquely. In dorsal-lateral ROIs (V3a, V3d) responses are explained by 3D RDM uniquely. In LO1, LO2, and V3b 3D and semantic DNN RDMs explained significant variance uniquely while in V2d all 2D, 3D, and semantic DNN RDMs explained significant unique variance. It is crucial to note that for the ROI analysis here we use grouped DNN RDMs as compared to Fig 2B where we selected top-3 single DNNs that showed the highest R^2 with a given ROI. The comparison with grouped DNN RDMs provides a holistic view of the functional role of ROIs which might be missed if one of the DNNs that is related to the functional role of a ROI is not in the top-3 DNNs (as analyzed in Fig 2B). For instance, in Fig 3B the results suggest both 3D and semantic functional roles of V3b which is not evident from Fig 2B where the top 3-DNNs were all optimized on 3D tasks.

Together, we found that the functional role of the early visual cortex is related to low-level visual tasks (2D), the dorsal stream is related to tasks involved in 3-dimensional perception and categorical understanding of the scene (3D and semantic), and in the ventral stream is related to the categorical understanding of the scene (semantic).

2.4 Functional roles of scene-selective regions

In the previous sections, we focused on discovering functions of regions anatomically defined by an atlas. Since the stimulus set used to record fMRI responses consisted of indoor scenes, in

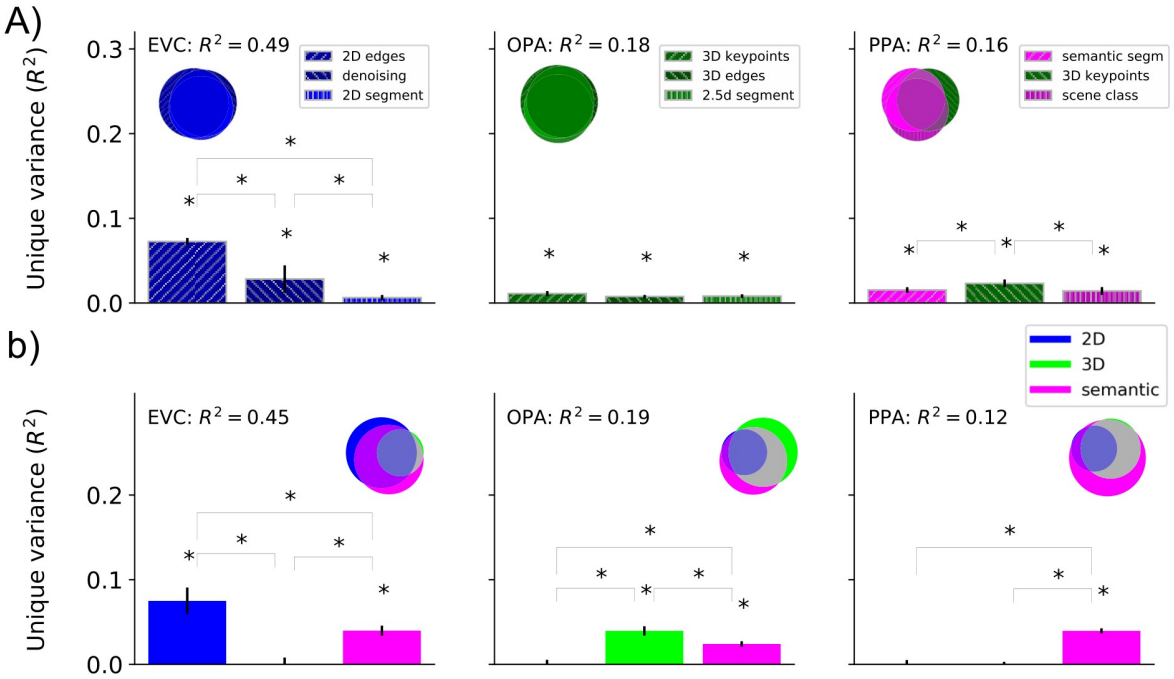

Fig 4. Functional roles of localized ROIs. A) Absolute total variance (R^2) explained in functionally localized ROIs by using the top-3 DNNs together. The Venn diagram for each ROI illustrates the unique and shared variance of the ROI responses explained by the combination of the top-3 DNNs. The bar plot shows the unique variance of each ROI explained by each of the top-3 DNNs individually. The asterisk denotes the significance of unique variance and the difference in unique variance ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs). The error bars show the standard deviation calculated by bootstrapping 90% of the conditions (10,000 iterations). **B)** Absolute total variance (R^2) explained in functionally localized ROIs by using 3 DNN RDMs averaged across task groups (2D, 3D, or semantic). The Venn diagram for each ROI illustrates the unique and shared variance of the ROI responses explained by the combination of 3 DNN task groups. The bar plot shows the unique variance of each ROI explained by each task group individually. The asterisk denotes whether the unique variance or the difference in unique variance was significant ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs). The error bars show the standard deviation calculated by bootstrapping 90% of the conditions (10,000 iterations).

https://doi.org/10.1371/journal.pcbi.1009267.g004

this section we investigate functional differences in functionally localized scene-selective regions. We here focus on two major scene-selective ROIs: occipital place area (OPA) and parahippocampal place area (PPA), putting results into context with the early visual cortex (EVC) as an informative contrast region involved in basic visual processing. The analysis followed the general rationale as used before.

We first investigated the functional differences in these regions by performing variance partitioning analysis using top-3 DNNs (see R^2 based ranking of all DNNs in S4 Fig) that best explained a given ROIs' responses (Fig 4A). We found that the DNN that detects edges explained significantly higher variance ($p<0.05$, permutation test, FDR-corrected) in EVC uniquely than the other two DNNs, suggesting a function related to edge detection. 3D DNNs (3D Keypoints, 2.5D Segmentation, 3D edges) were best predicting DNNs for OPA suggesting its role in 3D scene understanding. A combination of semantic (semantic segmentation, scene classification) and 3D (3D keypoints) DNNs were best predicting DNNs for PPA suggesting its role in both semantic and 3D scene understanding.

We then investigated the functional differences by performing variance partitioning analysis using aggregated 2D, 3D, and semantic DNN RDMs obtained by averaging the individual DNN RDMs in each task group (Fig 4B). We found that for EVC and OPA results are highly consistent with top-3 DNN analysis showing a prominent unique variance explained by the

2D DNN RDM in EVC and the 3D DNN RDM in OPA. Interestingly, in PPA we find that the semantic DNN RDM shows the highest unique variance with no significant unique variance explained by the 3D DNN RDM. The insignificant unique variance explained by the 3D DNN RDM is potentially due to averaging the DNN RDMs of all 3D DNNs (high ranked as well as low ranked) which may lead to diminishing the contribution of an individual high ranked 3D DNN RDM (e.g. 3D keypoints that was in top-3 DNNs for PPA). Overall, we find converging evidence that OPA is mainly related to tasks involved in 3-dimensional perception (3D), and PPA is mainly related to semantic (categorical) understanding of the scene.

3. Discussion

In this study, we harvested the potential of discovering functions of the brain from comparison to DNNs by investigating a large set of DNNs optimized to perform a set of diverse visual tasks. We found a systematic mapping between cortical regions and function: different cortical regions were explained by DNNs performing different functions. Importantly, the selected DNNs explained 60% of the explainable variance in nine out of 15 visual ROIs investigated, demonstrating the accuracy of the AI-driven functional mapping obtained using our analysis.

Our study provides a systematic and comprehensive picture of human brain functions using DNNs trained on different tasks. Previous studies [3–7,17–24] have compared model performance in explaining brain activity, but were limited to a few preselected regions and models, or had a different goal (comparing task structure) [25]. Using the same fMRI dataset as used in this study, a previous study [18] showed that representation in scene-selective ROIs consists of both location and category information using scene-parsing DNNs. We go beyond these efforts by comparing fMRI responses across the whole visual brain using a larger set of DNNs, providing a comprehensive account of the function of human visual brain regions.

We obtained the functional mapping of different regions in the visual cortex on both individual (e.g., 2D edges, scene classification, surface normals, etc.) and group (2D, 3D, semantic) levels of visual functions. We discuss the novel insights gained at the level of individual functions that inform about the fine-grained functional role of cortical regions.

First, we consider 2D DNNs, where the denoising DNN explained significant unique variance in V1v, V1d, V2v, V2d, V3v, and hV4. The denoising task requires the DNN to reconstruct an unperturbed input image from slightly perturbed (e.g., adding Gaussian noise in the current case) input image that encourages learning representations robust to slight perturbations and limited invariance. This suggests that these ROIs might be generating a scene representation robust to high frequency noise.

When considering 3D DNNs, the 3D Keypoint and the 2.5d segment were among the top-3 best predicting DNNs in multiple ROIs. The 3D Keypoints DNN explained significant unique variance in V3d, PHC1, PHC2, LO2, LO1, V3a, V3b, OPA, and PPA. The 3D Keypoints task requires the DNN to identify locally important regions of the input image based on object boundary information and surface stability. This suggests that the ROIs in which 3D Keypoints DNN explained significant variance may be identifying locally important regions in a scene. The identification of locally important regions might be relevant to selectively attend to these key regions to achieve a behavioral goal e.g., searching for an object. The 2.5d segment DNN explained significant unique variance in V3d, LO2, LO1, V3b, V3a, and OPA. The 2.5d segment task requires the DNN to segment images into perceptually similar groups based on color and scene geometry (depth and surface normals). This suggests that the ROIs in which 2.5d segment DNN explained significant variance may be grouping regions in the images based on color and geometry cues even without any knowledge of the categorical information.

Grouping regions based on geometry could be relevant to behavioral goals such as reaching for objects or identifying obstacles.

Among semantic DNNs, the semantic segmentation DNN explained significant unique variance in VO1, VO2, PHC1, PHC2, V3a, and PPA. The semantic segmentation task requires the DNN to segment objects present in the image based on categories. This suggests that the ROIs in which semantic segmentation DNN explained significant variance may be grouping regions in the image based on categorical information.

Other DNNs (2D edges, scene classification, and object classification) that showed significant unique variance in ROIs provided functional insights mostly consistent with the previous studies [26–30]. Overall, the key DNNs (denoising, 3D keypoints, 2.5D segment, and semantic segmentation) that explained significant variance in multiple ROI responses uniquely promote further investigation by generating novel hypotheses about the functions of these ROIs. Future experiments can test these hypotheses in detail in dedicated experiments.

The functional mapping obtained using grouped DNNs is complementary to that at the individual level and helps us put functional mapping obtained here in context with previous literature. We found that early visual regions (V1v, V1d, V2v) have a functional role related to low-level 2D visual tasks which is consistent with previous literature investigating these regions [26–28]. In dorsal-ventral ROIs (V3a, V3d, LO1, and LO2) we found functional roles related to 3D and semantic tasks converging with evidence from previous studies [31–35]. Similarly, the prominent semantic functional role of later ventral-temporal ROIs (VO1, VO2, PHC1, and PHC2) found in this study converges with findings in previous literature [29–30]. In scene-selective ROIs, we found a semantic functional role for PPA and 3D functional role for OPA respectively. Our study extends the findings of a previous study [23] relating OPA and PPA to 3D models by differentiating between OPA and PPA functions through a much broader set of models. To summarize, the functional mapping using individual DNNs optimized to perform different functions revealed new functional insights for higher ROIs in the visual cortex while at the same time functional mapping using grouped DNNs showed highly converging evidence with previous independent studies investigating these ROIs.

Beyond clarifying the functional roles of multiple ROIs, our approach also identifies quantitatively highly accurate prediction models of these ROIs. We found that the DNNs explained 60% of the explainable variance in nine out of 15 ROIs. Our findings, thus, make advances towards finding models that generate new hypotheses about potential functions of brain regions as well as predicting brain responses well [21,36–38].

A major challenge in meaningfully comparing two or more DNNs is to vary only a single factor of interest while controlling the factors that may lead to updates of DNN parameters. In this study, we address this challenge by selecting a set of DNNs trained on the same set of training images using the same learning algorithm, with the same encoder architecture, while being optimized for different tasks. Our results, thus, complement previous studies that focused on other factors influencing the learning of DNN parameters such as architecture [20,39,40], and the learning mechanism [41–43]. Our approach accelerates the divide-and-conquer strategy of investigating human brain function by systematically and carefully manipulating the DNNs used to map the brain in their fundamental parameters one by one [21,44–46]. Our high-throughput exploration of potential computational functions was initially inspired by Marr's computational level of analysis [47] which aims at finding out what the goal of the computation carried out by a brain region is. While Marr's approach invites the expectation of a one-to-one mapping between regions and goals, we found evidence for multiple functional roles (3D + semantic) using DNNs in some ROIs (e.g. LO1, LO2, PHC1, PHC2). This indicates a many-to-one mapping [48] between functions and brain regions. We believe such a systematic

approach that finds the functional roles of multiple brain regions provides a starting point for a further in-depth empirical inquiry into functions of the investigated brain regions.

Our study is related to a group of studies [49–52] applying DNNs in different ways to achieve a similar goal of mapping functions of brain regions using DNNs. Some studies [49–51] applied optimization algorithms (genetic algorithm or activation maximization) to find images that maximally activate a given neuron's or group of neurons' response. Another related study [52] proposes Neural Information Flow (NIF) to investigate functions of brain regions where they train a DNN with the objective function to predict brain activity while preserving a one-to-one correspondence between DNN layers and biological neural populations. While sharing the overall goal to discover functions of brain regions, investigating DNN functions allows investigation in terms of which computational goal a given brain region is best aligned with. With new computer vision datasets [53] investigating a diverse set of tasks relevant to human behavioral goals [54,55] our approach opens new avenues to investigate brain functions.

A limitation of our study is that our findings are restricted to functions related to scene perception. Thus, the functions we discovered for non-scene regions correspond to their functions when humans are perceiving scenes. In contrast, our study does not characterize the functions of these regions when humans perceive non-scene categories such as objects, faces, or bodies. We limited our study to scene perception because there are only a few image datasets [8,56] that have annotations corresponding to a diverse set of tasks, thus, allowing DNNs to be optimized independently on these tasks. The Taskonomy dataset [8] with annotations of over 20 diverse scene perception tasks and pretrained DNNs available on these tasks along with the availability of an fMRI dataset related to scene perception [9], therefore, provided a unique opportunity. However, the approach we presented in this study is not limited to scene perception. It can in principle be extended to more complex settings such as video understanding, active visual perception, and even outside the vision modality, given an adequate set of DNNs and brain data. While in this study we considered DNNs that were trained independently, future studies might consider investigating multitask models [57,58] which are trained to perform a wide range of functions using a single DNN. Multitask modeling has the potential to model the entire visual cortex using a single model as compared to several independent models used in this study. Another potential limitation is that our findings are based on a single fMRI and image dataset, so it is not clear how well they would generalize to a broader sample of images. Given the explosive growth of the deep learning field [59] and the ever increasing availability of open brain imaging data sets [60,61] we see a furtive ground for the application of our approach in the future.

Beyond providing theoretical insight with high predictive power, our approach can also guide future research. In particular, the observed mapping between cortical region and function can serve as a quantitative baseline and starting point for an in-depth investigation focused on single cortical regions. Finally, the functional hierarchy of the visual cortex from our results can inspire the design of efficient multi-task artificial visual systems that perform multiple functions similar to the human visual cortex.

4. Materials and methods

4.1 fMRI data

We used fMRI data from a previously published study [9]. The fMRI data were collected from 16 healthy subjects (8 females, mean age 29.4 years, SD = 4.8). The subjects were scanned on a Siemens 3.0T Prisma scanner using a 64-channel head coil. Structural T1-weighted images were acquired using an MPRAGE protocol (TR = 2,200 ms, TE = 4.67 ms, flip angle = 8˚,

matrix size = 192 × 256 × 160, voxel size = 0.9 × 0.9 × 1 mm). Functional T2*-weighted images were acquired using a multi-band acquisition sequence (TR = 2,000 ms for main experimental scans and 3,000 ms for localizer scans, TE = 25 ms, flip angle = 70˚, multiband factor = 3, matrix size = 96 × 96 × 81, voxel size = 2 × 2 × 2 mm).

During the fMRI scan, subjects performed a category detection task while viewing images of indoor scenes. On each trial, an image was presented on the screen at a visual angle of ~17.1˚ x 12.9˚ for 1.5 s followed by a 2.5s interstimulus interval. Subjects had to respond by pressing a button indicating whether the presented image was a bathroom or not while maintaining fixation on a cross. The stimulus set consisted of 50 images of indoor scenes (no bathrooms), and 12 control images (five bathroom images, and seven non-bathroom images). fMRI data were preprocessed using SPM12. For each participant, the functional images were realigned to the first image followed by co-registration to the structural image. Voxelwise responses to 50 experimental conditions (50 indoor images excluding control images) were estimated using a general linear model.

4.2 Deep neural networks

For this study, we selected 18 DNNs trained on the Taskonomy [8] dataset optimized on 18 different tasks covering different aspects of indoor scene understanding. The Taskonomy dataset is a large-scale indoor image dataset consisting of annotations for 18 single image tasks, thus, allowing optimization of DNNs on 18 different tasks using the same set of training images. We briefly describe the objective functions and DNN architectures below. For a detailed description, we refer the reader to Zamir et al. [8].

4.2.1 Tasks and objective functions of the DNNs. The Taskonomy dataset consists of annotations for tasks that require pixel-level information such as edge detection, surface normal estimation, semantic segmentation, etc. as well as high-level semantic information such as object/scene classification probabilities. The tasks can be broadly categorized into 4 groups: relating to low-level visual information (2D), the three-dimensional layout of the scene (3D), high-level object and scene categorical information (semantic), and low-dimensional geometry information(geometrical). The above task categorization was obtained by analyzing the relationship between the transfer learning performance on a given task using the models pretrained on other tasks as the source tasks. The 2D tasks were edge detection, keypoint detection, 2D segmentation, inpainting, denoising, and colorization; 3D tasks were surface normals, 2.5D segmentation, occlusion edges, depth estimation, curvature estimation, and reshading; semantic tasks were object/scene classification and semantic segmentation, and low-dimensional geometric tasks were room layout estimation and vanishing point. A detailed description of all the tasks and annotations is provided in http://taskonomy.stanford.edu/taskonomy_supp_CVPR2018.pdf. In this study, we did not consider low dimensional geometric tasks as they did not fall into converging clusters according to RSA and transfer learning as in the case of 2D, 3D, and semantics tasks. To perform a given task, DNN's parameters were optimized using an objective function that minimizes the loss between the DNN prediction and corresponding ground truth annotations for that task. All the DNNs' parameters were optimized using the corresponding objective function, on the same set of training images. Due to the use of the same set of training images the learned DNN parameters vary only due to the objective function and not the difference in training dataset statistics. A complete list of objective functions used to optimize for each task is provided in this link (https://github.com/StanfordVL/taskonomy/tree/master/taskbank). We downloaded the pretrained models using this link (https://github.com/StanfordVL/taskonomy/tree/master/taskbank), where further details can be found.

4.2.2 Network architectures. The DNN architecture for each task consists of an encoder and a decoder. The encoder architecture is consistent across all the tasks. The encoder architecture is a modified ResNet-50 [62] without average pooling and convolutions with stride 2 replaced by convolutions with stride 1. ResNet-50 is a 50-layer DNN with shortcut connections between layers at different depths. Consistency of encoder architecture allows us to use the outputs of the ResNet-50 encoder as the task-specific representation for a particular objective function. For all the analysis in this study, we selected the last two layers of the encoder as the task-specific representation of the DNN. Our selection criteria was based on an analysis (see S1 Text and S2 Fig) that shows task-specific representation is present in those layers as compared to earlier layers. In this way, we ensure that the difference in representations is due to the functions these DNNs were optimized for and not due to the difference in architecture or training dataset. The decoder architecture is task-dependent. For tasks that require pixel-level prediction, the decoder is a 15-layer fully convolutional model consisting of 5 convolutional layers followed by alternating convolution and transposed convolutional layers. For tasks, which require low dimensional output, the decoder consists of 2–3 fully connected layers.

4.3 Representational Similarity Analysis (RSA)

To compare the fMRI responses with DNN activations we first need to map both the modalities in a common representational space and then by comparing the resulting mappings we can quantify the similarity between fMRI and DNNs. We mapped the fMRI responses and DNN activations to corresponding representational dissimilarity matrices (RDMs) by computing pairwise distances between each pair of conditions. We used the variance of upper triangular fMRI RDM (R^2) explained by DNN RDMs as the measure to quantify the similarity between fMRI responses and DNN activations. To calculate R^2, we assigned DNN RDMs (RDMs of the last two layers of the encoder) as the independent variables and assigned fMRI RDM as the dependent variable. Then a multiple linear regression was fitted to predict fMRI RDM from the weighted linear combination of DNN RDMs. We evaluated the fit by estimating the variance explained (R^2). We describe how we mapped from fMRI responses and DNN activations to corresponding RDMs in detail below.

Taskonomy DNN RDMs. We selected the last two layers of the Resnet-50 encoder as the task-specific representation of DNNs optimized on each task. For a given DNN layer, we computed the Pearson's distance between the activations for each pair of conditions resulting in a condition x condition RDM for each layer. This resulted in a single RDM corresponding to each DNN layer. We followed the same procedure to create RDMs corresponding to other layers of the network. We averaged the DNN RDMs across task clusters (2D, 3D, and semantic) to create 2D, 3D, and semantic RDMs.

Probabilistic ROI RDMs. We downloaded probabilistic ROIs [15] from the link (http://scholar.princeton.edu/sites/default/files/napl/files/probatlas_v4.zip). We extracted activations of the probabilistic ROIs by applying the ROI masks on the whole brain response pattern for each condition, resulting in ROI-specific responses for each condition for each subject. Then for each ROI, we computed the Pearson's distance between the voxel response patterns for each pair of conditions resulting in a RDM (with rows and columns equal to the number of conditions) independently for each subject. To compare the variance of ROI RDM explained by DNN RDMs with the explainable variance we used independent subject RDMs. For all the other analyses, we averaged the RDMs across the subjects resulting in a single RDM for each ROI due to a higher signal to noise ratio in subject averaged RDMs.

Searchlight RDMs. We used Brainiak toolbox code [63] to extract the searchlight blocks for each condition in each subject. The searchlight block was a cube with radius = 1 and edge

size = 2. For each searchlight block, we computed the Pearson's distance between the voxel response patterns for each pair of conditions resulting in a RDM of size condition times condition independently for each subject. We then averaged the RDMs across the subjects resulting in a single RDM for each searchlight block.

4.4 Variance partitioning

Using RSA to compare multiple DNNs we do not obtain a complete picture of how each model is contributing to explaining the fMRI responses when considered in conjunction with other DNNs. Therefore, we determined the unique and shared contribution of individual DNN RDMs in explaining the fMRI ROI RDMs when considered with the other DNN RDMs using variance partitioning.

We performed two variance partitioning analyses on probabilistic ROIs: first using the top-3 DNNs that best explained a given ROI's responses and second using RDMs averaged according to task type (2D, 3D, and semantic). For the first analysis, we assigned a fMRI ROI RDM as the dependent variable (referred to as predictand) and assigned RDMs corresponding to the top-3 DNNs as the independent variables (referred to as predictors). For the second analysis, we assigned an fMRI ROI (searchlight) RDM as the dependent variable (referred to as predictand). We then assigned three DNN RDMs (2D, 3D, and semantic) as the independent variables (referred to as predictors).

For both variance partitioning analyses, we performed seven multiple regression analyses: one with all three independent variables as predictors, three with different pairs of two independent variables as the predictors, and three with individual independent variables as the predictors. Then, by comparing the explained variance (R^2) of a model used alone with the explained variance when it was used with other models, we can infer the amount of unique and shared variance between different predictors (see S1 Fig).

4.5 Searchlight analysis

We perform two different searchlight analyses in this study: first to find out if different regions in the brain are better explained by DNNs optimized for different tasks and second to find the pattern by taking the averaged representation DNNs from three task types (2D, 3D, and semantic). In the first searchlight analysis, we applied RSA to compute the variance of each searchlight block RDM explained by 19 DNN RDMs (18 Taskonomy DNNs and one randomly initialized as a baseline) independently. We then selected the DNN that explained the highest variance as the preference for the given searchlight block. In the second searchlight analysis, we applied variance partitioning with 2D, 3D, and semantic DNN RDMs as the independent variables, and each searchlight block RDM as the dependent variable. For each searchlight block, we selected the task type whose RDMs explained the highest variance uniquely as the function for that block. We used the nilearn (https://nilearn.github.io/index.html) library to plot and visualize the searchlight results.

4.6 Comparison of explained with explainable variance

To relate the variance of fMRI responses explained by a DNN to the total variance to be explained given the noisy nature of the fMRI data, we first calculated the lower and upper bounds of the noise ceiling as a measure of explainable variance and then compared cross-validated explained variance of each ROI by top-3 best predicting DNNs. In detail, the lower noise ceiling was estimated by fitting each individual subject RDMs as predictand with mean subject RDM of other subjects (N-1) as the predictor and calculating the R^2. The resulting subject-specific R^2 values were averaged across the N subjects. The upper noise ceiling was estimated in a

similar fashion while using mean subject RDMs of all the subjects (N) as the predictor. To calculate variance explained by the best predicting DNNs we fit the regression using cross validation in 2N folds (2 folds across conditions, N folds across subjects) where the regression was fit using the subject averaged RDMs of N-1 subjects and the fit was evaluated using R^2 on the left out subject and left out conditions. Finally, we then calculated the mean R^2 across 2N folds and divided it by the lower bound of the noise ceiling to obtain the ratio of the explainable variance explained by the DNNs.

4.7 Statistical testing

We applied nonparametric statistical tests to assess the statistical significance in a similar manner to a previous related study [64]. We assessed the significance of the R^2 through a permutation test by permuting the conditions randomly 10,000 times in either the neural ROI/searchlight RDM or the DNN RDM. From the distribution obtained using these permutations, we calculated p-values as one-sided percentiles. We calculated the standard errors of these correlations by randomly resampling the conditions in the RDMs for 10,000 iterations. We used re-sampling without replacement by subsampling 90% (45 out of 50 conditions) of the conditions in the RDMs. We used an equivalent procedure for testing the statistical significance of the correlation difference and unique variance difference between different models.

For ROI analysis, we corrected the p-values for multiple comparisons by applying FDR correction with a threshold equal to 0.05. For searchlight analyses, we applied FDR correction to correct for the number of DNNs compared as well as to correct for the number of searchlights that had a significant noise ceiling.

We applied a two-sided t-test to assess the statistical significance of the cross-validated explained variance across N subjects. We corrected the p-values for multiple comparisons by applying FDR correction.

Supporting information

S1 Fig. Variance partitioning overview. Given a set of multiple independent variables and dependent variables, multiple linear regression results in R-squared (R^2) that represents the proportion of the variance for a dependent variable that's explained by independent variables in a regression model. To find how 3 DNN RDMs together explain the variance of a given fMRI RDM we perform 7 multiple regression and illustrate unique and shared variance explained by models through a Venn diagram.
(TIFF)

S2 Fig. Selecting task-specific DNN representation to compare with fMRI data. A) Spearman's correlation of all DNN RDMs at a given layer of the encoder with other DNN RDMs computed at the same layer. We report the mean pairwise correlation of all 18 DNNs at different layers of the encoder. **B)** Spearman's correlation of all DNN RDMs at a given layer of the encoder with a randomly initialized model with the same architecture computed at the same layer. We report the mean correlation of all 18 DNNs with the randomly initialized DNN at different layers of the encoder. **C)** Spearman's correlation of all DNN RDMs at a given layer of the encoder with deeper layers (block4 and encoder output) of 2D DNNs. We report the mean correlation of the key layers of all 18 DNNs with deeper layers (block4 and encoder output) of 2D DNNs. **D)** Spearman's correlation between layers at different depths for DNNs corresponding to different task types. We report the mean correlation between different layers averaged across different DNNs of the same task type. **E)** Effect of adding all the key layers on unique and shared variance of fMRI RDMs from different ROIs as compared to selecting only

task-specific layers for variance partitioning analysis. We report the change in variance explained (variance change) for 7 variance partitions when all key layers were used for analysis as compared to selecting task-specific layers.
(TIFF)

S3 Fig. R^2 ranking for 18 Taskonomy DNNs and random baseline in anatomical ROIs. The bar plot shows the absolute total variance of each ROI RDM explained by task-specific layer RDMs of a given DNN. The asterisk denotes the significance of total variance ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs). The error bars show the standard deviation calculated by bootstrapping 90% of the conditions (10,000 iterations).
(TIFF)

S4 Fig. R^2 ranking for 18 Taskonomy DNNs and random baseline in functionally localized ROIs. The bar plot shows the absolute total variance of each ROI RDM explained by task-specific layer RDMs of a given DNN. The asterisk denotes the significance of total variance ($p<0.05$, permutation test with 10,000 iterations, FDR-corrected across DNNs). The error bars show the standard deviation calculated by bootstrapping 90% of the conditions (10,000 iterations).
(TIFF)

S5 Fig. Effect of cross validation on variance explained (R^2). A) Variance of each ROI explained by top-3 best predicting DNNs compared for different cross-validation settings (blue bars: no cross validation; orange bars: cross validation across subjects; green bars: cross validation across subjects and stimuli). The error bars show the 95% confidence interval calculated across N = 16 subjects. All the R^2 values are statistically significant ($p<0.05$, two-sided t-test, FDR-corrected across ROIs) **B)** Variance of each ROI explained by 1000 randomly generated RDMs compared for different cross-validation settings (blue bars: no cross validation; orange bars: cross validation across subjects; green bars: cross validation across subjects and stimuli). The error bars show the 95% confidence interval calculated across N = 16 subjects.
(TIFF)

S1 Text. Selecting task-specific DNN representations.
(DOCX)

Author Contributions

Conceptualization: Kshitij Dwivedi, Radoslaw Martin Cichy, Gemma Roig.

Data curation: Michael F. Bonner.

Formal analysis: Kshitij Dwivedi, Michael F. Bonner.

Funding acquisition: Radoslaw Martin Cichy, Gemma Roig.

Investigation: Kshitij Dwivedi, Gemma Roig.

Methodology: Kshitij Dwivedi, Michael F. Bonner, Radoslaw Martin Cichy, Gemma Roig.

Project administration: Gemma Roig.

Software: Kshitij Dwivedi.

Supervision: Radoslaw Martin Cichy, Gemma Roig.

Validation: Kshitij Dwivedi.

Visualization: Kshitij Dwivedi.

Writing – original draft: Kshitij Dwivedi, Radoslaw Martin Cichy, Gemma Roig.

Writing – review & editing: Kshitij Dwivedi, Michael F. Bonner, Radoslaw Martin Cichy, Gemma Roig.

References

1. Mishkin M, Ungerleider LG. Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. Behavioural brain research. 1982 Sep 1; 6(1):57–77. https://doi.org/10.1016/0166-4328(82)90081-x PMID: 7126325
2. Grill-Spector K, Malach R. The human visual cortex. Annu. Rev. Neurosci. 2004 Jul 21; 27:649–77. https://doi.org/10.1146/annurev.neuro.27.070203.144220 PMID: 15217346
3. Cadieu CF, Hong H, Yamins DL, Pinto N, Ardila D, Solomon EA, Majaj NJ, DiCarlo JJ. et.al Deep neural networks rival the representation of primate IT cortex for core visual object recognition. PLoS computational biology. 2014 Dec 18; 10(12):e1003963. https://doi.org/10.1371/journal.pcbi.1003963 PMID: 25521294
4. Cichy RM, Khosla A, Pantazis D, Torralba A, Oliva A. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Scientific reports. 2016 Jun 10; 6(1):1–3.
5. Güçlü U, van Gerven MA. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. Journal of Neuroscience. 2015 Jul 8; 35(27):10005–14. https://doi.org/10.1523/JNEUROSCI.5023-14.2015 PMID: 26157000
6. Khaligh-Razavi SM, Kriegeskorte N. Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS computational biology. 2014 Nov 6; 10(11):e1003915. https://doi.org/10.1371/journal.pcbi.1003915 PMID: 25375136
7. Yamins DL, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ. et.al Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the national academy of sciences. 2014 Jun 10; 111(23):8619–24. https://doi.org/10.1073/pnas.1403112111 PMID: 24812127
8. Zamir AR, Sax A, Shen W, Guibas LJ, Malik J, Savarese S. Taskonomy: Disentangling task transfer learning. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 3712–3722).
9. Bonner MF, Epstein RA. Coding of navigational affordances in the human visual system. Proceedings of the National Academy of Sciences. 2017 May 2; 114(18):4793–8. https://doi.org/10.1073/pnas.1618228114 PMID: 28416669
10. Etzel JA, Zacks JM, Braver TS. Searchlight analysis: promise, pitfalls, and potential. Neuroimage. 2013 Sep 1; 78:261–9. https://doi.org/10.1016/j.neuroimage.2013.03.041 PMID: 23558106
11. Haynes JD, Sakai K, Rees G, Gilbert S, Frith C, Passingham RE. et.al Reading hidden intentions in the human brain. Current Biology. 2007 Feb 20; 17(4):323–8. https://doi.org/10.1016/j.cub.2006.11.072 PMID: 17291759
12. Kriegeskorte N, Goebel R, Bandettini P. Information-based functional brain mapping. Proceedings of the National Academy of Sciences. 2006 Mar 7; 103(10):3863–8. https://doi.org/10.1073/pnas.0600244103 PMID: 16537458
13. Kriegeskorte N, Mur M, Bandettini PA. Representational similarity analysis-connecting the branches of systems neuroscience. Frontiers in systems neuroscience. 2008 Nov 24; 2:4. https://doi.org/10.3389/neuro.06.004.2008 PMID: 19104670
14. Dwivedi K, Roig G. Representation similarity analysis for efficient task taxonomy & transfer learning. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019 (pp. 12387–12396).
15. Wang L, Mruczek RE, Arcaro MJ, Kastner S. Probabilistic maps of visual topography in human cortex. Cerebral cortex. 2015 Oct 1; 25(10):3911–31. https://doi.org/10.1093/cercor/bhu277 PMID: 25452571
16. Legendre P. Studying beta diversity: ecological variation partitioning by multiple regression and canonical analysis. Journal of plant ecology. 2008 Mar 1; 1(1):3–8.
17. Storrs KR, Kietzmann TC, Walther A, Mehrer J, Kriegeskorte N. Diverse deep neural networks all predict human IT well, after training and fitting. bioRxiv. 2020 Jan 1.
18. Dwivedi K, Cichy RM, Roig G. Unraveling Representations in Scene-selective Brain Regions Using Scene-Parsing Deep Neural Networks. Journal of Cognitive Neuroscience. 2020 Mar 10:1–2. https://doi.org/10.1162/jocn_a_01624 PMID: 32897121

19. Groen II, Greene MR, Baldassano C, Fei-Fei L, Beck DM, Baker CI. et.al Distinct contributions of functional and deep neural network features to representational similarity of scenes in human brain and behavior. Elife. 2018 Mar 7; 7:e32962. https://doi.org/10.7554/eLife.32962 PMID: 29513219

20. Nayebi A, Bear D, Kubilius J, Kar K, Ganguli S, Sussillo D, DiCarlo JJ, Yamins DL. et.al Task-driven convolutional recurrent models of the visual system. arXiv preprint arXiv:1807.00053. 2018 Jun 20.

21. Yamins DL, DiCarlo JJ. Using goal-driven deep learning models to understand sensory cortex. Nature neuroscience. 2016 Mar; 19(3):356–65. https://doi.org/10.1038/nn.4244 PMID: 26906502

22. Kell AJ, Yamins DL, Shook EN, Norman-Haignere SV, McDermott JH. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. Neuron. 2018 May 2; 98(3):630–44. https://doi.org/10.1016/j.neuron.2018.03.044 PMID: 29681533

23. Lescroart MD, Gallant JL. Human scene-selective areas represent 3D configurations of surfaces. Neuron. 2019 Jan 2; 101(1):178–92. https://doi.org/10.1016/j.neuron.2018.11.004 PMID: 30497771

24. Güçlü U, van Gerven MA. Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. NeuroImage. 2017 Jan 15; 145:329–36. https://doi.org/10.1016/j.neuroimage.2015.12.036 PMID: 26724778

25. Wang Aria Y., et al. "Neural Taskonomy: Inferring the Similarity of Task-Derived Representations from Brain Activity." BioRxiv, July 2019, p. 708016. www.biorxiv.org.

26. Avidan G, Harel M, Hendler T, Ben-Bashat D, Zohary E, Malach R. et.al Contrast sensitivity in human visual areas and its relationship to object recognition. Journal of neurophysiology. 2002 Jun 1; 87 (6):3102–16. https://doi.org/10.1152/jn.2002.87.6.3102 PMID: 12037211

27. Boynton GM, Demb JB, Glover GH, Heeger DJ. Neuronal basis of contrast discrimination. Vision research. 1999 Jan 1; 39(2):257–69. https://doi.org/10.1016/s0042-6989(98)00113-8 PMID: 10326134

28. Ress D, Heeger DJ. Neuronal correlates of perception in early visual cortex. Nature neuroscience. 2003 Apr; 6(4):414–20. https://doi.org/10.1038/nn1024 PMID: 12627164

29. Arcaro MJ, McMains SA, Singer BD, Kastner S. Retinotopic organization of human ventral visual cortex. Journal of neuroscience. 2009 Aug 26; 29(34):10638–52. https://doi.org/10.1523/JNEUROSCI.2807-09.2009 PMID: 19710316

30. Grill-Spector K, Weiner KS. The functional architecture of the ventral temporal cortex and its role in categorization. Nature Reviews Neuroscience. 2014 Aug; 15(8):536–48. https://doi.org/10.1038/nrn3747 PMID: 24962370

31. Backus BT, Fleet DJ, Parker AJ, Heeger DJ. Human cortical activity correlates with stereoscopic depth perception. Journal of neurophysiology. 2001 Oct 1; 86(4):2054–68. https://doi.org/10.1152/jn.2001.86.4.2054 PMID: 11600661

32. Grill-Spector K, Kourtzi Z, Kanwisher N. The lateral occipital complex and its role in object recognition. Vision research. 2001 May 1; 41(10–11):1409–22. https://doi.org/10.1016/s0042-6989(01)00073-6 PMID: 11322983

33. Kourtzi Z, Erb M, Grodd W, Bülthoff HH. Representation of the perceived 3-D object shape in the human lateral occipital complex. Cerebral cortex. 2003 Sep 1; 13(9):911–20. https://doi.org/10.1093/cercor/13.9.911 PMID: 12902390

34. Moore C, Engel SA. Neural response to perception of volume in the lateral occipital complex. Neuron. 2001 Jan 1; 29(1):277–86. https://doi.org/10.1016/s0896-6273(01)00197-0 PMID: 11182098

35. Stanley DA, Rubin N. fMRI activation in response to illusory contours and salient regions in the human lateral occipital complex. Neuron. 2003 Jan 23; 37(2):323–31. https://doi.org/10.1016/s0896-6273(02)01148-0 PMID: 12546826

36. Cichy RM, Kaiser D. Deep neural networks as scientific models. Trends in cognitive sciences. 2019 Apr 1; 23(4):305–17. https://doi.org/10.1016/j.tics.2019.01.009 PMID: 30795896

37. Khaligh-Razavi SM, Henriksson L, Kay K, Kriegeskorte N. Fixed versus mixed RSA: Explaining visual representations by fixed and mixed feature sets from shallow and deep computational models. Journal of Mathematical Psychology. 2017 Feb 1; 76:184–97. https://doi.org/10.1016/j.jmp.2016.10.007 PMID: 28298702

38. Schrimpf M, Kubilius J, Lee MJ, Murty NA, Ajemian R, DiCarlo JJ. et.al Integrative benchmarking to advance neurally mechanistic models of human intelligence. Neuron. 2020 Sep 11. https://doi.org/10.1016/j.neuron.2020.07.040 PMID: 32918861

39. Kar K, Kubilius J, Schmidt K, Issa EB, DiCarlo JJ. Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. Nature neuroscience. 2019 Jun; 22(6):974–83. https://doi.org/10.1038/s41593-019-0392-5 PMID: 31036945

40. Kietzmann TC, Spoerer CJ, Sörensen LK, Cichy RM, Hauk O, Kriegeskorte N. et.al Recurrence is required to capture the representational dynamics of the human visual system. Proceedings of the

National Academy of Sciences. 2019 Oct 22; 116(43):21854–63. https://doi.org/10.1073/pnas.1905544116 PMID: 31591217

41. Lillicrap TP, Santoro A, Marris L, Akerman CJ, Hinton G. Backpropagation and the brain. Nature Reviews Neuroscience. 2020 Jun; 21(6):335–46. https://doi.org/10.1038/s41583-020-0277-3 PMID: 32303713

42. Roelfsema PR, Holtmaat A. Control of synaptic plasticity in deep cortical networks. Nature Reviews Neuroscience. 2018 Mar; 19(3):166–80. https://doi.org/10.1038/nrn.2018.6 PMID: 29449713

43. Whittington JC, Bogacz R. Theories of error back-propagation in the brain. Trends in cognitive sciences. 2019 Mar 1; 23(3):235–50. https://doi.org/10.1016/j.tics.2018.12.005 PMID: 30704969

44. Epstein RA, Baker CI. Scene perception in the human brain. Annual review of vision science. 2019 Sep 15; 5:373–97. https://doi.org/10.1146/annurev-vision-091718-014809 PMID: 31226012

45. Lindsay GW. Convolutional neural networks as a model of the visual system: Past, present, and future. Journal of cognitive neuroscience. 2020 Feb 6:1–5.

46. Richards BA, Lillicrap TP, Beaudoin P, Bengio Y, Bogacz R, Christensen A, Clopath C, Costa RP, de Berker A, Ganguli S, Gillon CJ. et.al A deep learning framework for neuroscience. Nature neuroscience. 2019 Nov; 22(11):1761–70. https://doi.org/10.1038/s41593-019-0520-2 PMID: 31659335

47. Marr D. *Vision*: *A Computational Investigation Into the Human Representation and Processing of Visual Information*. (MIT Press, 2010).

48. Klein C. Cognitive ontology and region-versus network-oriented analyses. Philosophy of Science. 2012 Dec 1; 79(5):952–60.

49. Ponce CR, Xiao W, Schade PF, Hartmann TS, Kreiman G, Livingstone MS. et.al Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. Cell. 2019 May 2; 177(4):999–1009. https://doi.org/10.1016/j.cell.2019.04.005 PMID: 31051108

50. Bashivan P, Kar K, DiCarlo JJ. Neural population control via deep image synthesis. Science. 2019 May 3; 364(6439). https://doi.org/10.1126/science.aav9436 PMID: 31048462

51. Gu Z, Jamison KW, Khosla M, Allen EJ, Wu Y, Naselaris T, Kay K, Sabuncu MR, Kuceyeski A. et.al NeuroGen: activation optimized image synthesis for discovery neuroscience. arXiv preprint arXiv:2105.07140. 2021 May 15.

52. Seeliger K, Ambrogioni L, Güçlütürk Y, van den Bulk LM, Güçlü U, van Gerven MA. et.al End-to-end neural system identification with neural information flow. PLOS Computational Biology. 2021 Feb 4; 17 (2):e1008558. https://doi.org/10.1371/journal.pcbi.1008558 PMID: 33539366

53. Weihs L, Salvador J, Kotar K, Jain U, Zeng KH, Mottaghi R, Kembhavi A. et.al Allenact: A framework for embodied ai research. arXiv preprint arXiv:2008.12760. 2020 Aug 28.

54. Batra D, Gokaslan A, Kembhavi A, Maksymets O, Mottaghi R, Savva M, Toshev A, Wijmans E. et.al Objectnav revisited: On evaluation of embodied agents navigating to objects. arXiv preprint arXiv:2006.13171. 2020 Jun 23.

55. Weihs L, Kembhavi A, Ehsani K, Pratt SM, Han W, Herrasti A, Kolve E, Schwenk D, Mottaghi R, Farhadi A. et.al Learning generalizable visual representations via interactive gameplay. arXiv preprint arXiv:1912.08195. 2019 Dec 17.

56. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. et.al Microsoft coco: Common objects in context. InEuropean conference on computer vision 2014 Sep 6 (pp. 740–755). Springer, Cham.

57. Scholte HS, Losch MM, Ramakrishnan K, de Haan EH, Bohte SM. Visual pathways from the perspective of cost functions and multi-task deep neural networks. cortex. 2018 Jan 1; 98:249–61. https://doi.org/10.1016/j.cortex.2017.09.019 PMID: 29150140

58. Kokkinos I. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 6129–6138).

59. LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015 May; 521(7553):436–44. https://doi.org/10.1038/nature14539 PMID: 26017442

60. Poldrack RA, Gorgolewski KJ. Making big data open: data sharing in neuroimaging. Nature neuroscience. 2014 Nov; 17(11):1510–7. https://doi.org/10.1038/nn.3818 PMID: 25349916

61. Allen EJ, St-Yves G, Wu Y, Breedlove JL, Dowdle LT, Caron B, Pestilli F, Charest I, Hutchinson JB, Naselaris T, Kay K. et.al A massive 7T fMRI dataset to bridge cognitive and computational neuroscience. bioRxiv. 2021 Jan 1.

62. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770–778).

63. Kumar M, Ellis CT, Lu Q, Zhang H, Capotă M, Willke TL, Ramadge PJ, Turk-Browne NB, Norman KA et.al. BrainIAK tutorials: User-friendly learning materials for advanced fMRI analysis. PLoS computational biology. 2020 Jan 15; 16(1):e1007549. https://doi.org/10.1371/journal.pcbi.1007549 PMID: 31940340

64. Bonner MF, Epstein RA. Computational mechanisms underlying cortical responses to the affordance properties of visual scenes. PLoS computational biology. 2018 Apr 23; 14(4):e1006111. https://doi.org/10.1371/journal.pcbi.1006111 PMID: 29684011

2 Understanding representations in the deep neural networks

2.1 Representation Similarity Analysis for Efficient Task taxonomy and Transfer Learning

Representation Similarity Analysis for Efficient Task taxonomy & Transfer Learning

Kshitij Dwivedi Gemma Roig
Singapore University of Technology and Design
kshitij_dwivedi@mymail.sutd.edu.sg, gemma_roig@sutd.edu.sg

Abstract

Transfer learning is widely used in deep neural network models when there are few labeled examples available. The common approach is to take a pre-trained network in a similar task and finetune the model parameters. This is usually done blindly without a pre-selection from a set of pre-trained models, or by finetuning a set of models trained on different tasks and selecting the best performing one by cross-validation. We address this problem by proposing an approach to assess the relationship between visual tasks and their task-specific models. Our method uses Representation Similarity Analysis (RSA), which is commonly used to find a correlation between neuronal responses from brain data and models. With RSA we obtain a similarity score among tasks by computing correlations between models trained on different tasks. Our method is efficient as it requires only pre-trained models, and a few images with no further training. We demonstrate the effectiveness and efficiency of our method for generating task taxonomy on Taskonomy dataset. We next evaluate the relationship of RSA with the transfer learning performance on Taskonomy tasks and a new task: Pascal VOC semantic segmentation. Our results reveal that models trained on tasks with higher similarity score show higher transfer learning performance. Surprisingly, the best transfer learning result for Pascal VOC semantic segmentation is not obtained from the pre-trained model on semantic segmentation, probably due to the domain differences, and our method successfully selects the high performing models.

1. Introduction

For an artificial agent to perform multiple tasks and learn in a life-long manner, it should be able to re-utilize information acquired in previously learned tasks and transfer it to learn new tasks from a few examples. A solution to the aforementioned setting is to use transfer learning. Transfer learning allows to leverage representations learned from one task to facilitate learning of other tasks, even when labeled data is expensive or difficult to obtain. [30, 3, 23, 10].

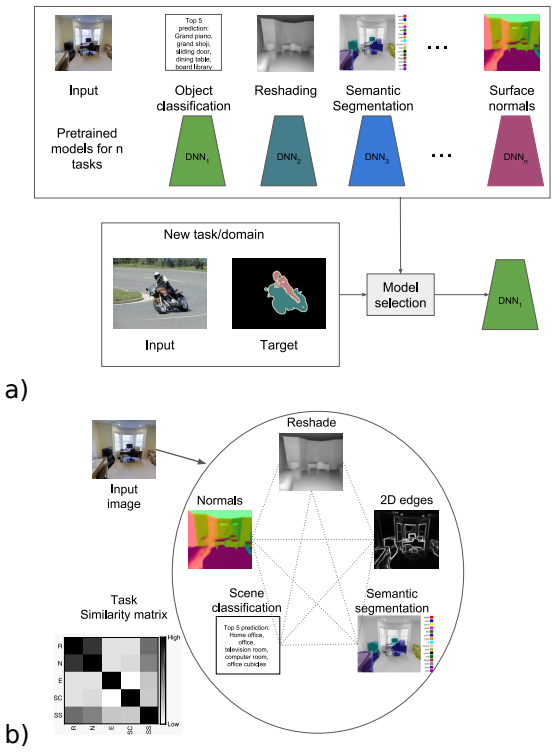

Figure 1. **Aims of this paper: a)** Deploy a strategy for model selection in transfer learning by **b)** Finding relationship between visual tasks.

With the recent success of deep neural networks (DNN), these have become the *ipso facto* models for almost all visual tasks [20, 32, 14, 35, 13, 34]. The deployment of DNN has become possible mostly due to a large amount of available labeled data, as well as advances in computing resources [20, 32, 14]. The need for data is a limita-

tion that researchers have overcome by introducing transfer learning techniques. Transfer learning in DNN commonly consists of taking a pre-trained model in a similar task or domain, and finetune the parameters to the new task. For instance, [30, 10] used a pre-trained model on ImageNet and finetuned it for object detection on Pascal VOC.

With a large number of pre-trained models (Figure 1a) available, trained on a variety of vision tasks, it is not trivial how to select a pre-trained representation suitable for transfer learning. To devise a model selection strategy, it is crucial to understand the underlying structure and relationship between tasks (Figure 1b). If the relationship between different tasks is known, the model selection can be performed by evaluating similarity rankings of different tasks with a new task, using available pre-trained models.

In a recent work, [34] modeled the relationship between tasks with a fully computational approach. They also introduce a dataset called Taskonomy, which contains labels of different visual tasks, ranging from object classification to edge occlusions detection. In this paper, we use the term Taskonomy for both the approach and the dataset from [34].

Taskonomy approach successfully computes the relationship between tasks. Yet, the relationship between a new task with an existing set of tasks is calculated with the transfer learning performance, which is tedious and computationally expensive. The performance on the new task is referred to transfer learning performance. To obtain the relationship of all previous tasks with the new task, Taskonomy approach also needs to compute the transfer learning performance on all the previous tasks using a model trained on the new task as a source. This defeats the purpose of not training a model from scratch for the new task, and all the procedure is computationally demanding as it is repeated for all the existing set of specific-task models. In this work, we address the above limitations by providing an alternative method to find the relationship between tasks.

We propose a novel approach to obtain task relationships using representation similarity analysis (RSA). In computational neuroscience, RSA is widely used as a tool to compare brain responses with computational and behavioral models. Motivated by the success of RSA in neuroscience [18, 4, 16, 1, 5, 25, 11], we investigate the application of RSA in obtaining task similarities (Figure 1b) and in transfer learning (Figure 1a). Our approach relies on the assumption that the representations of the models that perform a related task will be more similar as compared to tasks that are not related, which we validate in our analysis.

In our approach, we compute the similarity scores using pre-trained task-specific models and a few examples. Thus, our RSA method only requires the representations of a few randomly selected images for all the tasks to compute the similarity, and we do not need to obtain transfer learning performance by finetuning on previous tasks' models. Further, we show in our results on Taskonomy dataset that task ranking similarity is independent of model size. Using small models trained with few samples for the existing tasks show similar results as the high performing models trained with all images. This allows to save computational time and memory, as well as it is more scalable to new tasks compared to Taskonomy approach.

We first validate the transfer learning applicability of our method on Taskonomy dataset. We find that for 16 out 17 Taskonomy tasks, the best model selected using RSA is in top-5 according to transfer learning performance. We also report results on Pascal VOC semantic segmentation task by analyzing the relationship of RSA similarity scores and the transfer learning performance. Our results show a strong relationship between RSA similarity score and transfer learning performance. We note that semantic segmentation model from Taskonomy dataset showed a lower similarity score than most of the 3D and semantic tasks, and a similar trend was observed in transfer learning performance. Our results suggest that in domain-shift, a model trained on the same task may not be the best option for transfer learning, and using our similarity score one can find a better model to achieve better performance. Using our RSA similarity scores method, we can select models with better transfer learning performance.

2. Related Works

Here, we discuss the works that are most closely related to the aim of this paper, namely transfer learning in DNNs and Taskonomy. Then, we briefly introduce the computational neuroscience literature that motivated our work.

2.1. Transfer Learning

The usual transfer learning approach in deep neural networks (DNNs) is to take a model pre-trained on a large dataset with annotations as an initialization of a part of the model. Then, some or all of the parameters are finetuned with backpropagation for a new task. The finetuning is performed because for most of the tasks there are insufficient annotations to train a DNN from scratch, which would lead to overfitting. Most of the works in the literature generally initialize the model parameters from a model pre-trained on Imagenet [6] dataset for image classification [20, 32, 14, 31, 22]. For example, [30] use Imagenet initialized models for object detection on Pascal VOC, [23] use Imagenet initialized models for semantic segmentation.

It has been noted in multiple works [24, 33, 28], that the initialization plays a significant role in performance in transfer learning. Hence, a strategy is required to select models for initialization. Our proposed similarity-based ranking approach offers a solution to this problem, and as we discuss in the rest of the paper, tackles the limitations

from Taskonomy [34], which is one of the first attempts to tackle the model selection for transfer learning in DNN.

2.2. Taskonomy

Our work is most closely related to Taskonomy [34], where the aim is to find the underlying task structure by computing the transfer performance among tasks. To achieve this goal, they create a dataset of indoor scene images with annotations available for 26 vision tasks. The task set, which they refer as task dictionary, covers common 2D, 3D, and semantics computer vision tasks. Then, task-specific independent models are trained in a fully supervised manner for each task in the task dictionary. They obtain a task similarity score by comparing the transfer learning performance from each of the task-specific models and computing an affinity matrix using a function of transfer learning performance. In this paper, instead of transfer learning performance, we rely on the similarity of the feature maps of the pre-trained models. Thus, we avoid additional training on pre-trained models to obtain transfer learning performance, saving computational time and memory, and still obtaining a meaningful relation with transfer learning performance as we will see in the results section.

2.3. Similarity of computational models and brain responses

In computational neuroscience, representation similarity analysis (RSA) is widely used to compare a computational or behavioral model with the brain responses. In [18], RSA is used to compute similarities between brain responses in different regions of visual cortex with categorical models and computational vision models. In [16], the authors use several unsupervised and supervised vision models to show that supervised models explain IT cortical area better than unsupervised models, and [25] uses RSA to correlate the dynamics of the visual system with deep neural networks. We note that as the approach can be used to assess the similarity between a computational model and brain data, the approach can also be utilized to assess similarities between two computational models. RSA has been rarely used in the pure computational domain. Only in [26] the RSA was introduced as a loss function for knowledge distillation [15], and in [27], the consistency of RSA correlations with different random initialization seeds within the same model trained on CIFAR-10 [19] dataset is explored. However, RSA is still unexplored in comparing DNNs for assessing similarity among them. Our work introduces, for the first time, the use of RSA as a similarity measure to find the relationship between tasks, and we believe it opens a new research line for the deep learning and computer vision.

We use RSA similarity measure for two applications namely task taxonomy and transfer learning. Our approach is not limited to only these two applications and can be

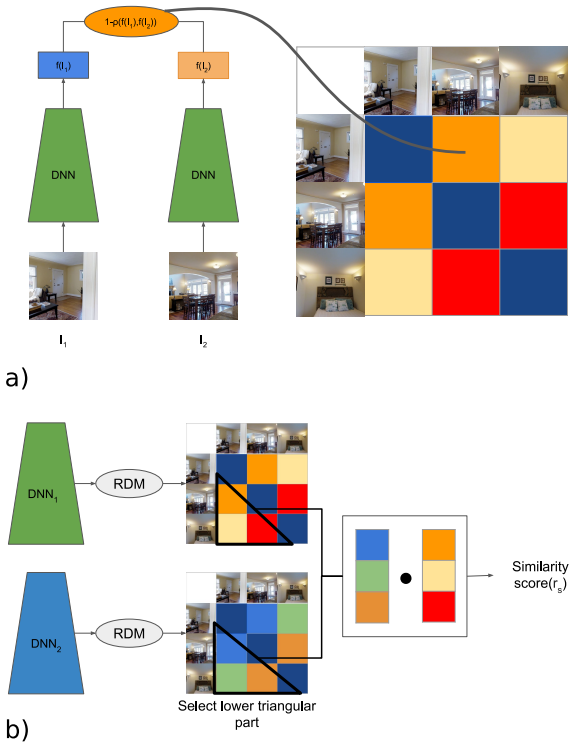

Figure 2. **Representation Similarity Analysis (RSA): a)** Representation dissimilarity matrices (RDMs) are generated by computing the pairwise dissimilarity (1 - Pearson's correlation) of each image pair in a subset of selected images. **b)** Similarity score: Spearman's correlation (r_s) (denoted with •) of the low triangular RDMs of the two models is used as the similarity score. Here DNN_1 and DNN_2 refer to the models trained on task 1 and 2 respectively.

further applied in other computer vision problems. For instance, in multi-task learning [17, 13, 7, 21, 8] RSA could be used for deciding different branching out locations for different tasks, depending on their similarity with the representations at different depth of the shared root.

3. Representation Similarity Analysis (RSA)

Representation Similarity Analysis (RSA) [18], illustrated in Figure 2, is a widely used data-analytical framework in the field of computational neuroscience to quantitatively relate the brain activity measurement with computational and behavioral models. In RSA, a computational model and brain activity measurements are related by comparing representation-activity dissimilarity matrices. The dissimilarity matrices are obtained by comparing the pairwise dissimilarity of activity/representation associated with each pair of conditions.

In this work, we introduce RSA as a tool to quantify the relationship between DNNs and its application in transfer

learning for model selection. We explain the steps to obtain the dissimilarity matrix for a computational model such as DNN in the following paragraph.

Representation Dissimilarity Matrix (RDM) We first select a subset of images as conditions for dissimilarity computation. For a given DNN, we then obtain the representation of each image by performing a forward pass through the model. For each pair of conditions (images), we compute a dissimilarity score $1-\rho$, where ρ is the Pearson's correlation coefficient. The RDM for this subset of conditions is then populated by the dissimilarity scores for each pair of conditions, see Figure 2a.

In our method, the RDMs computed for DNNs are used for obtaining the similarity between two computer vision tasks. Note that by using RDMs, the representation for different tasks can be of different length. The similarity is computed with the Spearman's correlation (r_s) between the upper or lower triangular part of the RDMs of the two DNNs. This is: $r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$, where d_i is the difference between the ranks of i_{th} elements of the lower triangular part of the two RDMs in Figure 2b, and n are the number of elements in the lower triangular part of the RDM.

The Spearman's correlation provides a quantitative measure of similarity between the task the DNNs were optimized for (Figure 2b). We explore the application of this similarity score in obtaining the relationship between computer vision tasks [34], and in transfer learning.

4. RSA for Task Taxonomy and Transfer Learning

In this section, we introduce our RSA approach for getting a task taxonomy of computer vision tasks, as well as its application in transfer learning. We show the effectiveness of RSA for obtaining task similarity by answering three questions: 1) we investigate if we can group tasks into meaningful clusters based on task type using RSA on pre-trained task-specific models; 2) we analyze if the performance is important for computing task similarity or we can use a smaller subset of data with smaller suboptimal models; and 3) we investigate if the similarity we obtain using RSA is related to transfer learning.

4.1. Is task similarity related to task type?

We validate our hypothesis that tasks similar according to RSA are grouped into clusters according to task type, for instance, 2D, 3D, semantic. To do so, we randomly select 500 images from the Taskonomy dataset, and select 20[1] tasks from the task dictionary. Then, we compute the RDMs of the pre-trained models for each of the 20 tasks using the task-specific representations of the 500 sampled images, as described in section 3. The task-specific representations are obtained by doing a forward pass on the pre-trained task-specific DNN models. With the resulting RDMs per task, we compute a pairwise correlation of RDMs of each task with the 19 other tasks to get a 20×20 task similarity matrix (Figure 3a). We perform a hierarchical clustering from the similarity matrix, to visualize if the clustering groups the tasks according to the task type or some other criteria. We report the results in the experiments section and compare it with the clustering obtained with the Taskonomy approach.

We note that RSA is symmetric, as compared to the transfer performance based metric in Taskonomy [34]. Yet, symmetry does not affect task similarity rankings, as the positions of the tasks in the rankings are computed by relative comparison, and therefore, independent of symmetry.

4.2. Does ranking using RSA depends on dataset and model size?

We analyze whether RSA based task similarity depends on the model size and amount of training data. Intuitively, it should be independent of model and dataset size, because our method is based on relative similarities. To investigate this, we select a subset of Taskonomy tasks (details in supp. material section S1) and trained smaller models, one per task, with fewer parameters than the models provided by Taskonomy, and on a small subset of Taskonomy data. First, we evaluate if we obtain a similar task clustering using the small models on the selected tasks. Then, for each small model, we compute the similarity score with the pre-trained Taskonomy models on all 20 tasks. The same analysis is repeated with pre-trained Taskonomy model trained on the same task, and we compare the relative similarity based rankings of the small and Taskonomy high-performing models. If the relative rankings of both small and Taskonomy model are similar, then the result suggests that for a completely new task one can train a small model and compute similarity scores to rank them.

4.3. Is RSA related to transfer performance?

We investigate if RSA based task similarity can be applied to transfer learning problem. We first compute the correlation between each column of Taskonomy affinity matrix with RSA matrix after removing the diagonal. As the Taskonomy affinity matrix is populated by raw losses/evaluations, it is indicative of transfer learning performance [34]. We next select a task and dataset different from Taskonomy and obtained the similarity scores of a model trained on the new task with Taskonomy pre-trained models. The pre-trained models were ranked according to the similarity score. We then use the pre-trained models for initializing the model and add the last task dependent layers on top of the initialized model to train on the new task. The

[1] we exclude Jigsaw task as it is unrelated to all other tasks

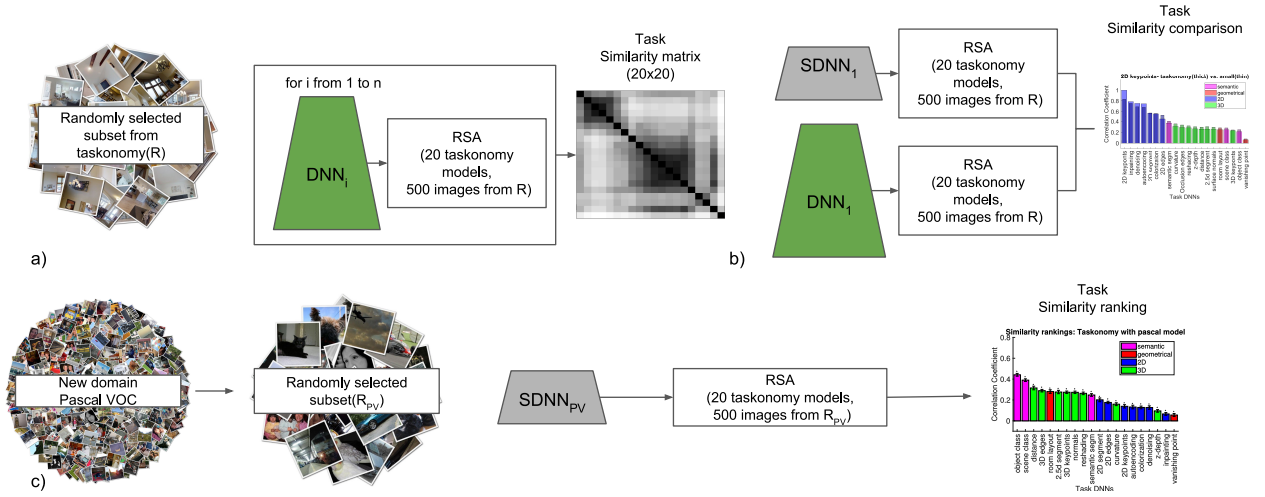

Figure 3. **Our approach: a)** RSA of task-specific pre-trained DNN models (from Taskonomy) to compute a task similarity matrix, **b)** RSA of small model (SDNN) trained on small datasets and comparison with Taskonomy pre-trained models. **c)** RSA of small model (SDNN_{PV}) trained on new task (Pascal VOC semantic segmentation) with Taskonomy pretrained models.

ranking based on the transfer performance is compared with the ranking based on RSA to evaluate the relation between transfer performance and RSA. As we will see in the results, RSA can be used to select the high performing models for transfer learning.

5. Experimental set-up

We first provide the details of datasets used for the experiments, followed by the details of the models' architecture.

5.1. Datasets

Taskonomy dataset It includes over 4 million indoor images from 500 buildings with annotations available for 26 image tasks. 21 of these tasks are single image tasks, and 5 tasks are multi-image tasks. For this work, we select 20 single image task for obtaining task similarities[1].

We randomly selected 500 images from the Taskonomy training dataset as 500 different conditions to perform RSA. These images are used as input to generate representations of different task-specific models to compute the RDMs.

To analyze the dependency of RSA on dataset and model size used for training, we select one building (Hanson) from Taskonomy dataset, which contains 12138 images. We divide them into 10048 training and 2090 validation images.

Pascal VOC semantic segmentation To evaluate the application of RSA in transfer learning, we select the Pascal VOC [9, 12] dataset for semantic segmentation task. It has pixelwise annotations for $10,582$ training images,$1,449$ validation and $1,456$ test images. We argue that this task is different from the Taskonomy semantic segmentation as the images are from a different domain.

5.2. Models

Below, we provide details of the network architectures of pre-trained Taskonomy models, small models trained for Taskonomy tasks, and models used for Pascal VOC.

Taskonomy models The Taskonomy models [2] consist of an encoder and decoder. The encoder for all the tasks is a Resnet-50 [14] model followed by convolution layer that compresses the channel dimension of the encoder output from 2048 to 8. The decoder is task-specific and varies according to the task. For classification tasks and tasks where the output is low dimensional the decoder consists of 2-3 fully connected (FC) layers. For all the other tasks, the decoder consists of 15 layers (except colorization with 12 layers) consisting of convolution and deconvolution layers.

We select the final compressed output of the encoder as the representation for RSA as in [34]. In Taskonomy approach, the compressed output of the encoder was used as an input to transfer function to evaluate the transfer learning performance. Selecting the compressed output of the encoder ensures that the architecture for all the task is the same, and the differences in representation can only arise due to the task that the model was optimized for, as images are also the same for all tasks.

We also explore the representation of earlier layers of the encoder and the task labels as the representation for computing RSA based similarity score. We perform this analysis to investigate how task specificity varies across the depth in the network and if the task's labels are enough to understand the relationship between tasks.

[2] publicly available at https://github.com/StanfordVL/taskonomy

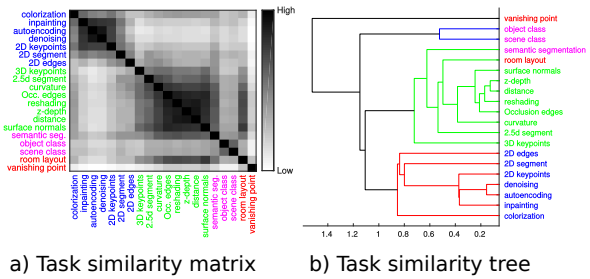

Figure 4. **Task similarity using RSA: a)** Similarity matrix of the 20 Taskonomy tasks, **b)** Agglomerative clustering using RDM.

Small models The smaller version of the models follows a similar style to Taskonomy and consists of an encoder and decoder. The encoder consists of 4 convolution layer each with a stride of 2 to generate a final feature map with the dimensions same as that of Taskonomy encoder. For this experiment, we select the tasks which require a fully-convolution decoder structure and use 4 convolution layers each followed by an upsampling layer. The models are trained on Hanson subset of Taskonomy dataset.

Pascal VOC Models We use two types of models for Pascal VOC semantic segmentation task: 1) a small model to compute similarity score with pre-trained Taskonomy models; 2) models initialized with pre-trained Taskonomy encoders to evaluate transfer learning performance. The small model consists of an encoder and a decoder. The encoder consists of 4 convolution layer each with a stride of 2 to generate a final feature map with the dimensions same as that of Taskonomy encoder. The decoder is an Atrous Spatial Pyramid Pooling (ASPP) [2], which contains convolution layers that operate in parallel with different dilations. The model is trained on Pascal VOC training set with learning rate 10^{-4} for $200,000$ iterations. The encoder representation of the small model trained on Pascal VOC is then used to compute similarity with Taskonomy pre-trained models. The models for evaluating transfer learning performance consists of an encoder with similar architecture as Taskonomy models and an ASPP decoder. The encoder part is initialized by the pre-trained Taskonomy models of the corresponding task.

Implementation and evaluation details We use the publicly available tensorflow implementation [3] of deeplabv3 [3] and modify the code for transfer learning experiments. We use RSA Matlab toolbox [29] for RSA related analysis[4]. We refer to the supplementary material for further details.

6. Results

Here, we present the results of RSA for computing task similarity and its relation to transfer learning performance.

[3]https://github.com/sthalles/deeplab_v3

[4]Code available at https://github.com/kshitijd20/RSA-CVPR19-release

We follow the same nomenclature of task type as in [34], and color code 2D, 3D, semantic, and geometric tasks.

6.1. Task similarity using RSA

Figure 4a shows the similarity matrix of the tasks computed using RSA with the compressed encoder output as the task representation. Recall that we compute the 20×20 similarity matrix using RSA with given task-specific representations for all the randomly selected 500 images. To visualize the relationship between tasks, we applied agglomerative hierarchical clusteringto the similarity matrix. The resulting dendrogram from this clustering is shown in Figure 4b. We can see that the tasks are clustered following visual criteria of 2D, 3D, and semantic tasks.

We further investigate the task similarity using RSA at different depths in the encoder architecture and task labels. Figure 5 shows the task similarity matrix for different depths of the Resnet-50 encoder, namely blocks 1, 2, 3 and 4. We also compare the similarity matrix computed using the tasks' labels. We observe, in Figure 5, that at block 1 all the similarity values are very high implying that at initial layers representations of most of the tasks are similar irrespective of the task type. As we go deeper, the similarity score between tasks starts decreasing, and in compressed encoder output, we can see three dark blocks corresponding to 2D, 3D, and semantic tasks. The above results further validate our choice of using compressed encoder output as the task-specific representation for assessing the similarity between tasks. Interestingly, the clustering using task labels does not group into tasks of the same type, and most of the similarity scores are low. Instead, the labels clustering follows the output structure of the labels, independently of the task type. This is because the labels contain only limited information about the task, and it depends on the annotator criteria on how to represent the output.

We next compare our approach with Taskonomy approach[5]. We use hierarchical clustering to visually compare the dendrograms obtained using both the methods in Figure 6. For quantifying the similarity, we compute the correlation of Taskonomy similarity matrix with RSA similarity matrix (ρ = 0.62, r_s = 0.65). The results show that both approaches group the tasks into similar clusters with few exceptions. Room layout is grouped with the vanishing point in Taskonomy approach and in 3D tasks with our approach. Denoising is clustered with inpainting and autoencoding using our approach, which are related tasks. We argue that our results are plausible.

[5]We show 17 tasks as we had access to only affinity values of these tasks. For comparison with figure 13 in [34], please refer to section S2 of supplementary material

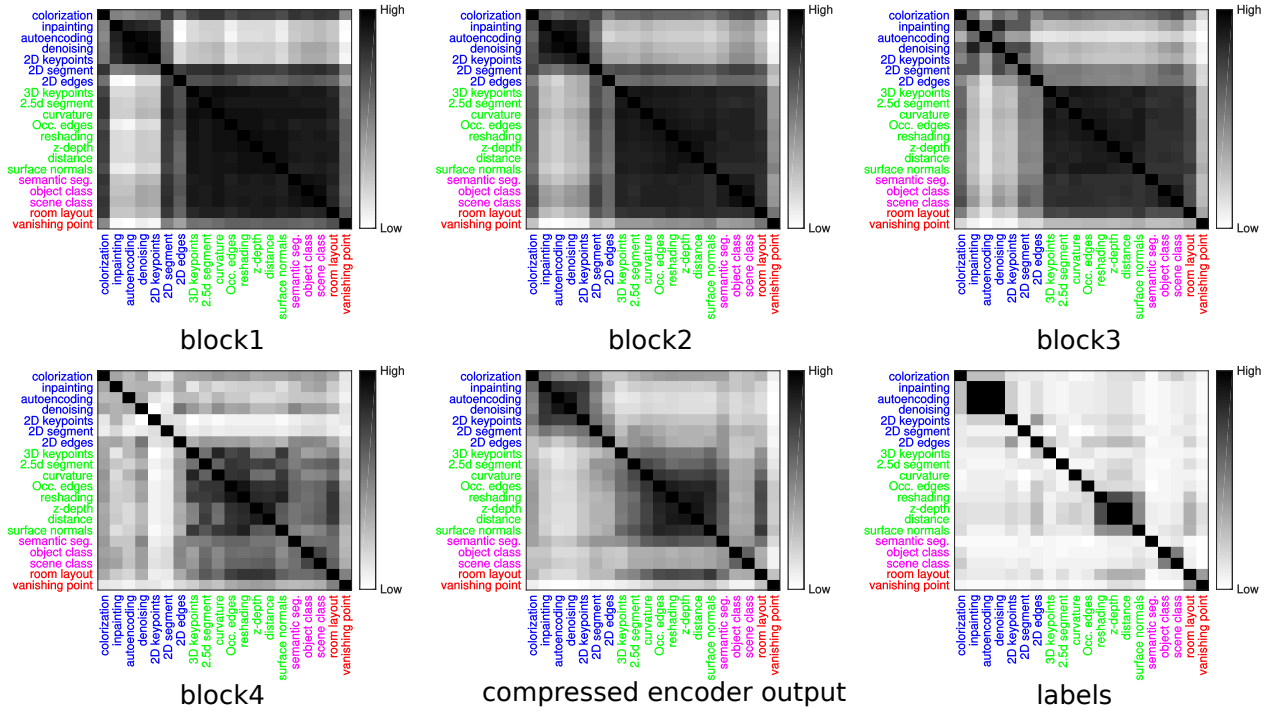

Figure 5. **Task taxonomy using RSA:** $1-5$) Similarity matrix of 20 Taskonomy tasks using features at different depth in the model as task-specific representations 6) Similarity matrix of 20 Taskonomy tasks using labels as task-specific representations.

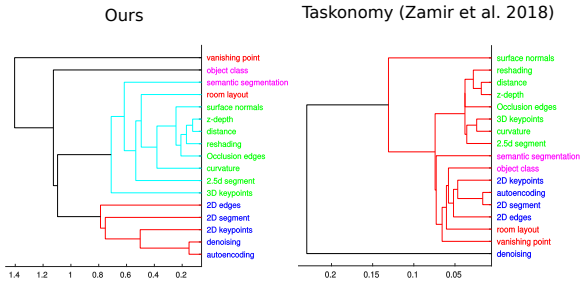

Figure 6. **RSA vs Taskonomy:** Clustering comparison.

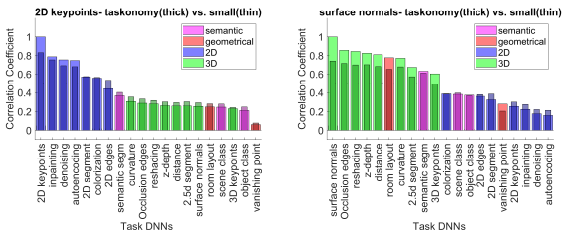

Figure 7. **Task taxonomy using small models:** Similarity ranking of **(a)** keypoint2d Taskonomy model vs small model. **(b)** surface normals Taskonomy model vs small model.

6.2. Does model size impact similarity score?

In this experiment, we investigate how the model and dataset size affect task similarity. We show the results of similarity rankings for 2 tasks: 2D keypoints and surface normals (for other tasks, please see section S1 in supplementary material). We compare the similarity rankings obtained using the small model trained on Hanson subset of Taskonomy data with the Taskonomy model trained on the same task. As we visually observe from the comparison (Figure 7) in both the tasks the ranking look similar. For all the tasks considered in the above comparison the mean correlation is high ($\rho = 0.84$, $r_s = 0.85$).

Next, we also computed task similarity matrices by comparing a small model with small models trained on other tasks. We find that the correlation ($\rho = 0.85$, $r_s = 0.88$) between task similarity matrices (Figure S3) using Taskonomy model and small model is comparable to previous correlation results. The above results together provide strong evidence that the model and dataset size do not have much effect on the similarity score.

6.3. Model selection for transfer learning

We first report the model selection using RSA for Taskonomy tasks and then on Pascal VOC semantic seg-

Top-1	Top-3	Top-5
7/17	14/17	16/17

Table 1. Number of tasks for which best model selected for transfer learning using RSA is in top-n models according to transfer performance for 17 tasks

mentation task.

Taskonomy We obtain high mean correlation (ρ = 0.70, r_s = 0.76) between RSA and transfer learning for 17 tasks from the Taskonomy dataset. We also report in Table 1 that for 16 out of 17 tasks, the best model selected by RSA for transfer learning is in top-5 models selected using Taskonomy approach (transfer learning performance).

Pascal VOC We show the relation of similarity score using RSA with transfer learning by selecting a new task (semantic segmentation in Pascal VOC). We compare the transfer learning performance of models initialized by different task-specific pre-trained models from Taskonomy dataset. Then we compare the transfer learning performance based ranking with similarity score ranking. Here we select the small Pascal model to compute the similarity with the Taskonomy models. We report the robustness of similarity ranking using RSA with respect to model size, number of images used for RSA analysis, and different training stages in supplementary section S3.

We show the similarity score based ranking in Figure 8. Surprisingly, semantic segmentation model from Taskonomy shows a lower similarity score as compared to other models trained on semantic (scene class, object class) and 3D tasks (occlusion edges, surface normals). Most of the 2D tasks show low similarity scores.

To investigate if similarity scores are related to transfer learning performance we evaluated the models initialized with task-specific Taskonomy models, finetuned with Pascal VOC training set, and compared the performance on Pascal VOC test set. Table 1 shows the comparison of transfer learning performance for models with initialization from a set of selected tasks (For a complete comparison refer to section S3 in the supplementary material). The tasks are listed in the order of their similarity scores. We note from the table that the tasks on the top (object class, scene class, occlusion edges, and semantic segmentation) shows higher performance while autoencoder and vanishing point performance is even less than model trained from scratch (random in Table 2). We note that our results are comparable to the results (64.81%) reported in [3], when they use Resnet-50 trained on Imagenet for initialization. The results provide evidence that the similarity score obtained using RSA provide an estimate of the expected transfer performance.

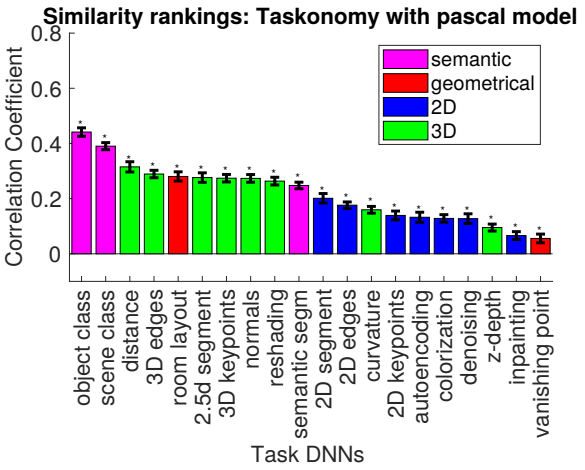

Figure 8. RSA based similarity of scores of pre-trained Taskonomy models with the small model trained on Pascal VOC.

Initialization(Task)	mIoU
Object class	0.6492
Scene class	0.6529
Occlusion edges	0.6496
Semantic segmentation	0.6487
Autoencoder	0.5901
Vanishing point	0.5891
Random(Taskonomy encoder)	0.6083
Random(Small encoder)	0.4072

Table 2. Transfer learning performance on Pascal VOC test set.

7. Conclusion

We presented an efficient alternative approach to obtain the similarity between computer vision models trained on different tasks using their learned representations. Our approach uses RSA, and it is suitable for obtaining task similarity by just using the pre-trained models without any further training, as opposed to the earlier state of the art method Taskonomy for this problem.

We provided strong evidence that for obtaining the similarity, the model and training dataset size does not play a significant role and we can obtain a task similarity relative ranking using small models as well as state of the art models with few data samples. This comes with computational and memory savings.

We also showed the relationship of the task similarity using RSA with the transfer learning performance and its applicability. We demonstrated on both, Taskonomy and Pascal VOC semantic segmentation, that the transfer learning performance is closely related to the similarity obtained with RSA. The above results showed that for domain shift the model trained on the same task might not be the best fit for transfer learning and our proposed approach can help in

model selection for transfer learning. Our method is applicable to a wide range of potential problems, such as multi-task models, architecture selection.

Acknowledgements This work was funded by the SUTD-MIT IDC grant (IDG31800103). K.D. was also funded by SUTD Presidents Graduate Fellowship. We thank Taskonomy authors for the support and the code.

References

[1] Michael F Bonner and Russell A Epstein. Computational mechanisms underlying cortical responses to the affordance properties of visual scenes. *PLOS Computational Biology*.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[4] Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific Reports*, 6(June):1–13, 2016.

[5] Radoslaw Martin Cichy, Dimitrios Pantazis, and Aude Oliva. Resolving human object recognition in space and time. *Nature neuroscience*, 17(3):455, 2014.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] Thanuja Dharmasiri, Andrew Spek, and Tom Drummond. Joint prediction of depths, normals and surface curvature from rgb images using cnns. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 1505–1512. IEEE, 2017.

[8] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[11] Iris IA Groen, Michelle R Greene, Christopher Baldassano, Li Fei-Fei, Diane M Beck, and Chris I Baker. Distinct contributions of functional and deep neural network features to representational similarity of scenes in human brain and behavior. *Elife*, 7:e32962, 2018.

[12] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 991–998. IEEE, 2011.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[16] Seyed Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Computational Biology*, 10(11), 2014.

[17] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory.

[18] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:4, 2008.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[22] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[24] Arun Mallya and Svetlana Lazebnik. Piggyback: Adding multiple tasks to a single, fixed network by learning to mask. *arXiv preprint arXiv:1801.06519*, 2018.

[25] Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, and Aude Oliva. Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. *NeuroImage*, 153:346–358, 2017.

[26] Patrick McClure and Nikolaus Kriegeskorte. Representational distance learning for deep neural networks. *Frontiers in computational neuroscience*, 10:131, 2016.

[27] Johannes Mehrer, Nikolaus Kriegeskorte, and Tim Kietzmann. Beware of the beginnings: intermediate and higher-

level representations in deep neural networks are strongly affected by weight initialization. In *Conference on Cognitive Computational Neuroscience*, 2018.

[28] Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Tom Yan, Alex Andonian, Kandan Ramakrishnan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding.

[29] Hamed Nili, Cai Wingfield, Alexander Walther, Li Su, William Marslen-Wilson, and Nikolaus Kriegeskorte. A toolbox for representational similarity analysis. *PLoS computational biology*, 10(4):e1003553, 2014.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[31] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

[34] Amir R Zamir, Alexander Sax, and William Shen. Taskonomy: Disentangling task transfer learning.

[35] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Supplementary materials: Representation Similarity Analysis for Efficient Task Taxonomy & Transfer Learning

Kshitij Dwivedi Gemma Roig
Singapore University of Technology and Design
kshitij_dwivedi@mymail.sutd.edu.sg, gemma_roig@sutd.edu.sg

Here we report the additional details and results which we left in the main text to the supplementary material. In the first section, we provide details about the small models used and report the results and comparison with the Taskonomy pretrained models. In the second section, we compare the task similarity matrix and clustering using our RSA approach with that of Taskonomy[34] approach. In the third section, we report the consistency of RSA based similarity ranking and transfer learning performance for all the tasks.

S1. Small models for task taxonomy

We select the tasks (a total of 14 tasks) which can be optimized using only L1/L2/triple-metric loss and the output of the task is spatial such that all the tasks can have the same decoder except the final layer. The architecture of the small model is reported in Table S1.

We show the task similarity comparison results (Figure S1) of all the selected tasks. We note that for most of the 2D tasks the correlation (Pearson's ρ) of similarity rankings between small vs. Taskonomy models is very high (>0.97 except segment2d) and visually look similar. Although the correlation for all the 3D tasks is still high (>0.77), correlation values are relatively lower than 2D tasks.

We also evaluated the predicted output of 3D tasks and 2D tasks visually. We observed that for the tasks where the predicted output looks more similar to the target, the correlation is higher (Figure S2). The difference in correlation could also be attributed to different training setting of Taskonomy and small models as it was not possible to exactly replicate the Taskonomy training with small models because the training code is not publicly available, and the small models are trained using only a subset of the whole dataset. We computed the task similarity matrix for the selected tasks using both small models and Taskonomy models. Although the similarity ranking using small models on 3D task did not show as high correlation with the Taskonomy models, we found that the Pearson's correlation between them is high (0.8510). On visual inspection of both similarity matrices (Figure S3), 2D tasks of small models show similar scores as with Taskonomy models. The 3D tasks although show higher similarity with corresponding 3D tasks rather than 2D tasks but similarity scores within 3D tasks are lower and therefore matrix looks lighter as compared to the similarity matrix with Taskonomy models.

Layer	Kernel size	# Channels	Stride
Encoder			
Conv1	3×3	16	2
Conv2	3×3	32	2
Conv3	3×3	64	2
Conv4	3×3	64	2
Conv5	3×3	8	1
Decoder			
Conv6	3×3	32	1
$Upscale \times 2$			
Conv7	3×3	16	1
$Upscale \times 2$			
Conv8	3×3	4	1
$Upscale \times 2$			
Conv9	3×3	4	1
$Upscale \times 2$			
Conv10	3×3	n	1

Table S1. Small model architecture.The number of channel in Conv10 n was task-specific

S2. Taskonomy[34] vs RSA(Our approach)

We show the clustering obtained using Taskonomy approach and compare it our approach in Figure S4. From the figure, we observe that almost all of the 20 single image task we select for our paper (except room layout and denoise) belong in the same cluster as using Taskonomy approach. It is also possible that the difference in clustering arises due to

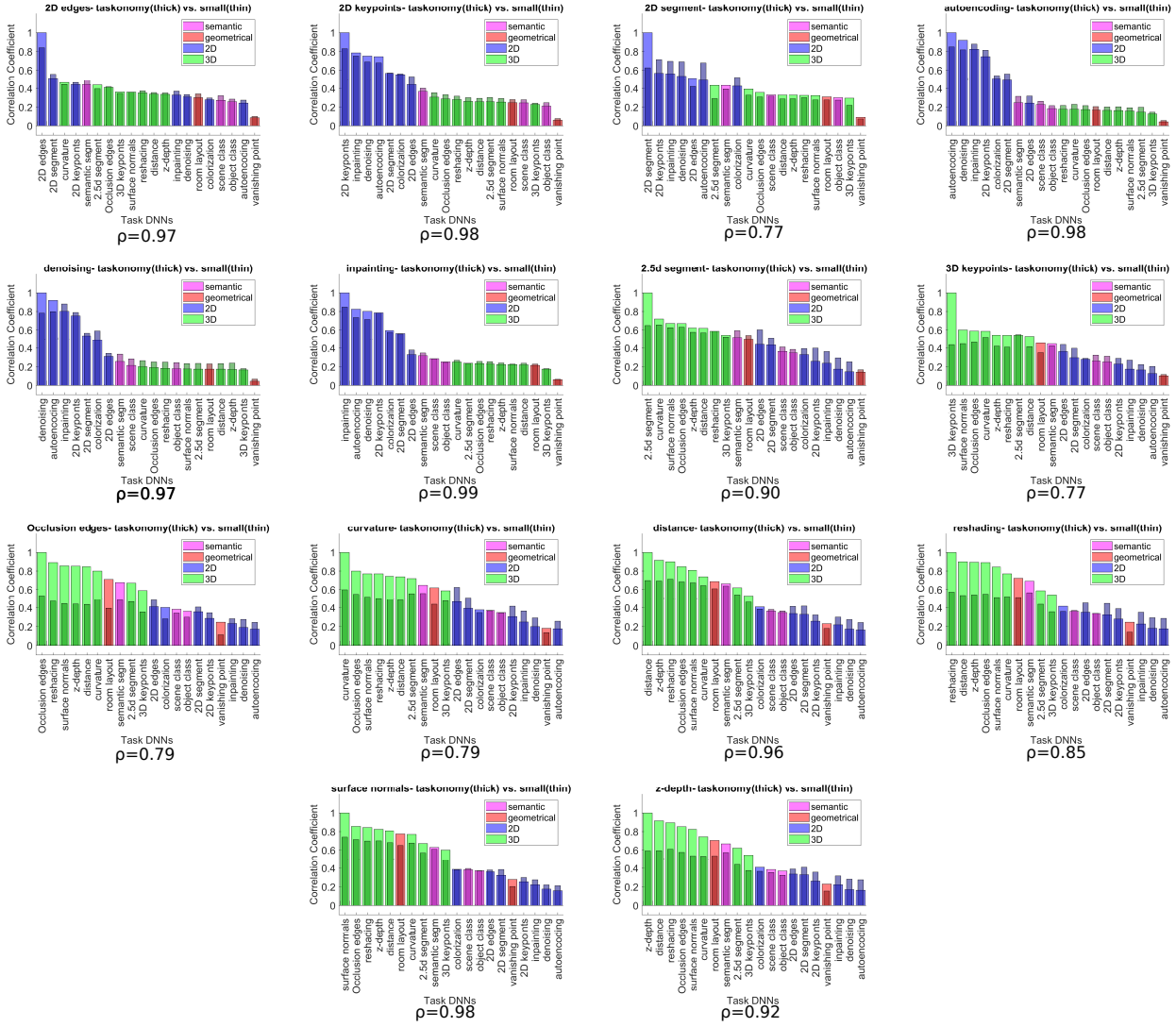

Figure S1. **Similarity ranking with taskonomy model vs small models for 14 tasks.** The ρ value below each plot specifies the Pearson's correlation coefficient between the two similarity rankings.

different clustering method, which was not specified, used in [34].

One other advantage of our approach over Taskonomy is that our similarity scores lie between -1 and 1 and thus similarity matrix is easy to visualize and evaluate. In Taskonomy approach, an exponential scaling of the similarity score has to be performed to bring them in a good range for visualization. Figure S5 shows both the similarity matrix without any scaling.

S3. Transfer learning in Pascal VOC

In the first three subsections below, we show the consistency of RSA with varying number of iterations, the model size, and the number of images selected for RDM computation. In the last subsection, we report the transfer learning performance of all the task DNNs used for initialization.

S3.1. Consistency with training stage

We show in Figure S6 that even at $1/10$ of the final training stage the Pearson's correlation with the final stage is 0.88 and after $1/2$ of the training the correlation with the final stage stays above 0.99. This shows that one can also use models from an early stage of training for task similarity using RSA.

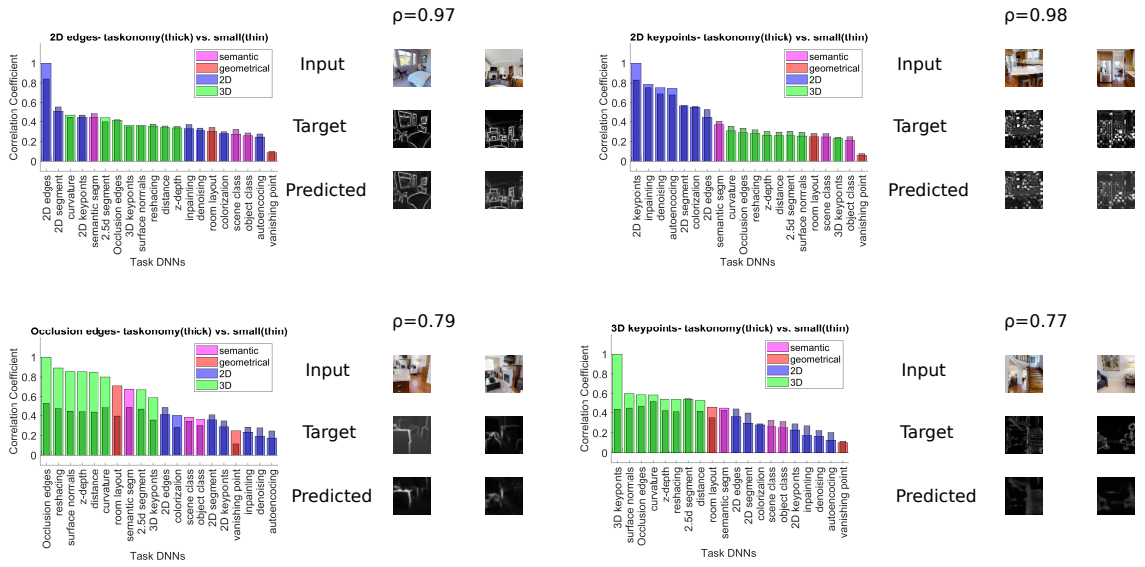

Figure S2. **Is correlation related to visual similarity of the predicted output with the target?**

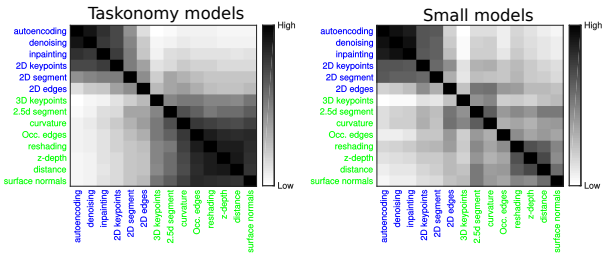

Figure S3. **Task similarity matrix using Taskonomy models vs small models.**

S3.2. Consistency with model size

We show in Figure S7 the comparison of task similarity obtained using a small encoder (thin bars) vs. task similarity obtained using taskonomy encoder architecture (thick bars). A high correlation (ρ = 0.95, r_s = 0.96) suggests that we can use small models to train on a new task and use RSA to select a good model for initialization.

S3.3. Consistency with the number of images

We varied the number of images from 100 to 2000 and plot the Pearson's correlation of task similarity ranking obtained using n images with the task similarity ranking obtained using 2000 images (Figure S8). After 400 images the Pearson's correlation with the task similarity ranking is always above 0.99, thus suggesting that around 500 images are sufficient for RDM computation.

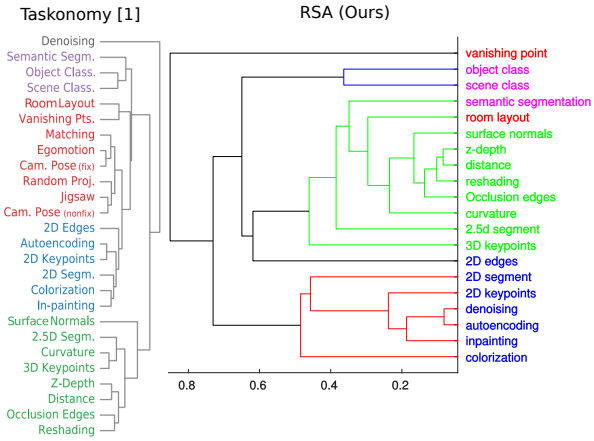

Figure S4. **Clustering: Taskonomy vs RSA (Ours)** Image source: Figure 13 from [34]

S3.4. Transfer learning performance for all the tasks

Figure S9 shows the transfer learning performance (mIoU) for 17 single image tasks [1] in the descending order of similarity rankings. The curve shows that the performance in most of the tasks seems to decrease as the similarity score decreases (although it is not a perfect monotonically decreasing curve).Also, generally the tasks with

[1] We ignore denoise, autoencoding, and colorization as these tasks require modified input

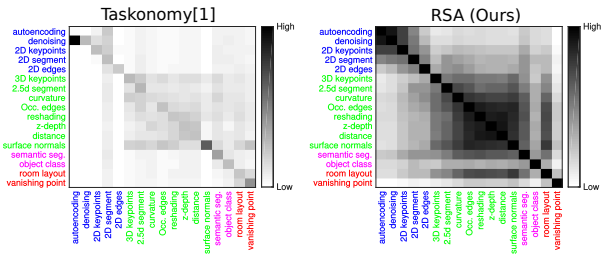

Figure S5. **Similarity matrix: Taskonomy vs RSA(Ours)**

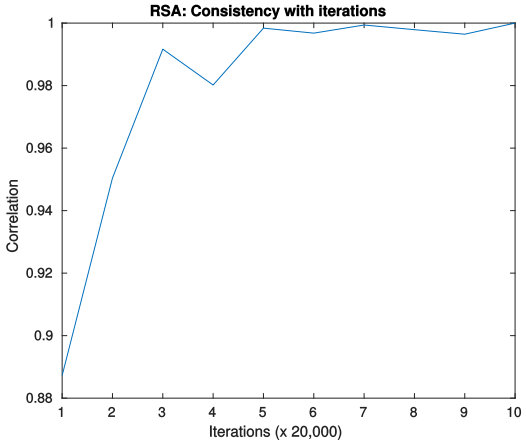

Figure S6. **Consistency with training iterations**

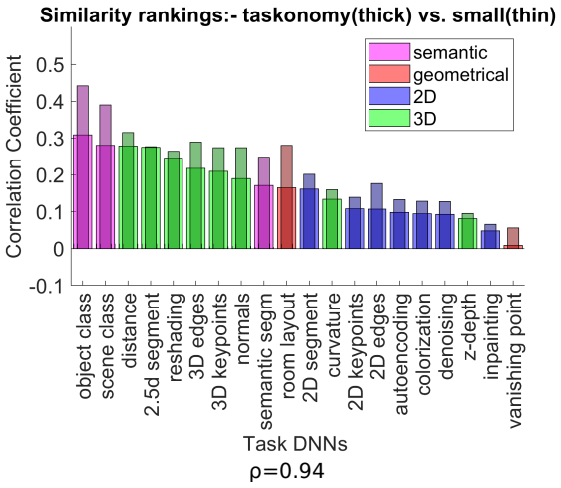

Figure S7. **Consistency with model size**

higher similarity ranking (object class, surface normals, segment25d) showed high transfer learning performance, and tasks with lower similarity score (autoencoding, vanishing point) showed lower performance.

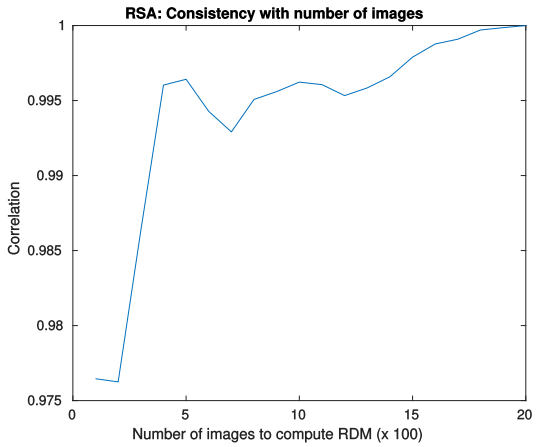

Figure S8. **Consistency with number of images**

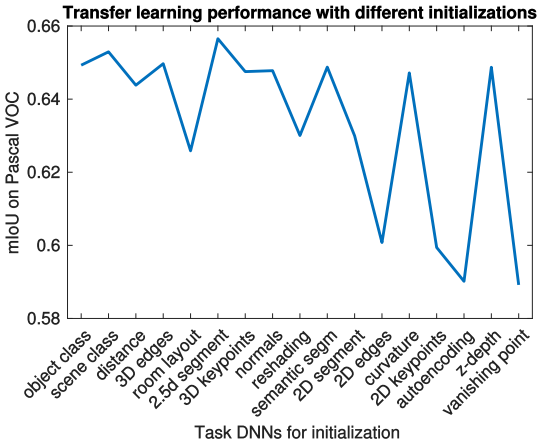

Figure S9. **Transfer learning performance in descending order of similarity scores with task DNNs on the x-axis as initialization**

2.2 Duality diagram similarity: a generic framework for initialization selection in task transfer learning

Duality Diagram Similarity: a generic framework for initialization selection in task transfer learning

Kshitij Dwivedi[1,3][0000−0001−6442−7140], Jiahui Huang[2][0000−0002−0389−1721], Radoslaw Martin Cichy[3][0000−0003−4190−6071], and Gemma Roig[1][0000−0002−6439−8076]

[1] Department of Computer Science, Goethe University Frankfurt, Germany
kshitijdwivedi93@gmail.com, roig@cs.uni-frankfurt.de
[2] ISTD, Singapore University of Technology and Design, Singapore
jiahui_huang@sutd.edu.sg
[3] Department of Education and Psychology, Free University Berlin, Germany
rmcichy@zedat.fu-berlin.de

Abstract. In this paper, we tackle an open research question in transfer learning, which is selecting a model initialization to achieve high performance on a new task, given several pre-trained models. We propose a new highly efficient and accurate approach based on duality diagram similarity (DDS) between deep neural networks (DNNs). DDS is a generic framework to represent and compare data of different feature dimensions. We validate our approach on the Taskonomy dataset by measuring the correspondence between actual transfer learning performance rankings on 17 taskonomy tasks and predicted rankings. Computing DDS based ranking for 17×17 transfers requires less than 2 minutes and shows a high correlation (0.86) with actual transfer learning rankings, outperforming state-of-the-art methods by a large margin (10%) on the Taskonomy benchmark. We also demonstrate the robustness of our model selection approach to a new task, namely Pascal VOC semantic segmentation. Additionally, we show that our method can be applied to select the best layer locations within a DNN for transfer learning on 2D, 3D and semantic tasks on NYUv2 and Pascal VOC datasets.

Keywords: Transfer Learning, Deep Neural Network Similarity, Duality Diagram Similarity, Representational Similarity Analysis

1 Introduction

Deep Neural Networks (DNNs) are state-of-the-art models to solve different visual tasks, *c.f.* [18,42]. Yet, when the number of training examples with labeled data is small, the models tend to overfit during training. To tackle this issue, a common approach is to use transfer learning by selecting a pre-trained network on a large-scale dataset and use it as initialization [28,19]. But how does one choose the model initialization that yields the highest accuracy performance when learning a new task?

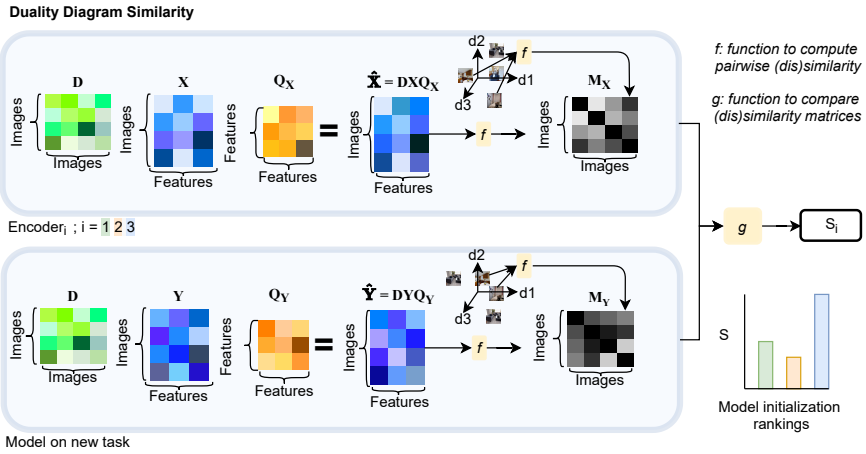

Fig. 1. *Duality Diagram Similarity (DDS):* We apply DDS to compare features of a set of initialization options (encoders) with features of a new task to get model initialization rankings to select the encoder initialization for learning a new task. The task feature for an image is obtained by doing a feedforward pass through a model trained on that task. $\mathbf{D}$ is the matrix that weights the images, $\mathbf{X}$ ($\mathbf{Y}$) is the matrix that stores the features from the encoder for all images, $\mathbf{Q_X}$ ($\mathbf{Q_Y}$) is a matrix that stores relations between features dimensions, $\mathbf{M_X}$ ($\mathbf{M_Y}$) contains the pairwise (dis)similarity distances between images, and S_i is the score for the ranking.

Nowadays, there are a plethora of online available pre-trained models on different tasks. However, there are only a few methods [8,35] that automatically assist in selecting an initialization given a large set of options. Due to lack of a standard benchmark with standard evaluation metrics, comparing and building upon these methods is not trivial. Recently, Dwivedi and Roig [8] and Song *et al.* [35] used the transfer learning performance on the Taskonomy dataset [42] as groundtruth to develop methods for model selection. Both aforementioned methods for model selection are efficient compared to the bruteforce approach of obtaining transfer performance from all the models and selecting the best one. Yet, they used different metrics to evaluate against the groundtruth, and hence, they are not comparable in terms of accuracy. Although different, both of them used metrics that evaluate how many models in top-K ranked model initializations according to transfer learning performance were present in the top-K ranked models obtained using their method. We argue that such a metric doesn't provide a complete picture as it ignores the ranking within the top-K models as well as the ranking of models not in the top-K.

In this work, we first introduce a benchmark with a standard evaluation metric using Taskonomy [42] transfer learning dataset to compare different model initialization selection methods. We use Spearman's correlation between the rankings of different initialization options according to transfer learning per-

formance and the rankings based on a model initialization selection method as our metric for comparison. We argue that our proposed benchmark will facilitate the comparison of existing and new works on model selection for transfer learning. We then introduce a duality diagram [9,12,6] based generic framework to compare DNN features which we refer to as duality diagram similarity (DDS). Duality diagram expresses the data taking into account the contribution of individual observations and individual feature dimensions, and the interdependence between observations as well as feature dimensions (see Fig. 1). Due to its generic nature, it can be shown that recently introduced similarity functions [8,17] for comparing DNN features are special cases of the general DDS framework.

We find that model initialization rankings using DDS show very high correlation (>0.84) with transfer learning rankings on Taskonomy tasks and outperform state-of-the-art methods [8,35] by a 10% margin. We also demonstrate the reliability of our method on a new dataset and task (PASCAL VOC semantic segmentation) in the experiments section.

Previous works [41,22] have shown the importance of selecting which layer in the network to transfer from. In this paper, we also explore if the proposed method could be used to interpret representations at different depths in a pre-trained model, and hence, it could be used to select from which layer the initialization of the model should be taken to maximize transfer learning performance. We first show that the representation at different blocks of pre-trained ResNet [11] model on ImageNet [7] varies from 2D in block1, to 3D in block 3 and semantic in block 4. These observations suggest that representation at different depths of the network is suitable for transferring to different tasks. Tranfer learning experiments using different blocks in a ResNet-50 trained on ImageNet as initialization for 2D, 3D, and semantic tasks on both, NYUv2 [23] and Pascal VOC [10] datasets, reveal that it is indeed the case.

2 Related Works

Our work relies on comparing DNN features to select pre-trained models as initialization for transfer learning. Here, we first briefly discuss related literature in transfer learning, and then, different methods to compare DNN features.

Transfer Learning. In transfer learning [25] the representations from a source tasks are re-used and adapted to a new target task. While transfer learning in general may refer to task transfer[28,19,40,42], or domain adaptation[30,37], in this work we focus specifically on task transfer learning. Razavian *et al.* [32] showed that features extracted from Overfeat [31] network trained on ImageNet [7] dataset can serve as a generic image representation to tackle a wide variety of recognition tasks. ImageNet pre-trained models also have been used to transfer to a diverse range of other vision related tasks [28,5,19,40]. Other works [17,14] have investigated why ImageNet trained models are good for transfer learning. In contrast, we are interested in improving the transfer performance by finding a better initialization that is more related to the target task.

Azizpour *et al.* [1] investigated different transferability factors. They empirically verified that the effectiveness of a factor is highly correlated with the distance between the source and target task distance obtained with a predefined categorical task grouping. Zamir *et al.* [42] showed in a fully computational manner that initialization matters in transfer learning. Based on transfer performance they obtained underlying task structure that showed clusters of 2D, 3D, and semantic tasks. They introduced the Taskonomy dataset [42], which provides pre-trained models on over 20 single image tasks with transfer learning performance on each of these tasks with every pre-trained model trained on other tasks as the initialization, and thus, providing groundtruth for a large number of transfers. Recent works [8,35] have used the Taskonomy transfer performance as groundtruth to evaluate methods of estimating task transferabilities. Those works use different evaluation metrics, which makes the comparison between those methods difficult. Following those works, we use Taskonomy transfer performance as a benchmark, and propose a unified evaluation framework to facilitate comparison between existing and future methods.

Yosinski *et al.* [41] explored transferability at different layers of a pre-trained network, and Zhuo *et al.* [44] showed the importance of focusing on convolutional layers of the model in domain adaptation. We also investigate if the similarity between DNN representations can be applied to both model and layer selection for transfer learning, which indeed is the case, as we show in the results section.

Similarity Measures for Transfer Learning Performance. Our approach is built under the assumption that the higher the similarity between representations is, the higher will be the transfer learning performance. Some previous works used similarity measures to understand the properties of DNNs. Raghu *et al.* [26] proposed affine transform invariant measure called Singular Vector Canonical Correlation Analysis (SVCCA) to compare two representations. They applied SVCCA to probe the learning dynamics of neural networks. Kornblith *et al.* [16] introduced centered kernel alignment (CKA) that shows high reliability in identifying correspondences between representations in networks trained using different initializations. However, in the above works, the relation between similarity measures and transfer learning was not explored.

Dwivedi and Roig [8] showed that Representational Similarity Analysis (RSA) can be used to compare DNN representations. They argued that using the model parameters from a model that has a similar representation to the new task's representation as initialization, should give higher transfer learning performance compared to an initialization from a model with a lower similarity score. Recently, Song *et al.* [35] used attribution maps [34,2,33] to compare two models and showed that it also reflects transfer learning performance. Our work goes beyond the aforementioned ones. Besides proposing an evaluation metric to set up a benchmark for comparison of these methods, we introduce a general framework using duality diagrams for similarity measures. We show that similarity measures, such as RSA and CKA, can be posed as a particular case in our gen-

Distances	Pearson's: $1 - \frac{(\mathbf{x}_i - \overline{\mathbf{x}_i}) \cdot (\mathbf{x}_j - \overline{\mathbf{x}_j})}{\|\|\mathbf{x}_i - \overline{\mathbf{x}_i}\|\| \cdot \|\|\mathbf{x}_j - \overline{\mathbf{x}_j}\|\|}$	Euclidean: $\sqrt{\mathbf{x}_i^T.\mathbf{x}_i + \mathbf{x}_j^T.\mathbf{x}_j - 2 * \mathbf{x}_i^T.\mathbf{x}_j}$	cosine: $1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\|\mathbf{x}_i\|\| \cdot \|\|\mathbf{x}_j\|\|}$
Kernels	linear: $\mathbf{x}_i^T \mathbf{x}_j$	Laplacian: $\exp(-\gamma_1 \|\|\mathbf{x}_i - \mathbf{x}_j\|\|_1)$	RBF: $\exp(-\gamma_2 \|\|\mathbf{x}_i - \mathbf{x}_j\|\|^2)$

Table 1. *Distance and Kernel functions used in DDS.* Notation: $\mathbf{x}_i \in \mathbb{R}^{d1}$ and $\mathbf{x}_j \in \mathbb{R}^{d1}$ refer to the features corresponding to i^{th} and j^{th} image (i^{th} and j^{th} row of feature matrix $\mathbf{X}$), respectively. Here, γ_1 and γ_2 refer to the bandwidth of Laplacian and RBF kernel.

eral formulation. It also allows to use other more powerful similarities that are more highly correlated to transfer learning performance.

There is evidence in the deep learning literature, that normalization plays a crucial role. For instance, batch normalization allows training of deeper networks [15], efficient domain adaptation [20,3] and parameter sharing across multiple domains [27]. Instance normalization improves the generated image quality in fast stylization [38,13], and group normalization stabilizes small batch training [39]. In our DDS generic framework, it is straightforward to incorporate feature normalization. Thus, we further take into account the normalization of features before assessing the similarity between two DNN features and compare it to transfer learning performance.

3 Duality Diagram Similarity (DDS)

The term duality diagram was introduced by Escoufier [9] to derive a general formula of Principal Component Analysis that takes into account change of scale, variables, weighing of feature dimensions and elimination of dependence between samples. With similar motivation, we investigate the application of duality diagrams in comparing two DNNs. Let $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ refer to a matrix of features with dimensionality d_1 obtained from feedforwarding n images through a DNN. The duality diagram of matrix $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ is a triplet ($\mathbf{X}$,$\mathbf{Q}$,$\mathbf{D}$) consisting of a matrix $\mathbf{Q} \in \mathbb{R}^{d_1 \times d_1}$ that quantifies dependencies between the individual feature dimensions, and a matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ that assigns weights on the observations, *i.e.*, images in our case. Hence, a DNN representation for a set of n examples can be expressed by its duality diagram. By comparing duality diagrams of two DNNs we can obtain a similarity score. We denote the two DNN duality diagrams as ($\mathbf{X}$,$\mathbf{Q_X}$,$\mathbf{D}$) and ($\mathbf{Y}$,$\mathbf{Q_Y}$,$\mathbf{D}$), in which the subindices in the matrix $\mathbf{Q}$ denote that they are computed from the set of features and images in $\mathbf{X}$ and $\mathbf{Y}$.

To compare two duality diagrams, Robert and Escoufier [29] introduced the RV coefficient. The motivation behind RV coefficient was to map n observations of $\mathbf{X}$ in the d_1-dimensional space and $\mathbf{Y}$ in the d_2-dimensional space. Then, the similarity between $\mathbf{X}$ and $\mathbf{Y}$ can be assessed by comparing the pattern of obtained maps or, equivalently, by comparing the set of distances between all pairwise observations of both maps. To estimate the distances between pair-

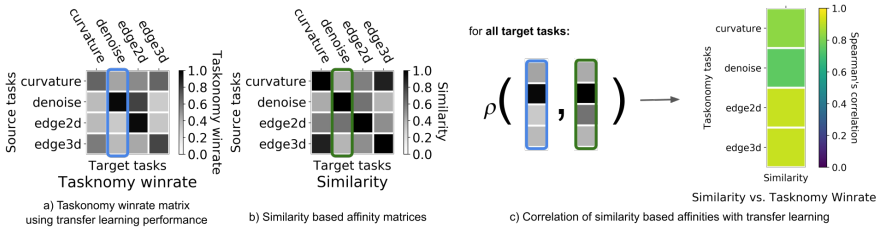

Fig. 2. *Transfer learning vs. similarity measures.* We consider a) Taskonomy winrate matrix, b) an affinity matrix obtained by measuring similarity between DNNs trained on different tasks. c) the Spearman's correlation (denoted by ρ) between the columns of two matrices. The resulting vector shows the correlation of the similarity based rankings with transfer learning performance based rankings for 4 Taskonomy tasks. Here we illustrate the results using DDS ($f = Laplacian$), and the procedure remains the same using any similarity measure.

wise observation, Robert and Escoufier [29] used dot product and compared two (dis)similarity matrices using the cosine distance.

In a nutshell, to compare two sets of DNN features $\mathbf{X}$ and $\mathbf{Y}$, we require three steps (Fig. 1): first, transforming the data using $\mathbf{Q_X}$ and $\mathbf{D}$ to $\hat{\mathbf{X}}$, using $\hat{\mathbf{X}} = \mathbf{DXQ_X}$, and $\hat{\mathbf{Y}}$ with $\hat{\mathbf{Y}} = \mathbf{DYQ_Y}$. Second, using a function, which we denote as f, to measure (dis)similarity between each pair of data points to generate pairwise distance maps. Let $\mathbf{M_X}$ be the matrix that stores the (dis)similarity between pairwise distance maps for $\hat{\mathbf{X}}$, also referred to as representational (dis)similarity matrices. It is computed as $\mathbf{M_X}(i,j) = f(\hat{\mathbf{X}}(i,:), \hat{\mathbf{X}}(j,:))$, in which i and j denote the indices of the matrices. Analogously, $\mathbf{M_Y}$ is the matrix that stores the (dis)similarity between pairwise distance maps of $\hat{\mathbf{Y}}$. Third, a function g to compare $\mathbf{M_X}$ and $\mathbf{M_Y}$ to obtain a final similarity score, denoted as S, and computed as $S = g(\mathbf{M_X}, \mathbf{M_Y})$ is applied. The above formulation using duality diagrams provides a general formulation that allows us to investigate empirically which combination of $\mathbf{Q}$,$\mathbf{D}$, f and g is suitable for a given application, which in our case is estimating transferability rankings to select the best model (or layer in a model) to transfer given a new dataset and/or task.

Interestingly, using the above DDS framework, we can easily show that recently used similarity measures, e.g., CKA and RSA, can be formulated as special cases of DDS. For RSA [8], $\mathbf{Q}$ is an identity matrix, $\mathbf{I} \in \mathbb{R}^{d_1 \times d_1}$, and $\mathbf{D}$ is a centering matrix, *i.e.*, $\mathbf{C} = \mathbf{I_n} - \frac{1}{n}\mathbf{1_n}$. f is Pearson's distance and g is Spearman's correlation between lower/upper triangular part of $\mathbf{M_X}$ and $\mathbf{M_Y}$. For CKA [17], $\mathbf{Q}$ and $\mathbf{D}$ are identity matrices $\mathbf{I} \in \mathbb{R}^{d_1 \times d_1}$ and $\mathbf{I} \in \mathbb{R}^{n \times n}$ respectively, f used is linear or RBF kernel and g is cosine distance between unbiased centered (dis)similarity matrices. In the supplementary section S1, we derive RSA and CKA as particular cases of the DDS framework.

In this work, we focus on exploring different instantiations of $\mathbf{Q}$, $\mathbf{D}$, f and g from our DDS framework that are most suitable for estimating transfer learning

performance. We consider different formulations of **Q** and **D**, resulting in z-scoring, batch normalization, instance normalization, layer normalization and group normalization (details in Supplementary S2). For function f we explore cosine, Euclidean, and Pearson's distance, as well as kernel based similarities, namely linear, RBF, and Laplacian. Mathematical equations for all functions are in Table 1. For function g, we consider Pearson's correlation to compare (dis)similarity matrices with and without unbiased centering [36,17].

4 Our Approach

4.1 Which DDS combination (Q, D, f ,g) best predicts transferability?

After having defined the general formulation for using similarity measures for transfer learning, we can instantiate each of the parameters (**Q**, **D**, f and g) to obtain different similarity measures. To evaluate which combination of **Q**, **D**, f and g best predicts transferability and compare it to state-of-the-art methods, we consider transfer learning performance based winrate matrix (Fig. 2a) and affinity matrix proposed in Taskonomy dataset [42], as a transferability benchmark. The affinity matrix is calculated by using actual transfer learning performance on the target task given multiple source models pre-trained on different tasks. The winrate matrix is calculated using a pairwise competition between all feasible sources for transferring to a target task. Both these matrices represent transfer learning performance obtained by bruteforce, and hence, can be considered as an upper bound for benchmarking transferability. We use the Taskonomy dataset as a benchmark as it consists of pre-trained models on over 20 single image tasks with transfer learning performance on each of these tasks with every pre-trained model trained on other tasks as the initialization, thus, providing groundtruth for a large number of task transfers.

We use DDS to quantify the similarity between two models trained on different Taskonomy tasks and use that value to compute the DDS based affinity matrix (Fig. 2b). A column vector corresponding to a specific task in the Taskonomy affinity matrix shows the transfer learning performance on the target task when different source tasks were used for initialization. To evaluate how well a DDS based affinity matrix represents transferability, we calculate the Spearman's correlation between columns of the Taskonomy winrate/affinity matrix and DDS based affinity matrix. Using the rank-based Spearman's correlation for comparison between two rankings allows comparing the source tasks ranking on the basis of transfer learning performance with DDS based ranking. The resulting vector (Fig. 2c) represents the per task correlation of DDS with transferability.

We further evaluate if the best combination(s) we obtained from the above proposed evaluation benchmark using Taskonomy are robust to a new dataset and task. For this, we consider a new task, Pascal VOC semantic segmentation, following [8]. For the benchmark, we use the transfer learning performance on Pascal VOC semantic segmentation task given all Taskonomy models as sources.

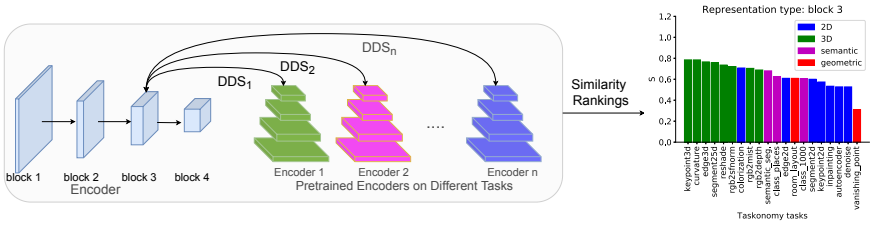

Fig. 3. *DNN Layer Selection.* Given a pre-trained encoder and a set of pre-trained models trained on diverse tasks, we can assess the representation type at different depth of the network by comparing the similarity between features at a given depth and pre-trained models.

We also investigate if the images selected to compute DDS have any effect on Spearman's correlation with transfer learning. For this purpose, we select images from NYUv2, Taskonomy, and Pascal VOC dataset and evaluate the proposed methods on both, Taskonomy and Pascal VOC benchmark. We further compute the variance performing bootstrap by randomly sampling 200 images from the same dataset 100 times to compute similarity. The bootstrap sampling generates a bootstrap distribution of correlation between transfer performance and similarity measures, which allows measuring the variance in Spearman's correlation with transfer performance when selecting different images from the same dataset.

4.2 Does DDS find best layer representation within a model to transfer from?

In previous works [42,8,35], a major focus was to select a model to initialize from. However, once the model is selected as an encoder for initialization, the new layers of decoder usually branch out from the last layer of the pre-trained encoder. Such an approach is based on the *a priori* assumption that, for any new task, the output from the last layer of the pre-trained encoder is the best representation for transfer learning. We argue that this is task-type dependent. For instance, it has been shown that earlier layers of DNNs trained on ImageNet object recognition learn low-level visual features while deeper layers learn high-level categorical features [24]. Therefore, one would expect for low-level visual task, the representation in earlier layers of DNN might be better for transfer learning. Based on this intuition, we investigate if layers at different depths of the network are better suited to transfer to different types of tasks. We compute DDS of a pre-trained model at different depths with Taskonomy models to assess representation types at different depths (Fig. 3). To validate it, we select 3 task types (2D, 3D, and semantic) from NYUv2 and Pascal VOC dataset and perform transfer learning by attaching the decoder to different encoder layers.

5 Experimental Setup

We implemented the DDS general framework in python[4], in which new parameters and functions ($\mathbf{Q}$, $\mathbf{D}$, f, g) can be incorporated in the future. Below, we first provide details of datasets and models used for comparing the DDS combinations for model selection. Then, we describe the datasets and models used for layer selection from a pre-trained encoder.

5.1 Dataset and models for Model Selection

Datasets. To compare different DDS combinations against the Taskonomy affinity and winrate matrix, we randomly select 200 images from the Taskonomy dataset. We use 200 images based on an analysis that shows that the correlation of DDS with transfer learning performance saturates at around 200 images (see Supplementary S3). To perform the bootstrap based comparison on a new semantic segmentation task on the Pascal VOC dataset, we randomly select 5000 images from Taskonomy, 5000 images from Pascal VOC, and all (1449) images from NYUv2 dataset.

Models. We use the selected 200 images to generate features from the last layer of the encoder of 17 models trained on 17 different tasks on the Taskonomy dataset. The Taskonomy models have an encoder/decoder architecture. The encoder architecture for all the tasks is a fully convolutional Resnet-50 without pooling to preserve high-resolution feature maps. The decoder architecture varies depending on the task. The models were trained on different tasks independently using the same input images but different labels corresponding to different tasks. For comparing two models, we use the features of the last layer of the encoder following [42,8]. The Pascal VOC semantic segmentation model that we use also has the same encoder architecture as Taskonomy models, and the decoder is based on the spatial pyramid pooling module, which is suitable for semantic segmentation tasks [5]. For comparison with the Pascal VOC model, we use the features of the last layer of the encoder of 17 Taskonomy models and the one Pascal VOC semantic segmentation model trained from scratch. We also report comparison with a small Pascal VOC model from [8] in Supplementary S4 to show that model selection can be performed even using small models.

5.2 Dataset and models for layer selection

Datasets. To validate whether the proposed layer selection using similarity measures reflects transferability, we perform training on different datasets and tasks by branching the decoders from different layers of the encoder. Specifically, we evaluate on 3 tasks (Edge Detection, Surface Normal Prediction and Semantics Segmentation) on Pascal VOC [10] dataset, and 3 tasks (Edge Detection, Depth Prediction and Semantic Segmentation) on NYUv2 [23] dataset. Following Zamir *et al.* [42], we use Canny Edge Detector [4] to generate groundtruth edge maps while other labels were downloaded from Maninis *et al.* [21].

[4] Code available at https://github.com/cvai-repo/duality-diagram-similarity

Q,**D** \ f	kernels			distances		
	linear	Laplacian	RBF	Pearson	euclidean	cosine
Identity	0.632	0.815	0. 800	0.823	0.688	0.742
Z-score	0.842	**0.860**	**0.841**	0.856	**0.850**	**0.864**
Batch norm	0.729	0.852	0.840	**0.857**	0.807	0.850
Instance norm	**0.849**	0.835	0.838	0.850	0.847	0.850
Layer norm	0.823	0.806	0.786	0.823	0.813	0.823
Group norm	0.829	0.813	0.790	0.829	0.814	0.829

Table 2. *Finding best DDS combination (***Q***,* **D***, f ,g).* We report the results of comparison with transferability for differents sets of **Q**, **D** and f. Top 3 scores are shown in green, blue, brown, respectively. Best **Q**, **D** for each f is shown in bold.

Models. We describe the models' encoder and decoder.
Encoder: We use a ResNet-50 [11] pre-trained on ImageNet [7] as our encoder, which has four blocks, each of the block consist of several convolution layers with skip connections, followed by a pooling layer. The branching locations that we explore are after each of the four pooling layers. We also consider Resnet-50 pre-trained on Places [43] using the same experimental set-up, and report the results in Supplementary S5.
Decoder: Following the success of DeepLabV3 [5] model, we use their decoder architecture in all our experiments. Since the output channels of the ResNet-50 encoder varies at different branching locations, we stack the output feature maps to keep the number of parameters in the downstream constant. More specifically, the encoder outputs 256, 512, 1024, 2048 channels for location 1, 2, 3 and 4 respectively, we stack the output of early branchings multiple times (8× for location 1, 4× for location 2 and 2× for location 3) to achieve a constant 2048 output channels to input to the decoder.

Training. ImageNet [7] pre-trained encoder is fine-tuned for the specific tasks, while the decoder is trained from scratch. In all the performed experiments, we use synchronized SGD with momentum of 0.9 and weight decay of 1e-4. The initial learning rate was set to 0.001 and updated with the "poly" learning rate policy [5]. The total number of epochs for the training was set to 60 and 200, for Pascal VOC [10] and NYUv2 [23], respectively as in Maninis *et al.* [21].

6 Results

In this section, we first report the comparison results of different similarity measures. After selecting the best similarity measure we apply it for identifying the representation type at different depth of the pre-trained encoder. Finally, we validate if the branching selection suggested using similarity measures gives the best transfer performance, by training models with different branching locations on NYUv2 and Pascal VOC datasets.

Method	Affinity	Winrate	Total time(s)
Taskonomy Winrate[42]	0.988	1	1.6×10^7
Taskonomy affinity[42]	1	0.988	1.6×10^7
saliency[35]	0.605	0.600	3.2×10^3
DeepLIFT[35]	0.681	0.682	3.3×10^3
ϵ-LRP[35]	0.682	0.682	5.6×10^3
RSA[8]	0.777	0.767	78.2
DDS ($f = cosine$)	0.862	0.864	84.14
DDS ($f = Laplacian$)	0.860	0.860	103.36

Table 3. *Correlation of DDS based affinity matrices with Taskonomy affinity and winrate matrix, averaged for 17 Taskonomy tasks, and comparison to state-of-the-art.* Top 2 scores are shown in green, and blue respectively. For this experiment, **Q** and **D** are selected to perform z-scoring, in all DDS tested frameworks.

6.1 Finding best DDS combination (Q, D, f ,g) for transferability

We perform a thorough analysis to investigate which combinations of (**Q**, **D**, f, and g) of the DDS lead to higher correlation with transferability rankings. We focus on how to assign weights to different feature dimensions using **Q**, **D** and distance functions f to compute the pairwise similarity between observations. In Table S2, we report results on the correlation with transferability rankings showing the effect of applying combination of **Q** and **D** instantiated as identity, z-score, batch/instance/group/layer normalization, and using different distance/kernel function as f. For g we use Pearson's correlation on unbiased centered dissimilarity matrices because it consistently showed a higher correlation with transfer learning performance (Supplementary Section S6). We observed a similar trend in results using Spearman's correlation for g (Supplementary Section S7).

In Table S2 we report the mean correlation of all the columns of the Taskonomy winrate matrix with the corresponding columns of a DDS based affinity matrix, which serve as the measure for computing how each of the similarity measures best predicts the transferability performance for each model. We first observe the results when **Q** and **D** are identity matrices. Laplacian and RBF kernels outperform linear kernel. For distance functions, Pearson outperforms euclidean and cosine. A possible reason for the better performance of Pearson's could be due to its invariance to translation and scale.

We next observe the effect of normalization using appropriate **Q** and **D**. We observe that the correlation with transferability rankings improves for all distance and kernel functions especially for low-performance distance and kernel functions. The gain in improvement is highest using z-scoring in most of the cases. A possible reason for overall performance improvement is that applying z-scoring reduces the bias in distance computation due to feature dimensions having high magnitude but low variance. Hence, for our next experiments, we choose z-scoring and select the top performing f: Laplacian and cosine.

Method	Taskonomy	Pascal VOC	NYUv2
DDS ($f = cosine$)	0.525 ±0.057	0.722 ±0.049	0.518 ±0.034
DDS ($f = Laplacian$)	0.5779 ±0.050	0.765 ±0.038	0.521 ±0.029

Table 4. *DDS correlation with transfer learning for Pascal VOC Semantic Segmentation.* Here each row represents a particular distance/kernel function as f, and each column represents a dataset. The values in the table are bootstrap mean correlation and standard deviation of a particular similarity measure computed using the image from a particular dataset. Top score is shown in green.

6.2 Comparison with state-of-the-art on Taskonomy

We first compare the DDS based affinity matrices on the Taskonomy transferability benchmark. To quantify in terms of mean correlation across all the tasks, we report mean correlation with Taskonomy affinity and winrate matrix in Table S3. In Table S3 (also Supplementary Section S8), we observe that all the proposed DDS based methods outperform the state-of-the-art methods [35,8] by a large margin. DDS ($f = cosine$) improves [8] and [35] by 10.9% (12.6%) and 26.3% (26.6%) on affinity (winrate), respectively. We report the correlation of different DDS based rankings with the rankings based on winrate and task affinities for 17 Taskonomy tasks in Supplementary S9 and find that proposed DDS based methods outperform state-of-the-art methods for almost all the tasks. We also report comparison using PR curve following [35] in Supplementary S10.

To compare the efficiency of different methods with respect to bruteforce approach, we report the computational budget required for different methods. A single forward pass of Taskonomy models on Tesla V100 GPU takes 0.022 seconds. Thus, for 17 tasks and 200 feedforward passes for each task, the total time for feedforward pass is 74.8 sec. Hence, the DDS based methods are several orders of magnitude faster than bruteforce approach, used in the Taskonomy approach [42], that requires several GPU hours to perform transfer learning on all the models. The number reported in Table S3 for Taskonomy was calculated by taking the fraction of the total transfer time (47,886 hours for 3000 transfers) for 17^2 transfers used for comparison in this work. Further, the time for obtaining DDS based rankings takes only a few seconds on CPU and is an order of magnitude faster than attribution maps based methods.

6.3 Evaluating robustness on a new task and dataset

In the evaluation benchmark that we proposed, which was used in the above reported experiments, we considered models that were trained using images from the Taskonomy dataset, and the images used to compute the DDS were also from the same dataset. To evaluate the robustness of DDS against a new task and images used to compute DDS, we consider a new task, namely Pascal VOC semantic segmentation, and use images from different datasets to compute DDS. To evaluate effect of selecting different images within the same dataset, we perform bootstrap to estimate the variance in correlation with transferability.

In Table 4, we report the bootstrap mean and standard deviation of correlation of different similarity measures with transfer learning performance on the Pascal VOC semantic segmentation task. We observe that the similarity measures show a high correlation (>0.70 for $f = cosine$ and >0.75 for $f = Laplacian$) when using images from Pascal VOC, but low correlation when using images from another dataset (Taskonomy and NYUv2). We also observed a similar trend in Taskonomy benchmark (Supplementary Section S11). Thus, the similarity measure is effective when using images from the same distribution as the training images for the model of the new task. We believe that using images from the same data distribution in DDS as the ones used to train the model on the new task for selecting the best initialization is important because the model in the new task is trained using data sampled from this distribution. Since high correlation of DDS ($f = Laplacian$) with transferability is obtained in all the investigated scenarios using the images from the dataset of the new task that we want to transfer to, we argue that this is the most suitable choice for estimating transferability as compared to other similarity measures and set-ups.

6.4 Finding representation type at different depth of a model

In the previous experiments, we demonstrated DDS ability to select models for transfer learning to a new task, given a set of source models. Here, we use DDS to interpret the representation type at different depths of the model, which would allow us to select which model layer to transfer from for a given type of task. For this purpose, we generate the features of the last layer of the encoder of 20 Taskonomy models to get the representation of each task type. We then compute the DDS ($f = Laplacian$) of Taskonomy features with each output block of the pre-trained ImageNet model. We use images from the same data distribution (Taskonomy) as used in the trained models to reveal the correlation with each layer and task type, as suggested in the previous experiment.

As shown in Fig. 4, we observe that the representation of block 1 is more similar to 2D models, block 3 is more similar to 3D and block 4 to semantic models. These results suggest that the representation of block 1 is better suited to transfer for 2D tasks, block 3 for 3D tasks and block 4 for semantic tasks. There is no clear preference for block 2. We observe a similar pattern with the pre-trained Places model (see Supplementary S5).

6.5 Does DDS predict best branching location from an encoder?

Here we report the results of transfer learning performances of 4 tasks: surface normal prediction on Pascal VOC [10], Depth Prediction on NYU Depth V2 [23], and edge detection and semantic segmentation on both datasets. These 4 tasks cover the 3 task clusters we observed in the previous section. The results are shown on Table 5. We report the qualitative comparison of different block outputs in Supplementary S5. We observe that branching out from block 3 gives the best performance on depth and surface normal, branching out from block 1 provides the best result on edge detection, and branching out from block 4 is

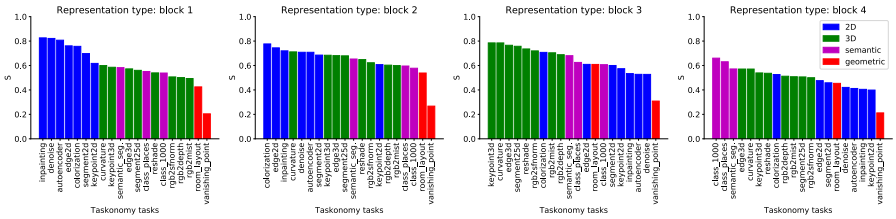

Fig. 4. *Block selection using DDS* on pre-trained encoder on Imagenet, and with DNNs trained on Taskonomy dataset on different tasks.

Block \ Task	Pascal VOC			NYUv2		
	Edge (MAE)	Normals (mDEG_DIFF)	Semantic (mIOU)	Edge (MAE)	Depth (log RMSE)	Semantic (mIOU)
1	**0.658**	18.09	0.257	**0.823**	0.322	0.124
2	0.686	15.59	0.392	0.857	0.290	0.165
3	0.918	**14.39**	0.627	1.297	**0.207**	0.219
4	0.900	15.11	**0.670**	1.283	0.208	**0.285**

Table 5. *Transfer learning performance of branching ImageNet pre-trained encoder on different tasks on Pascal VOC and NYUv2.* Results show that branching out from block 1, 3, 4 of the encoder have better performances on edge, normals (depth) and semantic tasks, respectively. This is consistent with the diagram similarity in Fig. 4.

best for semantic segmentation. The transfer learning results are consistent with the similarity results in Fig. 4, which suggests that DDS ($f = Laplacian$) is a robust method for encoder block selection for different tasks.

7 Conclusion

In this work, we investigated duality diagram similarity as a general framework to select model initialization for transfer learning. We found that after taking into account the weighing of feature dimension, DDS (for all distance functions) show a high correlation (>0.84) with transfer learning performance. We demonstrated on Taskonomy models that the DDS ($f = Laplacian$,$f = cosine$) shows 10% improvement in correlation with transfer learning performance over state-of-the-art methods. DDS ($f = Laplacian$) is highly efficient and robust to novel tasks to create a duality diagram. We further show the DDS ($f = Laplacian$) effectiveness in layer selection within a model to transfer from.

Acknowledgments. G.R. thanks the support of the Alfons and Gertrud Kassel Foundation. R.M.C. is supported by DFG grants (CI241/1-1, CI241/3-1) and the ERC Starting Grant (ERC-2018-StG 803370).

References

1. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. IEEE transactions on pattern analysis and machine intelligence **38**(9), 1790–1802 (2015)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Balaji, Y., Chellappa, R., Feizi, S.: Normalized wasserstein for mixture distributions with applications in adversarial learning and domain adaptation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 6499–6507 (2019)
4. Canny, J.: A computational approach to edge detection. In: Readings in computer vision, pp. 184–203. Elsevier (1987)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. ArXiv **abs/1706.05587** (2017)
6. De la Cruz, O., Holmes, S.: The duality diagram in data analysis: examples of modern applications. The annals of applied statistics **5**(4), 2266 (2011)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition pp. 248–255 (2009)
8. Dwivedi, K., Roig, G.: Representation similarity analysis for efficient task taxonomy & transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12387–12396 (2019)
9. Escoufier, Y.: The duality diagram: a means for better practical applications. In: Develoments in Numerical Ecology, pp. 139–156. Springer (1987)
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vision **88**(2), 303–338 (Jun 2010). https://doi.org/10.1007/s11263-009-0275-4, `http://dx.doi.org/10.1007/s11263-009-0275-4`
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
12. Holmes, S., et al.: Multivariate data analysis: the french way. In: Probability and statistics: Essays in honor of David A. Freedman, pp. 219–233. Institute of Mathematical Statistics (2008)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
14. Huh, M., Agrawal, P., Efros, A.A.: What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614 (2016)
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
16. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519–3529 (2019)
17. Kornblith, S., Shlens, J., Le, Q.V.: Do better imagenet models transfer better? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2661–2671 (2019)

18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**, 84–90 (2012)
19. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. 2016 Fourth International Conference on 3D Vision (3DV) pp. 239–248 (2016)
20. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. ArXiv **abs/1603.04779** (2016)
21. Maninis, K.K., Radosavovic, I., Kokkinos, I.: Attentive single-tasking of multiple tasks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1851–1860 (2019)
22. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3994–4003 (2016)
23. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
24. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017). https://doi.org/10.23915/distill.00007, https://distill.pub/2017/feature-visualization
25. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
26. Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In: Advances in Neural Information Processing Systems. pp. 6076–6085 (2017)
27. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Efficient parametrization of multi-domain deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8119–8127 (2018)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
29. Robert, P., Escoufier, Y.: A unifying tool for linear multivariate statistical methods: the rv-coefficient. Journal of the Royal Statistical Society: Series C (Applied Statistics) **25**(3), 257–265 (1976)
30. Rozantsev, A., Salzmann, M., Fua, P.: Beyond sharing weights for deep domain adaptation. IEEE transactions on pattern analysis and machine intelligence **41**(4), 801–814 (2018)
31. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)
32. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 806–813 (2014)
33. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713 (2016)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2015)
35. Song, J., Chen, Y., Wang, X., Shen, C., Song, M.: Deep model transferability from attribution maps. In: Advances in Neural Information Processing Systems (2019)
36. Székely, G.J., Rizzo, M.L., et al.: Partial distance correlation with methods for dissimilarities. The Annals of Statistics **42**(6), 2382–2412 (2014)

37. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7167–7176 (2017)
38. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
39. Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
40. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision. pp. 1395–1403 (2015)
41. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. pp. 3320–3328 (2014)
42. Zamir, A.R., Sax, A., Shen, W.: Taskonomy: Disentangling task transfer learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
43. Zhou, B., Khosla, A., Lapedriza, À., Torralba, A., Oliva, A.: Places: An image database for deep scene understanding. CoRR **abs/1610.02055** (2017)
44. Zhuo, J., Wang, S., Zhang, W., Huang, Q.: Deep unsupervised convolutional domain adaptation. In: MM '17 (2017)

Supplementary Material
Duality Diagram Similarity: a generic framework for initialization selection in task transfer learning

Kshitij Dwivedi[1,3][0000−0001−6442−7140], Jiahui Huang[2][0000−0002−0389−1721], Radoslaw Martin Cichy[3][0000−0003−4190−6071], and Gemma Roig[1][0000−0002−6439−8076]

[1] Department of Computer Science, Goethe University Frankfurt, Germany
kshitijdwivedi93@gmail.com, roig@cs.uni-frankfurt.de
[2] ISTD, Singapore University of Technology and Design, Singapore
jiahui_huang@sutd.edu.sg
[3] Department of Education and Psychology, Free University Berlin, Germany
rmcichy@zedat.fu-berlin.de

We provide the following items in the supplementary material, which complement the results reported in the main paper:

S1 RSA and CKA as special cases of duality diagram similarity (DDS)

The duality diagram of a matrix $\mathbf{X} \in \mathbb{R}^{n \times d_1}$ can be calculated by the product of $\mathbf{Q_X}$, $\mathbf{X}$ and $\mathbf{D}$, where $\mathbf{Q} \in \mathbb{R}^{d_1 \times d_1}$ is a matrix that quantifies dependencies between the individual feature dimensions, and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a matrix that assigns weights on the observations.

Let $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ be the duality diagrams obtained from two different models (layers), the duality diagram similarity (DDS) between those two can be calculated by first computing pairwise distance matrices, $\mathbf{M_X}$, $\mathbf{M_Y}$, using a distance

function, f, then use another function, g, to compare $\mathbf{M_X}$ and $\mathbf{M_Y}$ to obtain the final similarity score, $\mathbf{S}$. The formulation of DDS can be written as:

$$\mathbf{S} = g\Big(f\Big(\mathbf{DXQ_X}\Big), f\Big(\mathbf{DYQ_Y}\Big)\Big) \tag{1}$$

RSA as DDS. To compute RSA, one needs to obtain for each model (layer) the Representation Dissimilarity Matrices (RDMs), which is populated by computing a dissimilarity score $1-\rho$, where ρ is the Pearson's correlation coefficient between each pair of images (observations). Once the RDMs for each model (layer) is computed, then Spearman's correlation of the upper triangular part of the 2 RDMs is used to compute the final similarity score between the two RDMs. Here, one can observe the connection between RSA and DDS. In Equation 1, RDMs are the above-mentioned pairwise distance matrices, $\mathbf{M_X}$ and $\mathbf{M_Y}$, the distance function f used in RSA is the dissimilarity score $1-\rho$. If no normalization is used, matrix $\mathbf{D}$ and matrix $\mathbf{Q}$ are both identity matrices, $\mathbf{I}$ (ones in the diagonal and the rest of the elements in the matrix are zeros). In [8], they use a centering matrix $\mathbf{C}$ ($\mathbf{C} = \mathbf{I_n} - \frac{1}{n}\mathbf{1_n}$, where $\mathbf{1}$ is the $n \times n$ matrix of all ones) as $\mathbf{D}$ in the formulation of the duality diagram, and $\mathbf{Q}$ as the identity matrix. The final similarity score, g, used in RSA is the Spearman's correlation between lower/upper triangular part of the two RDMs. Finally, RSA as a special case of DDS can be written as:

$$\mathbf{S} = r_s^t\Big(1-\rho\Big(\mathbf{CXI}\Big), 1-\rho\Big(\mathbf{CYI}\Big)\Big) \tag{2}$$

where r_s^t denotes the Spearman's correlation of the upper traingular part of the two input matrices, the dissimilarity score $1-\rho$ is computed with the Pearson's correlation, $\mathbf{C}$ is the centering matrix for $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{I}$ is the identity matrix.

CKA as DDS. The formulation of CKA [17] can be written as:

$$\mathrm{CKA}\Big(\mathbf{X}, \mathbf{Y}\Big) = \mathrm{tr}\Big(\mathbf{KHLH}\Big) / \sqrt{\mathrm{tr}\Big(\mathbf{KHKH}\Big)\mathrm{tr}\Big(\mathbf{LHLH}\Big)}, \tag{3}$$

in which $\mathbf{K}$ and $\mathbf{L}$ are the output matrices after applying either the RBF or linear kernel (kernel function k) on data matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. Mathematically, $\mathbf{K}$ and $\mathbf{L}$ can be expressed as , $\mathbf{K_{ij}} = k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma_2||\mathbf{x}_i - \mathbf{x}_j||^2)$, $\mathbf{L_{ij}} = k(\mathbf{y}_i, \mathbf{y}_j) = \exp(-\gamma_2||\mathbf{y}_i - \mathbf{y}_j||^2)$ for RBF, and, $\mathbf{K_{ij}} = k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$, $\mathbf{L_{ij}} = k(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_i^T\mathbf{y}_j$ for the linear kernel. Here $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the i^{th} column of $\mathbf{X}$ and $\mathbf{Y}$ respectively, $\mathbf{H}$ is the centering matrix($\mathbf{H} = \mathbf{I_n} - \frac{1}{n}\mathbf{1_n}$, where $\mathbf{1}$ is the $n \times n$ matrix of all ones) , and T denotes the transpose. To obtain the equivalent formulation in DDS, in the following equations, we substitute $\mathbf{KH}$ and $\mathbf{LH}$ with $\hat{\mathbf{K}}$ and $\hat{\mathbf{L}}$ for simplification. Since $\hat{\mathbf{K}}_{\mathbf{ij}} = \hat{\mathbf{K}}_{\mathbf{ji}}$, we can get $\mathbf{K^T} = \mathbf{K}$, similarly, $\mathbf{L^T} = \mathbf{L}$, thus the above equation can be written as:

$$\mathrm{CKA}\Big(\mathbf{X}, \mathbf{Y}\Big) = \sum_{ij} \hat{\mathbf{K}}_{\mathbf{ij}}\hat{\mathbf{L}}_{\mathbf{ij}} / \sqrt{\sum_{\mathbf{ij}} \hat{\mathbf{K}}_{\mathbf{ij}}^{\mathbf{2}} \sum_{\mathbf{ij}} \hat{\mathbf{L}}_{\mathbf{ij}}^{\mathbf{2}}}, \tag{4}$$

which corresponds to the cosine similarity between $\hat{\mathbf{K}}$ and $\hat{\mathbf{L}}$. Since $\mathbf{K_{ij}} = k(x_i, x_j)$ and $\mathbf{L_{ij}} = k(y_i, y_j)$, we can treat them as pairwise distance matrix $\mathbf{M_X}$, $\mathbf{M_Y}$, respectively, in the formulation of DDS. In both cases, $\mathbf{Q}$ and $\mathbf{D}$ are both identity matrix here, and the distance function f is the linear or the RBF kernel. The final similarity function g used here is the cosine distance combined with the multiplication of the input matrices with the centering matrix $\mathbf{H}$. From above, we can derive CKA as a special case of DDS, and it can be written as:

$$\mathbf{S} = cos\Big(k\Big(\mathbf{IXI}\Big)\mathbf{H}, k\Big(\mathbf{IYI}\Big)\mathbf{H}\Big) \tag{5}$$

where *cos* is the cosine distance, k is either the linear or the RBF kernel, and $\mathbf{I}$ is the identity matrix.

S2 Different normalizations in DDS framework

In Table S1, we show how different normalizations used in deep learning and z-scoring can be reformulated in the DDS framework. Let $\mathbf{X} \in \mathbb{R}^{n \times c \times h \times w}$ be the output feature map of a convolutional layer with number of channels c, height h, width w for n input images. By swapping axes and reshaping the feature map $\mathbf{X}$, all the normalizations investigated in this work can be described in Duality Diagram setup. It is crucial to note that after reshaping the feature map, $\mathbf{D}$ and $\mathbf{Q}$ no longer represent weighing image and feature dimensions.

Norm	**D**	**X**	**Q**
Z-score	$\mathbf{I}_{n\times n} - \mathbf{1}_{n\times n}/n$	$\mathbf{X}_{n\times chw}$	$\mathbf{S}_{chw\times chw}$
Batch Norm	$\mathbf{I}_{nhw\times nhw} - \mathbf{1}_{nhw\times nhw}/nhw$	$\mathbf{X}_{nhw\times c}$	$\mathbf{S}_{c\times c}$
Group Norm	$\mathbf{I}_{\frac{c}{g}hw\times\frac{c}{g}hw} - \mathbf{1}_{\frac{c}{g}hw\times\frac{c}{g}hw}/\frac{c}{g}hw$	$\mathbf{X}_{\frac{c}{g}hw\times ng}$	$\mathbf{S}_{ng\times ng}$

Table S1. *Different normalizations in DDS framework.* Here $\mathbf{I}$ denotes an identity matrix, $\mathbf{1}$ denotes matrix filled with all 1's, $\mathbf{X}$ is the output feature map of a convolutional layer with number of channels c, height h, width w for n input images, g is group size for group normalization, and $\mathbf{S}$ is a diagonal matrix with diagonal values equal to standard deviation of $\mathbf{X}$ calculated across its rows. For each normalization, $\mathbf{X}$ is reshaped as indicated in the table. Layer and Instance normalization can be described by setting the group size g in Group norm to 1 and c respectively.

S3 DDS's dependence on number of images

To calculate similarity between 2 Deep Neural Networks (DNNs) using DDS, we need to perform feedforward pass through both DNNs on a selected set of images. Here we analyse the impact of the number of images selected to compute

the similarity measure. We varied the number of images from 10 to 500, in increments of 10, in a randomly selected set of Taskonomy images for Taskonomy benchmark and Pascal VOC images for Pascal VOC transfer learning benchmark, to calculate DDS. We plotted the correlation with transfer learning on Taskonomy tasks and on Pascal VOC semantic segmentation task in Figure S1a and Figure S1b respectively. We show the results for DDS with different f, namely Laplacian, RBF, linear, cosine, Euclidean and Pearson's correlation. We observe from the plots that correlation value with transfer learning saturates at around 200 images for all functions. For this reason, in all the experiments reported in the main paper we use 200 images in the selected sets.

S4 Model selection using a coarse task representation

Using task affinities as a method for source model selection, which is common also in all related works [8,35,42], requires a pre-trained model on the new task itself to measure affinities. In [8], it was proposed to train a small model on the new task, instead of a full large model, because it can be trained faster. The small models learn a coarse representation of the new task, and the task affinities to the source models can be compared faster. We use the small model from [8], and compare the correlation with transfer learning performance for DDS($f = Laplacian$) using small models and big models. We show the comparison in Figure S2, and we observe that correlation with transfer learning performance using small model is very close to the correlation using fully trained Taskonomy type model. Further, we observe that using DDS($f = Laplacian$) even with small model we outperform baseline RSA [8] that uses fully trained Taskonomy type models. Overall, the above results suggest that even with a coarse representation obtained by training a small model on new task can assist in model selection using the similarity measures proposed in this work.

S5 Results of Layer selection(ImageNet/Places pre-trained encoder)

In addition to the experiments conducted with ImageNet pre-trained encoder, reported in the main paper, here we also provide results for an encoder pretrained on Places365 [43]. The representation type of different blocks of Places pre-trained model, as shown in Figure S3, is similar to what we observed in Imagenet pre-trained model, reported in the main paper. From Table S2 and Table S3, we observe that our similarity measure successfully predicted best branching location for 5 out of 6 cases. Only exception is NYUv2 depth estimation task, where the transfer learning performance of block 3, selected by the method, is slightly lower than the best branching location (block 4). Overall from the above results combined with ImageNet results from the main text, we find that the proposed method reliably selects high performing branching locations to transfer to new tasks.

Block \ Task	Edge (MAE)	Normals (mDEG_DIFF)	Semantic (mIOU)
1	**0.680**	17.89	0.244
2	0.777	15.62	0.368
3	1.012	**14.35**	0.532
4	1.002	14.73	**0.616**

Table S2. *Transfer learning performance of branching Places pre-trained encoder on 3 tasks on Pascal VOC dataset.* The results indicate that branching out from block 1, 3, 4 of the encoder have better performances on edge, normals and semantic tasks, respectively.

Block \ Task	Edge (MAE)	Depth (log RMSE)	Semantic (mIOU)
1	**1.027**	0.320	0.125
2	1.188	0.286	0.167
3	1.183	0.223	0.216
4	1.120	**0.201**	**0.291**

Table S3. *Transfer learning performance of branching Places pre-trained encoder on 3 tasks on NYUv2 dataset.* The results are mostly consistent with branching location prediction based on DDS.

We show qualitative results on Pascal VOC[10] and NYUv2[23] datasets in Figure S4 and Figure S5. Here we illustrate branching results of 3 tasks: Edge Detection, Surface Normal (Depth) Prediction and Semantic Segmentation. For each task, ImageNet pre-trained encoder results are shown on the upper row, and Places 365 pre-trained encoder results are shown on the lower row. We observed some visual quality degradation in the results of non-optimal branching locations predicted by our similarity measures: Edge contours become blurry as the branching location goes deeper; semantic segmentation maps become closer to the ground truth at deeper layers.

S6 Effect of unbiased centering

We report in Table S4 the effect of applying unbiased centering (eq. 3.1 in [36]) to $\mathbf{M_X}$ and $\mathbf{M_Y}$ on the correlation with transferability. We observe that for all cases unbiased centering improves the correlation with transfer learning, and hence, in all the reported results in the main paper we used unbiased centering.

S7 Results with Spearman's correlation as g

In the main text, we reported the results with g as Pearson's correlation between unbiased centered (dis)similarity matrices $\mathbf{M_X}$ and $\mathbf{M_Y}$. Here, in Table S5, we

Centering ($\mathbf{M_X}$, $\mathbf{M_Y}$) \ f	kernels			distances		
	linear	Laplacian	RBF	Pearson	euclidean	cosine
No centering	0.818	0.691	0.690	0.776	0.613	0.792
Unbiased centering	0.842	0.860	0.841	0.856	0.850	0.864

Table S4. *Effect of unbiased centering.* We report the results of comparison with transferability on Taskonomy transfer learning for with and without unbiased centering on pairwise (dis)similarity matrices $\mathbf{M_X}$ and $\mathbf{M_Y}$

$\mathbf{Q}$ \ f	kernels			distances		
	linear	Laplacian	RBF	Pearson	euclidean	cosine
Identity	0.778	0.828	0.803	0.816	0.798	0.803
Z-score	0.858	0.864	0.846	0.844	0.862	0.860

Table S5. *Spearman's as g.* We report the results of comparison with transferability on Taskonomy transfer learning benchmark for with and without z-scoring when using Spearman's as g.

report results when g is the Spearman's correlation between upper/lower triangular part of unbiased centered (dis)similarity matrices $\mathbf{M_X}$ and $\mathbf{M_Y}$, as in [8], on Taskonomy transfer learning benchmark. We observe that the results show similar trend with Spearman's correlation (improvement on applying z-scoring on $\mathbf{X}$,$\mathbf{Y}$) as using Pearson's correlation as g, shown in main text Table 2 .

S8 DDS results for all distance/kernels as f

In Table 3 and Table 4 of main paper we reported the best f selected using the results in Table 2. Here we report the complete results for Table 3 and Table 4 with all investigated functions as f. Due to efficiency of our method it was possible to perform multiple bootstrap to calculate standard deviation in correlation with transfer learning. In the tables below (Table S6 and Table S7), we report bootstrap mean and standard deviation of correlation with transfer learning for Taskonomy tasks and Pascal VOC semantic segmentation task. We observe that DDS($f = Laplacian$) is the most robust (in Top 1,2) measure across both Taskonomy benchmark and Pascal VOC semantic segmentation tranfer learning.

S9 DDS similarity measure comparison for 17 Taskonomy tasks

In the main paper, we reported the mean correlation of similarity measures with transfer learning across 17 Taskonomy tasks. In Figure S6, we provide the detailed results on all tasks. We find that almost on all the tasks our proposed similarity measures outperform the state-of-the-art method [35,8].

Method	Affinity	Winrate
DDS($f = pearson$)	0.853 ±0.090	0.851 ±0.090
DDS($f = euclidean$)	0.852 ±0.076	0.855 ±0.079
DDS ($f = cosine$)	0.862 ±0.076	0.863 ±0.078
DDS($f = linear$)	0.837 ±0.084	0.841 ±0.088
DDS ($f = Laplacian$)	0.862 ±0.072	0.861 ±0.072
DDS($f = rbf$)	0.854 ±0.086	0.854 ±0.088

Table S6. *Correlation (Bootstrap mean ±standarddev) of DDS based affinity matrices with Taskonomy affinity and winrate matrix, averaged for 17 Taskonomy tasks.* Top 2 scores are shown in green, and blue respectively. For this experiment, **Q** is set to z-scoring and **D** to the identity matrix, in all DDS tested frameworks.

Method	Taskonomy	Pascal VOC	NYUv2
DDS ($f = Pearson$)	0.534 ±0.063	0.726 ±0.049	0.505 ±0.033
DDS ($f = euclidean$)	0.534 ±0.055	0.746 ±0.051	0.518 ±0.030
DDS ($f = cosine$)	0.525 ±0.057	0.722 ±0.049	0.518 ±0.034
DDS ($f = linear$)	0.496 ±0.063	0.718 ±0.062	0.515 ±0.033
DDS ($f = Laplacian$)	0.577 ±0.050	0.765 ±0.038	0.521 ±0.029
DDS ($f = RBF$)	0.591 ±0.053	0.753 ±0.051	0.534 ±0.030

Table S7. *DDS correlation with transfer learning for Pascal VOC Semantic Segmentation.* Here each row represents DDS with a particular distance/kernel function as f, and each column represents the dataset from which the images were selected to get similarity scores. The values in the table are bootstrap mean correlation and standard deviation of a particular similarity measure computed using the image from a particular dataset. Top score is shown in green.

S10 Precision and Recall curve for DDS

In the main text, we used correlation of similarity measure based source model rankings with transfer learning performance based rankings as our evaluation criteria. Song *et al.* [35] used precision and recall of selecting top-5 source tasks as the evaluation criteria. We use the evaluation code provided by [35], and we plot precision and recall curve for one of our most robust proposed method, DDS($f = Laplacian$), against state-of-the-art methods [35,8], in Figure S7. In Figure S7, we plot results using 200 Taskonomy images for all the similarity measures that we compared. We further add the results of the methods from Song *et al.* [35] using 1000 images from indoor dataset used in Song *et al.* [35] that showed best performance in their paper. We observe from Precision and Recall plots in Figure S7 that DDS($f = Laplacian$) outperforms the state-of-the-art methods.

S11 DDS dependences on image dataset choice

In this section, we investigate the effect of image dataset used to calculate Duality Diagrams. We report the results of DDS correlation with Taskonomy winrate in Table Table S8when images from Taskonomy, Pascal VOC, and NYUv2 were used to calculate Duality Diagrams. We observe a slight drop in DDS's correla-

Method	Taskonomy	Pascal VOC	NYUv2
DDS ($f = cosine$)	0.864	0.818	0.822
DDS ($f = Laplacian$)	0.860	0.811	0.818

Table S8. *DDS correlation with transfer learning on Taskonomy Tasks.* Here each row represents DDS with a particular distance/kernel function as f, and each column represents the dataset from which the images were selected to obtains similarity scores.

tion with Taskonomy winrate matrix when using images from Pascal VOC and NYUv2 dataset.

These results are consistent with [35] where they show that their method is robust to choice of images used to compute similarity between neural networks. In the aforementioned results, both source and target tasks were trained using the same training dataset, i.e. Taskonomy, and we believe that is the reason we, as well as [35], do not observe much difference.

However, when we compare transferability on Pascal VOC, source models are trained on Taskonomy dataset and target task is on Pascal VOC, which has significantly different statistics than Taskonomy. In this more challenging setting, we observe the impact of using images from different datasets, as reported in Section 6.3 in the main text.

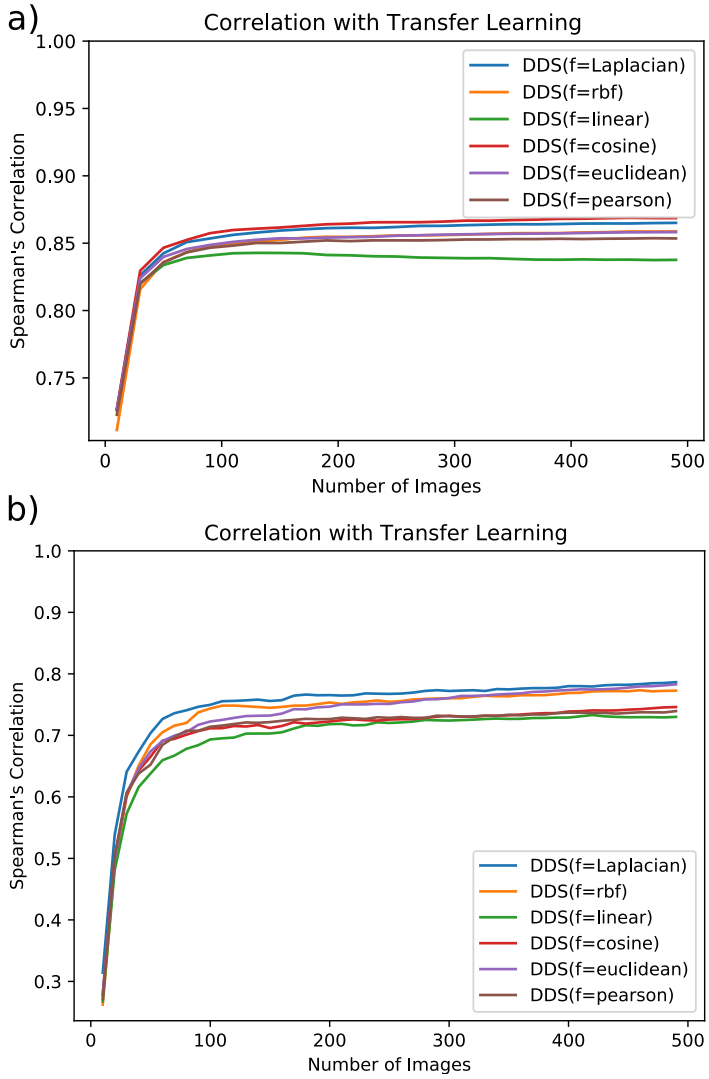

Fig. S1. *Spearman's correlation of DDS and transfer learning performance* on a) Taskonomy tasks, and b) Pascal VOC semantic segmentation task. The above plots shows how Spearman's correlation of DDS with transfer learning varies with the number of images used to compute similarity using DDS with different distance/kernel functions as f. The images are randomly sampled from the Pascal VOC dataset.

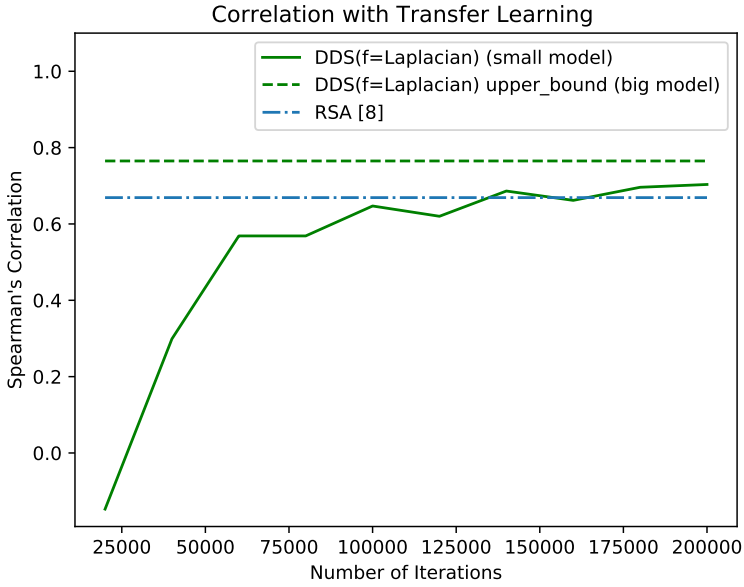

Fig. S2. *Correlation of DDS(f = Laplacian) based rankings obtained using a small model trained on Pascal VOC semantic segmentation task with the transfer learning performance.* We show how correlation varies with different stages of training. We further compare the results with the upper bound obtained by using the large trained Taskonomy type model on Pascal VOC semantic segmentation as the task representation and also a baseline RSA using the large model.

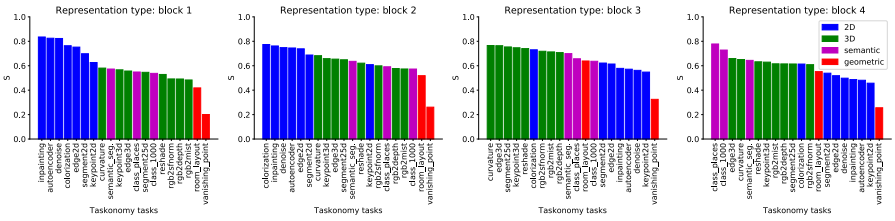

Fig. S3. *Block selection using DDS* on pre-trained encoder on Places, and with DNNs trained on Taskonomy dataset on different tasks.

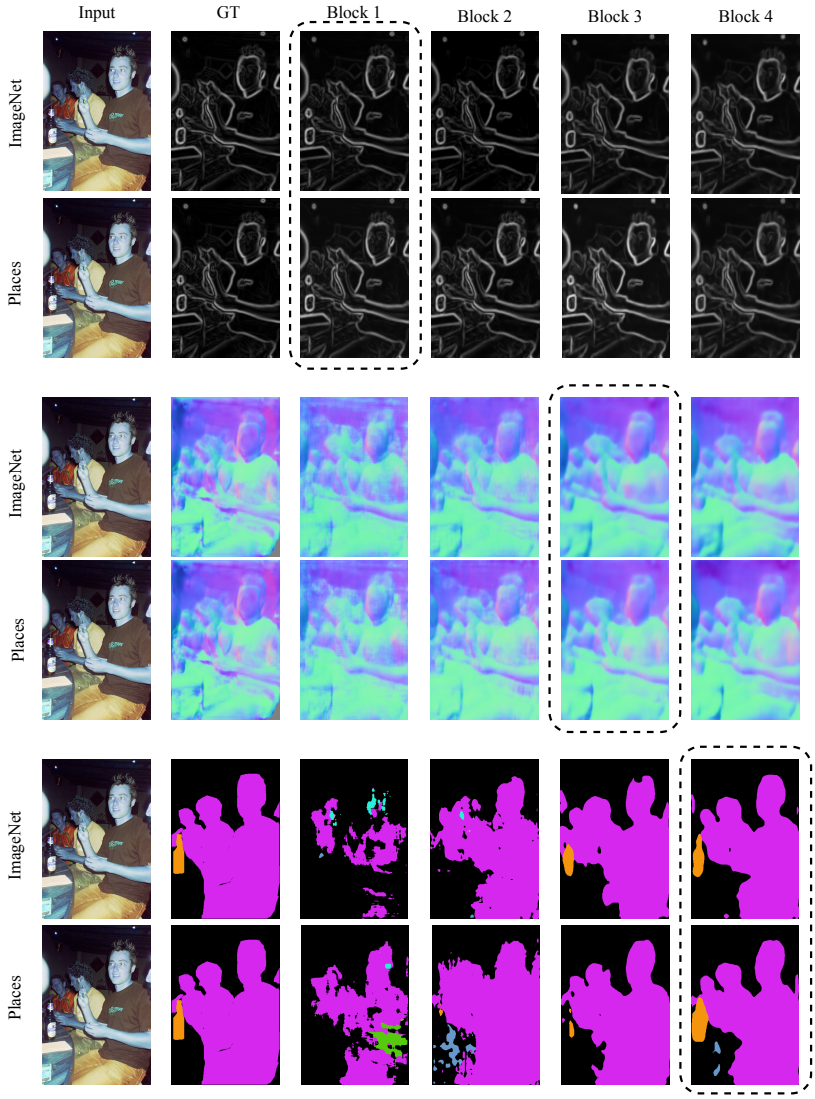

Fig. S4. ***Qualitative Results on Pascal VOC.*** *Branching results of all locations on three tasks are shown: Edge Detection, Surface Normal Prediction and Semantic Segmentation. For each task, ImageNet pre-trained encoder are shown on the upper row, while Places 365 pre-trained encoder are shown on the lower row. Best results are circled with dotted lines.*

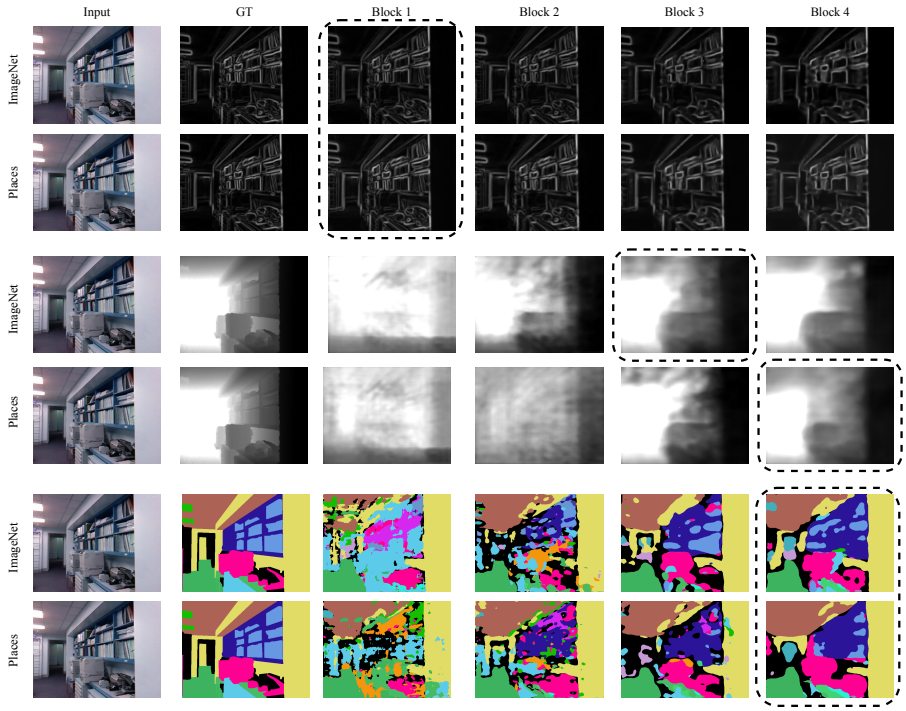

Fig. S5. ***Qualitative Results on NYUv2.*** *Branching results of all locations on three tasks are shown: Edge Detection, Depth Prediction and Semantic Segmentation. For each task, ImageNet pre-trained encoder are shown on the upper row, while Places 365 pre-trained encoder are shown on the lower row. Best results are circled with dotted lines.*

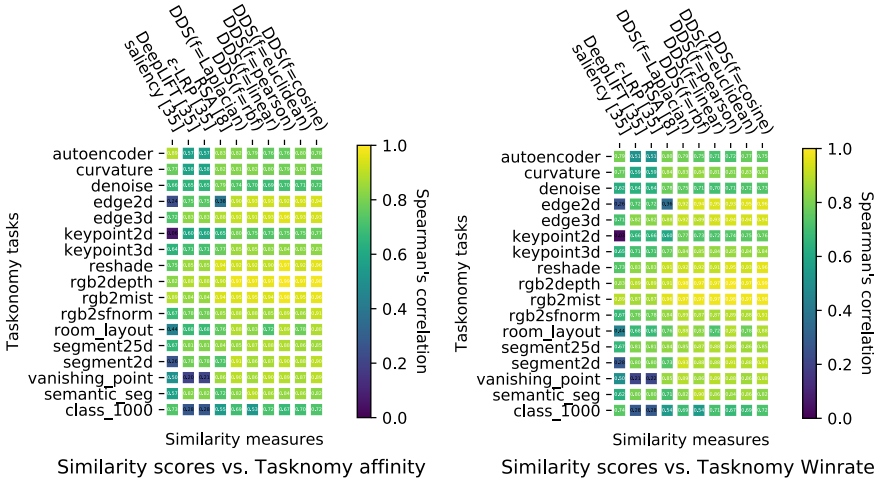

Fig. S6. *Similarity measures' comparison on Taskonomy Tasks.* Spearman's correlation of different similarity measure based rankings with transfer learning performance based rankings from Taskonomy affinity matrix (left), and Taskonomy winrate matrix (right) for 17 Taskonomy tasks as target. We show the results for 17 Taskonomy tasks (rows) for different similarity measures (columns). More yellow indicates higher the correlation, hence, is better.

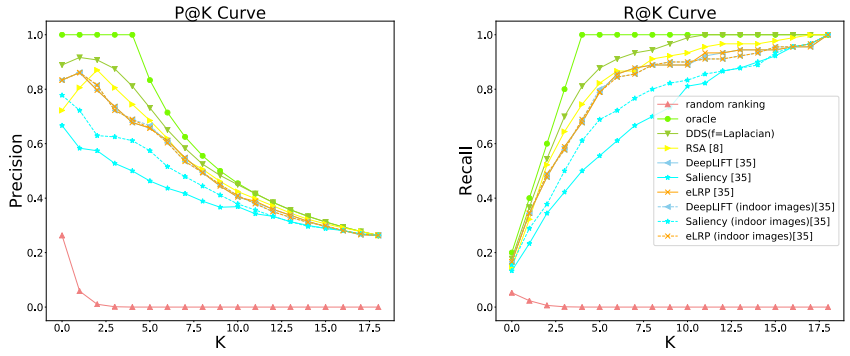

Fig. S7. *Precision and Recall Curve for comparing similarity measures.* The x-axis in all the plots above refers to the number of source tasks used for calculating precision and recall value.

3 Associating artificial neurons to concepts

What do navigation agents learn about their environment?

Kshitij Dwivedi, Gemma Roig
Goethe University Frankfurt
dwivedi@em.uni-frankfurt.de, roig@cs.uni-frankfurt.de

Aniruddha Kembhavi, Roozbeh Mottaghi
PRIOR @ Allen Institute for AI
anik@allenai.org, roozbehm@allenai.org

Abstract

Today's state of the art visual navigation agents typically consist of large deep learning models trained end to end. Such models offer little to no interpretability about the learned skills or the actions of the agent taken in response to its environment. While past works have explored interpreting deep learning models, little attention has been devoted to interpreting embodied AI systems, which often involve reasoning about the structure of the environment, target characteristics and the outcome of one's actions. In this paper, we introduce the Interpretability System for Embodied agEnts (iSEE) for Point Goal and Object Goal navigation agents. We use iSEE to probe the dynamic representations produced by these agents for the presence of information about the agent as well as the environment. We demonstrate interesting insights about navigation agents using iSEE, including the ability to encode reachable locations (to avoid obstacles), visibility of the target, progress from the initial spawn location as well as the dramatic effect on the behaviors of agents when we mask out critical individual neurons.

1. Introduction

The research area of Embodied AI – teaching embodied agents to perceive, communicate, reason and act in their environment – continues to receive a lot of interest from the computer vision, natural language processing and robotics communities. A growing body of work has resulted in the emergence of several powerful and visually rich simulators including AI2-THOR [20], Habitat [25] and iGibson [37]; works that require agents to navigate [2], reason [5], collaborate [18], manipulate [12] and follow instructions [3].

While fast progress is being made across a variety of tasks and benchmarks, most solutions being employed are black box neural networks trained to either imitate a sequence of human/oracle actions or trained via reinforcement learning with a careful selection of positive and negative rewards. These models offer little to no interpretability out-of-the-box about the concepts and skills learned by the model or about the actions taken by the model in response to a task or observation. Developing interpretable systems is particularly important in embodied AI since we expect these systems to eventually be deployed onto robots that will navigate the real physical world and interact with people in it.

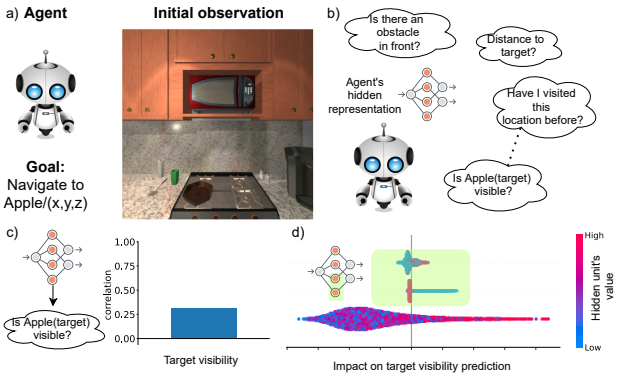

Figure 1. **The iSEE framework.** (a) An agent learns to perform the OBJECTNAV or POINTNAV tasks. (b) We wish to explore what information is encoded in the hidden representations of the agent. (c) To achieve this, we evaluate how well the agent's hidden representation can predict human interpretable concepts e.g. target visibility in ObjectNav. (d) Then we apply an explainablity method SHAP [23] to identify the top-k relevant units.

In the image classification literature, a number of interpretability methods have been developed over the past few years [7, 14, 28, 49]. These methods rely on probing model activations via various inputs or generating synthetic inputs that lead to a spike in an activation. While such methods are useful in probing Embodied AI models, they do not take into account the rich metadata (such as perfect segmentation, depth maps, precise object localization, etc.) available in synthetic environments commonly used to train these

models. Simulated worlds provide us a unique opportunity to expand interpretability research to embodied agents and develop new methods that take advantages of rich metadata.

We propose a framework to interpret the hidden representations of embodied agents trained in simulated worlds. We apply our framework to two navigation tasks (Figure 1a): Object Navigation (OBJECTNAV) [6], the task of navigating to a target object and Point Goal Navigation (POINTNAV) [2], the task of navigating to a specified relative co-ordinate, within the AI2THOR environment; but our methods are general and can be easily applied to more tasks and other environments. We train agents to perform these tasks and then probe their hidden representations to evaluate if they encode aspects of their task, progress and surroundings (Figure 1b and 1c). We then apply the model interpretation method SHAP [23] to identify which hidden units are most relevant for predicting these concepts (Figure 1d). Our framework allows us to gather evidence towards answering two fundamental questions about a trained model: (1) Has the model learned a particular concept ? (2) Which units within a recurrent layer encode this concept ? Using this framework, we were able to find several interesting insights about OBJECTNAV and POINTNAV agents.

The key contributions of this work are:

- A new interpretability framework specialized for navigation agents with no linearity assumptions between concepts and hidden units.
- New insights about what navigation agents encode and in which units:
  - sparse target representation in OBJECTNAV (50/512 units) and POINTNAV (5/512 units);
  - learning of concepts such as reachable locations and visit history by OBJECTNAV agents; encoding of progress towards target and less reliance on visual information by POINTNAV agents.
- Ablation experiments showing no impact on model performance after removal of 10% units suggesting redundancy in the representation.

2. Related Work

We explore representations stored within an agent's hidden units by predicting a human interpretable piece of information about the agent and its environment. Our work is related to two directions in interpretability research: (1) Interpretability of individual hidden units and (2) Explaining model's predictions.

Interpretability of hidden units. A common approach to investigate what a hidden unit encodes is to find the input image also referred to as "preferred image" that leads to a maximal activation of the unit of interest. The preferred image can be from within the examples in a dataset [49, 50] or obtained using gradient descent by optimizing over the input [13, 16, 28, 29, 39, 42]. One disadvantage of the methods using preferred images is that it is difficult to quantify the association of a unit with a concept. To address this issue, NetDissect [7, 51] uses overlap of a unit's spatial activation with groundtruth segmentation maps of a human interpretable concept as a measure to quantify a unit's association with a concept. The idea was further extended in Net2vec [14] to investigate whether a single unit or a group of units encode a concept. However, these approaches require groundtruth pixel-level annotation for every concept of interest and therefore for new concepts, new annotations are required. On the other hand, simulation environments [20, 25, 37] have annotations readily available as a part of the metadata. However, given the vast amount of metadata beyond simply object information, there is a need to develop new methods for these environments to interpret embodied agents. Recent embodied AI works [43, 48] have started focusing in interpretability by linear decoding of concepts from hidden units [43] and finding computational structure of the agent's recurrent units using fixed point analysis [48]. Patel et al. [31] explored interpretation of emergent communication in collaborative embodied agents. However these works do not focus on identifying which hidden units encode a given concept which is one of the main contributions of the present work.

Explaining model predictions. Saliency methods [4, 30, 33, 35, 40] use gradients to find which pixels of an image are relevant for model's prediction. Additive feature attribution methods [24, 34, 38] investigate the effect of adding an input feature in model prediction. A disadvantage of these methods is that they focus on explaining the model predictions on raw pixel level. To explain the model prediction using human-interpretable concepts, TCAV [19] and subsequent works [15, 17, 21] were proposed that use concept vectors instead of raw pixels to explain model prediction. To find concept vectors additional human annotations are required. In the embodied environments [20, 25, 37], we have the advantage of already annotated human interpretable concepts.

The above two directions of research have been considered as independent directions of interpretability research – one focusing on interpreting what the hidden units learn and the other on interpreting the decisions made by the model. In this work, we observe the potential of linking two approaches to interpret what the hidden units learn by using human interpretable concepts. Specifically, we train an interpretable model (Gradient boosted Tree) to predict human intepretable concepts from the hidden units of the model and then apply a global model explainability method SHAP [23] to explain which units are relevant for which concept prediction. In this work, we use SHAP because (a) it provides a unique solution with three desirable properties: local accuracy, missingness and consistency [24], (b) it unifies several model agnostic [34, 38] and tree based explanation methods [1], and (c) it provides explanation on

both local (single example) and global (dataset) levels.

Embodied tasks. Several approaches have been proposed [8–11, 22, 26, 27, 32, 37, 41, 45–47, 52] to tackle the navigation problem, which is a core task in Embodied AI. In this paper, we analyze standard base models for two popular navigation tasks, PointNav [2] and ObjectNav [6].

3. Interpretability Framework

We introduce the **I**nterpretability **S**ystem for **E**mbodied ag**E**nts (iSEE). iSEE probes agents at their understanding of the task given to them, their progress at this task and the environment they act in. This probing is done via training simple machine learning models that input network activations and output the desired information. Simulated environments provide us with a gamut of metadata about the agent, task and surroundings, allowing us to train a series of models for probing this information. iSEE also helps identifying specific neural units that store this information. This is done via computing the SHapley Additive exPlanations (SHAP) [36] values for individual neural units. Finally we study the effect of switching off individual neural units on the downstream tasks that the agents are trained for.

We study embodied agents trained for POINTNAV [2] (navigation towards a specific coordinate in a room) and OBJECTNAV [6] (navigation towards a specific object). Our agents encode their visual observations via a convolutional neural network and encode their target/goal via an embedding layer. The outputs of the visual and goal encoders are fed into a gated recurrent unit (GRU) to add memory. The hidden units of the GRU are then linearly transformed into the policy (distribution over actions) (Figure 2a). There are more complex, customized models for each of these tasks that achieve higher performance. However, we utilize these simple, generic models that can be applied to various tasks and make the comparisons across tasks more fair. In this work, we use iSEE to probe the hidden units in the GRU and use gradient boosted trees (GBT) as the ML model to determine the presence of relevant information within these hidden units (Figure 2b). We focus here specifically on GRU units since (a) we are interested in analyzing dynamic visual representations (GRU units) as opposed to static visual representations (CNN visual encoder) and (b) some of our models use a frozen visual encoder and only optimize the parameters within the GRU.

We now describe the metadata extracted from the simulator, probing for this metadata via building GBTs and using SHAP to identify individual hidden units that store the relevant information.

3.1. Metadata

We probe agents at their understanding of the target, their position in the scene, the reachability of objects in their surroundings and their memory of visited locations as they navigate their world. This information is easily extracted by us from the metadata provided by the simulator.

Target Information: Agents trained for the OBJECTNAV and POINTNAV tasks must navigate to the location of a specified object or a point, respectively. In either case, one might expect an agent to be able to estimate its positioning with respect to the goal. Therefore, at a given timestep t, we extract metadata containing the distance (R_t) and orientation (θ_t) of the agent from the target (Figure 2c). In OBJECTNAV, an agent is successful if the object lies within 1m of the agent and is visible; thus we additionally extract target visibility ($visible_t$). Since an object may be visible in the frame but not within the specified distance to determine success, we also extract the percent of pixels covered by the target object using segmentation masks provided by AI2-THOR ($Area_t$).

Agent's information: Memory of how far and in what direction one has travelled can be relevant to avoiding revisiting locations in the scene. Therefore, we extract the agent's distance (R_a) and orientation (θ_a) with respect to its starting location (Figure 2c).

Reachability: For an agent to successfully navigate in a scene it should be able to detect obstacles and its path around them. Thus, we extract metadata to detect whether a particular location with respect to the agent's current location is reachable or not. Given an agent's location, we first extract all reachable gridpoints in the scene. Then, with the agent's location as the center we consider three concentric circles with radii=2, 4,and 6 times the grid size and locate points on these circle that are at angles from 0 to 360 in the steps of the agents rotation angle (=30 degrees). For each of these points R_r, θ_{angle}, where r is the radius and $angle$ is the orientation of the grid point with respect to agent in degrees, we check whether the closest reachable gridpoint is within $gridSize/\sqrt{2}$ or not. Figure 2d illustrates such reachable gridpoints in the scene.

Visited History: The metadata extracted above captures a global summary of the agent's movements. We also extract its local visit history. This is done by checking if a location ($\texttt{visited}_l$), rotation ($\texttt{visited}_{lr}$) and camera horizon ($\texttt{visited}_{lrh}$) has been visited by the agent or not.

3.2. Metadata extraction

As the agent traverses around in a scene, we extract the GRU activations of the agent along with the agent and scene metadata described above. This is done within the training and validation scenes. The latest model architectures and training algorithms for POINTNAV and OBJECTNAV lead to very capable agents that (a) exhibit little variability in their trajectories (b) do not collide often (c) make few mistakes such as revisiting locations. Such trajectories are less useful to probe agents, since the events of interest occur sparsely. Hence we use human trajectories (trajectories

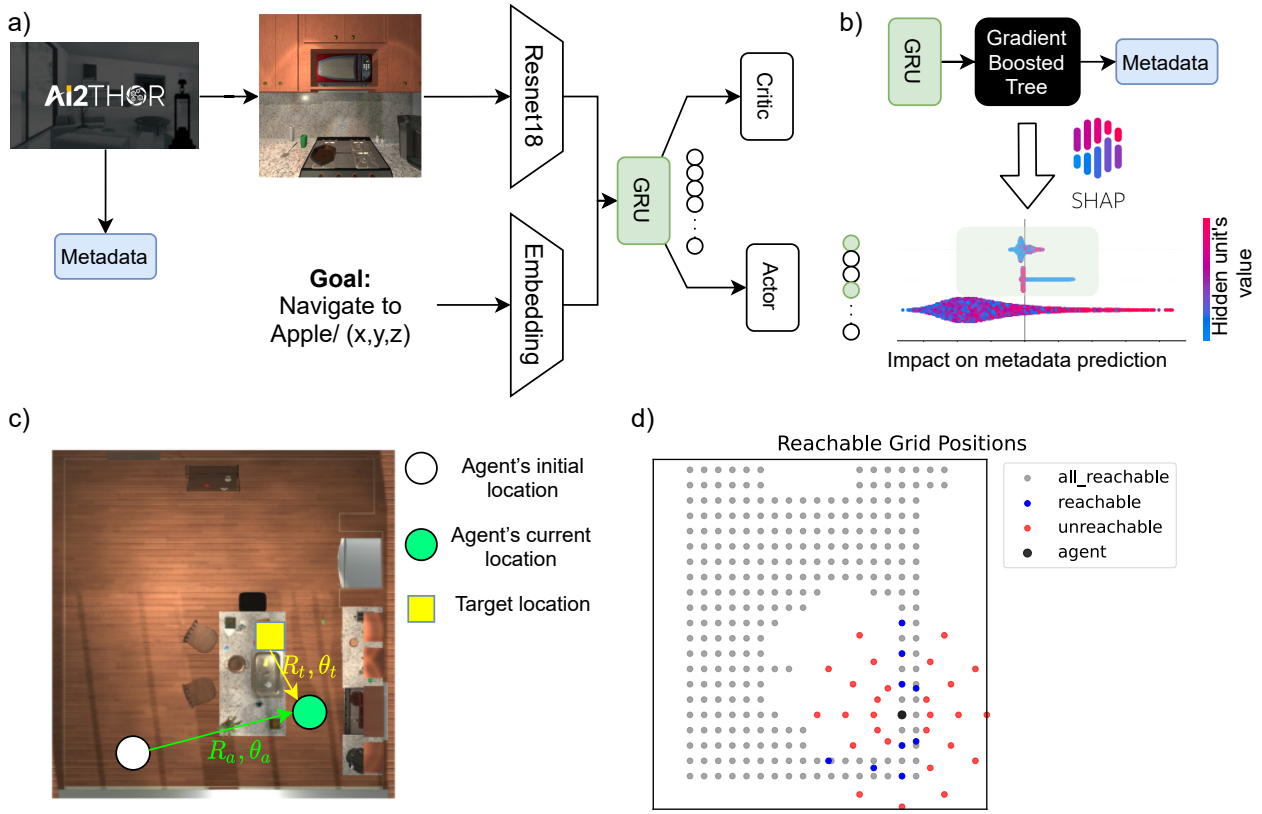

Figure 2. **iSEE:** a) At a given timestep, AI2THOR generates an observation that is fed as input to the agent along with a goal embedding. For that time step, we also extract relevant event metadata from AI2THOR which is unseen by the agent. b) After sampling rollouts from multiple training and validation episodes, we train a gradient boosted tree to predict metadata from the agent's hidden representation (GRU units). We then apply SHAP, an explainability method that identifies the top-k most relevant units for predicting a given metadata type. c) At a given timestep, we extract agent's orientation with respect to its initial spawn location (R_a, θ_a) and target location (R_t, θ_t). d) We extract reachable positions at distance 2,4,6 times the grid size and different angles with step size of 30 degrees to identify whether these locations can be reached by the agent or not.

specified by humans navigating around) that encourage exploration and have intentional collisions and mistakes. Using a pre-defined set of human trajectories also enables us to fairly compare findings across agents.

3.3. Metadata prediction

We train GBTs to predict specific metadata concepts using the GRU's hidden units as inputs. GBTs are trained using episodes within the training scenes and evaluated using correlation between the predicted metadata and groundtruth metadata on the validation episodes. For a given model, we trained one GBT of $depth = 10$ for each concept using xgboost library. For binary variables (such as target visibility) we use the logistic loss function and for continuous variables (such as distance from target/agent's initial position) we use the mean squared error loss function. Total training and evaluation time of GBT was 8 seconds on a single NVIDIA RTX 2070 GPU. We use GBTs because: (1) they are more interpretable in comparison to many other ML models when the mapping from inputs to outputs is not linear; (2) allow exact computation of SHAP values as compared to other models where SHAP values can only be approximated [23].

3.4. Identifying explainable units using SHAP

Given a set of hidden units, SHAP computes the importance of each individual unit by quantifying its contribution towards predicting a concept. SHAP values are based on a game theory concept called Shapley values [36]. We first train a GBT to predict a concept using all hidden units. We then use a subset of hidden units and mask other units to predict a concept using pretrained GBT. Then we add in a new hidden unit and compute the change in the model's prediction capability. This difference quantifies the contribution of a hidden unit with regards to the chosen subset. By averaging this contribution over all possible subsets of hidden units, we get the Shapley value of the unit of interest. For instance, we use this method to compute the contribution of a specific GRU hidden unit towards predicting the visibility of the specified target. Note that the obtained

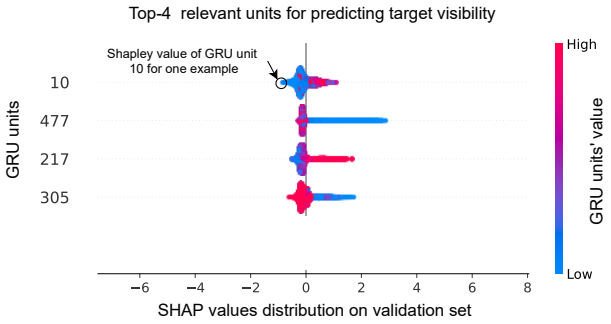

Figure 3. **Schematic to read a SHAP plot:** The plot shows the top-4 relevant GRU units to predict the target visibility. Each row shows the distribution of SHAP values of a given GRU unit for all the examples in the validation set with each dot in the row corresponding to an individual data point. The color of the dot indicates whether the GRU unit's output was low or high for that data point.

Shapley value indicates the impact of the hidden unit on the model's outcome for a single example. To quantify the global impact of hidden unit on model's outcome we calculate the mean of absolute SHAP values over all examples in the validation set (for more details please see Appendix A).

Figure 3 is a SHAP beeswarm plot to visualize the global contribution of the top-k relevant GRU units. We use this plot to explain how one can interpret SHAP plots. This plot visualizes the contribution of the top 4 relevant units to predict target visibility. Each row corresponds to a given GRU unit, and each dot in the row corresponds to the GRU unit's Shapley value for a given example. Each row displays the distribution of SHAP values on all the samples of the validation set. The location of a dot on the x axis shows whether the impact of the GRU unit on model's prediction (i.e. Shapley value) is positive or negative. The GRU unit's value for a sample is visualized using the colorbar on right. As an example, for the circled dot in Figure 3, the Shapley value of GRU unit 10 is negative and the color of the dot indicates that GRU unit 10's value is also low. For the examples on the right side of x-axis the shapley values are positive and the GRU unit's values are also higher. This means that GRU unit 10 is positively correlated with target visibility. Using a similar logic GRU unit 477 seems to be negatively correlated with target visibility. In a nutshell, the SHAP plot shows the global contribution of a GRU unit in prediction of a concept (rows sorted by contribution), displays the distribution over the validation examples (points in each row) and indicates whether a unit is positively or negatively correlated with the concept (colors of the points in accordance with the x-axis values).

4. Experimental Setup

We use the AllenAct [44] framework to train models for the tasks ObjectNav and PointNav tasks in the iTHOR rooms within AI2Thor [20]. For both tasks, we use the same split of rooms for training and validation.

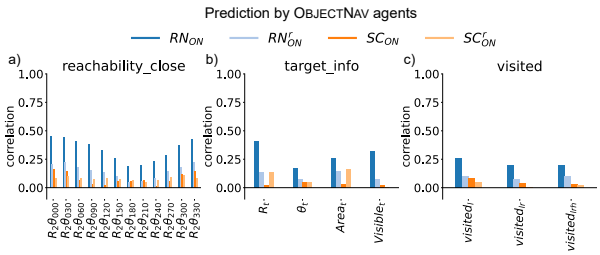

Figure 4. **Metadata prediction by ObjectNav GRU units:** a) Reachability b) Target information c) Visited history

4.1. ObjectNav Models and Baselines

We consider two models for ObjectNav. The first model uses a frozen ResNet18 as the visual encoder and is named RN_{ON}, while the second uses a 5 layer CNN (referred to as SimpleConv) as the visual encoder, denoted by SC_{ON}. In SC_{ON}, the visual encoder is optimized using the gradients of the actor critic loss. The visual representation is concatenated with the goal embedding which is then fed to a GRU. The GRU is connected to two linear layers predicting the policy and value. To ascertain if the representations learned by ObjectNav agents are due to training, we consider two randomly initialized models with the same architetcures as the baselines. For the random ResNet model, named RN^r_{ON}, we initialize ResNet with ImageNet weights and initialize the GRU randomly. For the random SimpleConv model, named SC^r_{ON}, both the visual encoder and GRU are initialized randomly. RN_{ON} and SC_{ON} are trained for 300 Million steps using the default hyperparameters from the AllenAct framework.

4.2. PointNav Models and Baselines

Similar to ObjectNav models we consider a ResNet based model (RN_{PN}) and a SimpleConv based model (SC_{PN}). The distance and orientation to target are used as a sensory input to the model for target information. The corresponding random baselines are named RN^r_{PN} and SC^r_{PN}. RN_{PN} and SC_{PN} are trained for 300 Million steps using the default hyperparameters from AllenAct.

	ObjectNav		**PointNav**	
	ResNet18	SimpleConv	ResNet18	SimpleConv
Trained	RN_{ON}	SC_{ON}	RN_{PN}	SC_{PN}
Random	RN^r_{ON}	SC^r_{ON}	RN^r_{PN}	RN^r_{PN}

4.3. Human Trajectories

After training the ObjectNav and PointNav models, we collect human sampled trajectories for the training and validation rooms. The training trajectories contain 59 episodes with average episode length of 480 while validation trajectories contain 42 episodes with average episode

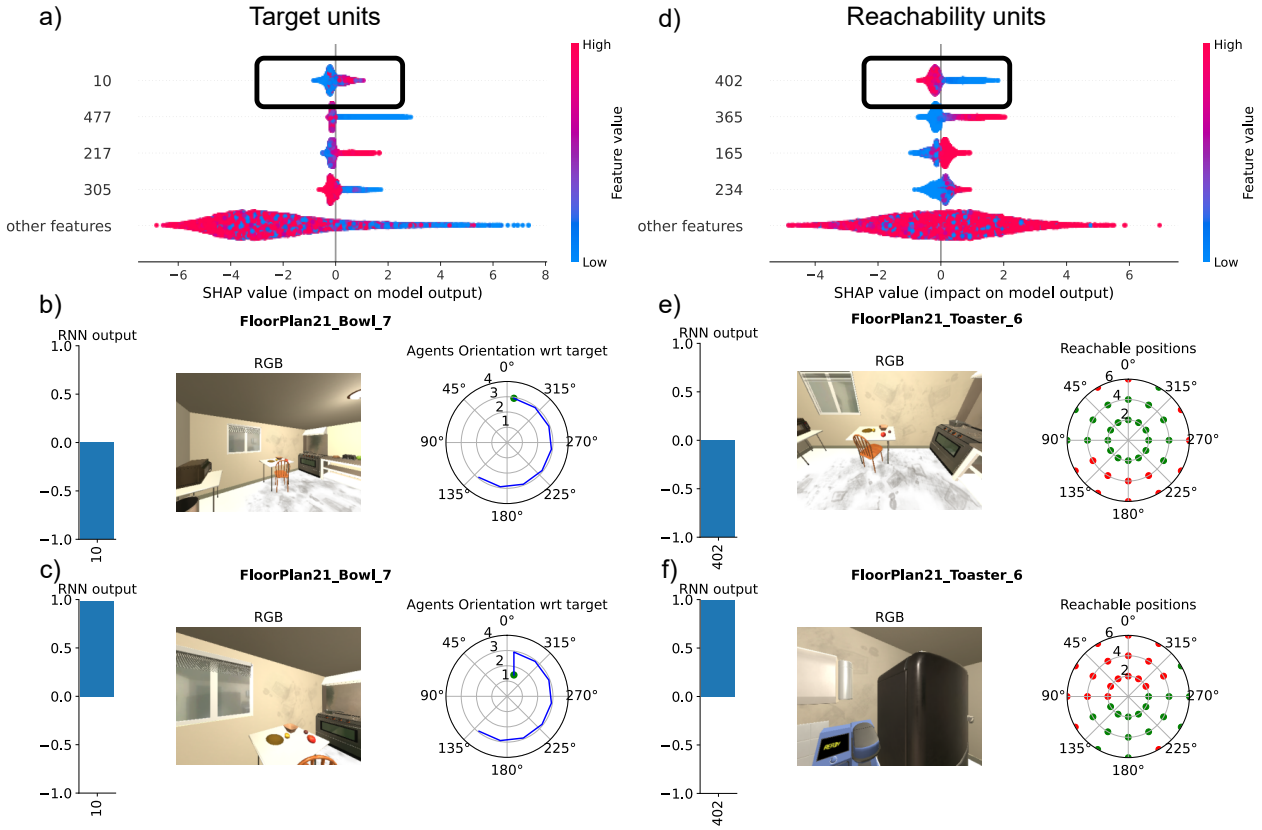

Figure 5. **Visualization of hidden units.** a) Target Visibility Unit: Top-4 most relevant units to predict target distance. b) The bar plot on left shows unit 10's (target unit) response. The center image is agent's current observation. The polar plot on right shows the distance (in meters) and orientation of the agent (in degrees) wrt target. In this case, the agent is at around 3 meters away from the target and is oriented around 0 degrees. The response of unit 10 is negative. c) In this case, the agent is now closer to target (around 1 meters) and unit 10's (target unit) response is positive. d) Reachability Unit: Top-4 most relevant hidden units to predict reachability at distance $2 \times gridSize$ and theta zero. e) The bar plot on the left shows unit 402's (reachability unit) response. The polar plot on right shows if the locations at radii of $2, 4, 6 \times gridSize$ and a given orientation in degrees are reachable or not. In this case, all the locations ahead are green i.e. reachable. The response of unit 402 is negative. f) All the locations ahead are red i.e. not reachable. The response of unit 402 is positive.

length of 470. The subject was encouraged to completely explore the rooms with intentional collisions and visits to previously visited locations with an episode length upper limit of 500. All 8 models are forced to follow these trajectories. The corresponding metadata and GRU activity was extracted resulting in 28,000 training samples and 20,000 validation samples for GBT training.

5. Results

5.1. ObjectNav

The validation performance of ObjectNav models saturates at around 50 million steps, therefore we select a checkpoint right after 50 million steps from both models. RN_{ON} (success = 0.458, SPL = 0.23) significantly outperforms SC_{ON} (success = 0.124, SPL = 0.056). Here success indicates the fraction of episodes the agent successfully reached the target and SPL refers to Success weighted by Path Length introduced in [2]. We consider concepts derived from metadata that are related to target information (R_t,θ_t,$\texttt{visible}_t$,$\texttt{Area}_t$) , reachability ($R_r\theta_{angle}$ where r is the radius and angle is the orientation of the neighboring grid point w.r.t. the agent), agent's information (R_a,θ_a) and visited history ($\texttt{visited}_l$,$\texttt{visited}_{lr}$,$\texttt{visited}_{lrh}$).

Metadata prediction: We train GBTs to predict metadata from the GRU units. We observe that RN_{ON} predicts reachability much better than the other three ObjectNav models (Figure 4a) with a correlation of 0.45 and ROC_AUC=0.75 for reachability in front ($R_2\theta_{000}$). We also observe an interesting pattern that prediction of reachability drops as one moves from 0 (front of the agent) to 180 (behind) degree then it starts increasing from 180 to 330 degrees suggesting the reachability of locations in front is more predictable than behind the agent. In Figure 4a, we show the results for reachability with radius = $2 \times gridsize$. We observe a similar pattern for radius = $4 \times gridsize$ and radius = $6 \times gridsize$ (refer to Appendix B).

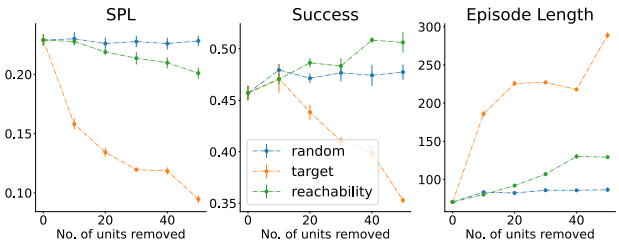

Figure 6. **Impact of removing units from RN_{ON}.**

For the target information (R_t,θ_t,$\texttt{Area}_t$,$\texttt{visible}_t$) RN_{ON} shows a higher correlation than the other three models (Figure 4b). Visited history also ($\texttt{visited}_l$,$\texttt{visited}_{lr}$,$\texttt{visited}_{lrh}$) shows a higher correlation for RN_{ON} (Figure 4c). The agent's information (R_a,θ_a) is not predicted well and RN_{ON} model shows a correlation similar to baselines suggesting this information is not learned by the agent during training (refer to Appendix B). Overall, we observe that RN_{ON} learns the reachability, target relevant information and visited history from ObjectNav training. This suggests that these three features are very crucial for performing this task.

While we present the results only on four concepts we also considered collision but found that it was poorly predicted for all 4 models (refer to Appendix B).

Hidden unit visualization: To identify which hidden units are relevant to the mentioned concepts we apply SHAP on the two most interesting concepts ($\texttt{visible}_t$ and $R_2\theta_{000}$). In Figure 5a we show the top-4 units that are most relevant in predicting the target visibility. On observing the SHAP plot of unit 10 (Figure 5a) we see that when the unit's value is higher it has a positive impact on target visibility and vice-versa suggesting that the unit's value is high when the target is visible (for aggregate SHAP values over units see Appendix E). The polar plots show the agent's trajectory (Figure 5 b,c), blue line represents trajectory and green dot indicates the agent's current location wrt target. Bar plot shows the RNN unit's response for current observation. Here, the target is a bowl; when the agent is away from the target its response is negative (Figure 5b) and when it is closer its response is positive (Figure 5c). These results also suggest that this unit might be positively correlated to target visibility.

In Figure 5d, we show the top-4 units most relevant in predicting $R_2\theta_{000}$ (for distribution of aggregate SHAP values over units see Appendix E). On observing the SHAP plot of unit 402 (Figure 5d) we can see that when the unit's value is higher it has negative impact on $R_2\theta_{000}$ and vice-versa suggesting that the unit value is high when the location ahead is not reachable. In Figure 5e,f, the dots are located at $radii = 2, 4, 6 \times stepsize$ from the agent and at angles from 0 to 330 in steps of 30°, where 0 is the front of the agent. Dot color indicates if the location is reachable (green) or not (red). Here, when the location in front of the agent is reachable the unit's response is negative (Figure 5e) and when there is an obstacle in front the unit's response is positive (Figure 5f). These results suggest that this unit might be detecting obstacles ahead.

Unit ablation: While SHAP provides a way to quantify the impact of hidden units on the prediction of a particular metadata concept, it does not imply causality. To identify causality we perform an ablation and measure the impact on the evaluation metrics. We remove units relevant to $\texttt{visible}_t$ and $R_2\theta_{000}$ prediction and measure the impact on the model's performance in terms of SPL, success, and episode length. We compare the ablation results to removing a random selection of units as a baseline. To remove a unit, we set the unit's activity as a constant that is equal to the mean of that unit's activity over the training episodes.

In Figure 6, we observe that removing only 10 target units leads to a huge drop in SPL as compared to removing as many as 50 random units or units encoding reachability. As we remove more target units, the success also begins to drop. This suggests that target units are crucial and removing them first deteriorates the agents ability to identify targets thus leading to longer episodes and low SPL scores and beyond a certain point, the agent ability to be successful is also affected. Removing reachability units also leads to drop in SPL but the impact is not as drastic as in the case of target units. Interestingly removing reachability units lead to increase in success rate potentially due to an increase in exploration. Removing randomly selected units do not significantly impact any of the performance measures.

5.2. PointNav

Similar to ObjectNav, we choose checkpoints after 50 million steps for our PointNav models. RN_{PN} (success = 0.925, SPL = 0.755) and SC_{PN} (success = 0.878, SPL = 0.712) are highly successful at this task. We consider concepts derived from metadata that are related to target information (R_t,θ_t) , reachability ($R_r\theta_{angle}$ where r is the radius and angle is the orientation of the neighboring grid point with respect to the agent), agent's information (R_a,θ_a) and visited history ($\texttt{visited}_l$,$\texttt{visited}_{lr}$,$\texttt{visited}_{lrh}$).

Metadata prediction: We train the GBTs to predict metadata from the GRU units. We first observe from Figure 7a (left) that reachability is predicted at all the angles well. Another interesting thing to note is that models that are not even trained on the PointNav task (RN^r_{PN} and SC^r_{PN}) can predict reachability. This result is surprising as compared to ObjectNav, where the only model that predicted reachability well was the one that performed well on the ObjectNav task (RN_{ON}). Further, RN_{ON} only predicted the reachability in the view of the agent. Our intuition for the above result is that this could be due to additional information from GPS + compass sensor that provides the distance

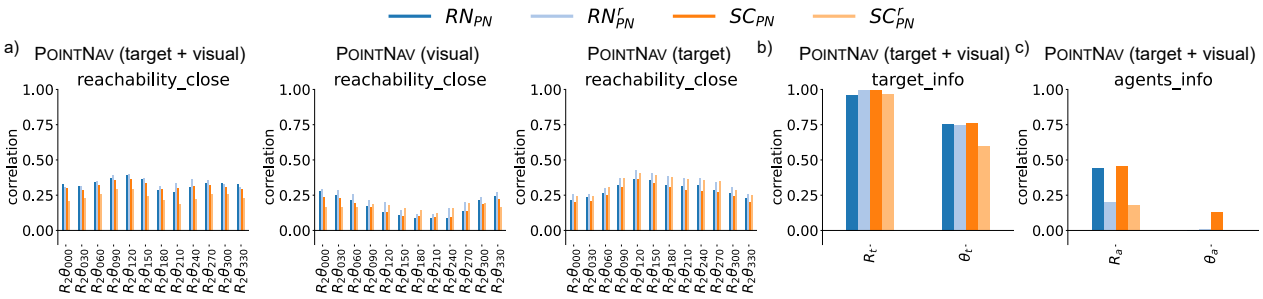

Figure 7. **Metadata prediction by PointNav GRU units:** a) Reachability b) Target information and c) Agent information

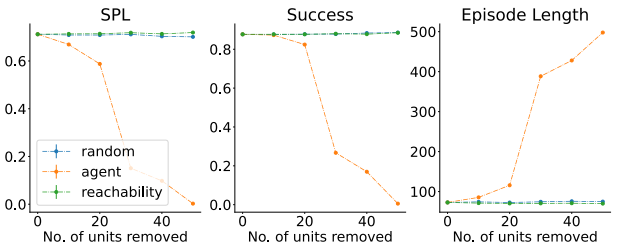

Figure 8. **Impact of removing units from SC_{PN}.**

and orientation of the target. To tease apart the prediction due to visual sensor and GPS sensor we perform an ablation study where in one case we replace the output of the GPS sensor by random noise (visual-only; Figure 7a center) and in the other we replace the image with all zeros (only GPS; Figure 7a right).

In the visual-only case, we now observe a pattern similar to ObjectNav where reachability in the field of view is more predictable than out of view. However, it is important to note that prediction of reachability does not improve with training RN_{PN} suggesting that ResNet with ImageNet weights is sufficient to predict reachability required to solve PointNav. SC^r_{PN} however does not seem to predict front reachability ($R_2\theta_{000}$) as effectively as SC_{PN} suggesting that a random initialization is not sufficient to predict reachability required to solve PointNav.

In the target-only case, we observe that the reachability of the backside of the agent is more predictable compared to the angles in the field of view. One possible explanation for this could be that when the distance between target and the agent changes in a given step that means the position at the back was reachable since the agent was there in the previous step. Therefore, using the change in GPS sensor values reachability at back can be predicted in some cases.

The target distance and orientation is predictable when the GPS sensor is available for all the models (Figure 7b and Appendix C). This finding is expected as we provide this information as input, and when the GPS sensor is noise it can not be predicted (refer to Appendix C). Interestingly when the GPS sensor is available (Figure 7c and Appendix C), hidden units in trained PointNav models can predict the distance of the agent (R_a) from the initial spawn location. When using the SHAP method to find the relevant units for predicting R_a, we observe that top most relevant units have a constant value (refer to Appendix D) at almost every step in the episode and show very low variance in its output. On further inspection, we found that the 2 units in top-20 most relevant units for R_a prediction were also relevant for target distance R_t prediction. To predict R_a, GBT might be using a combination of a constant unit(s) and unit that encodes the target information.

Unit ablation: Similar to ObjectNav we perform ablations by removing units and measuring the impact on the metrics. As shown in Figure 8 removing random and reachability units have almost no impact on the performance. Even after removing 50 units we observe similar performance on all three metrics. Removing the units that are relevant for predicting R_a causes a significant drop in the performance and on dropping 50 units both SPL and success rate almost reach zero. The episode length also reaches the highest possible value (500) set in the task definition i.e. the episode ends if agent takes 500 steps. On further inspection, we found that in top-50 R_a units, there are 6 units from the top-50 R_t units. This is the key reason why PointNav performance dropped as the target distance information is lost. We further performed an ablation by removing only these 6 target units, which resulted in a drastic drop.

6. Conclusion

We propose iSee to investigate if concepts about the agent, environment and task are encoded in the hidden representation of embodied agents. While we focus on visual navigation agents trained in AI2-THOR, the framework is generic and can be applied to agents trained on any task in any virtual environment with relevant metadata available. Our analysis shows the ObjectNav agent encodes target orientation, reachability and visited locations history in order to avoid obstacles and visiting the same locations repeatedly. PointNav agents encode target orientation and its progress towards the target and show less reliance on visual information.

References

[1] Treeinterpreter. https://github.com/andosa/treeinterpreter. [Accessed Nov, 2021]. 2

[2] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *arXiv*, 2018. 1, 2, 3, 6

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 1

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 2015. 2

[5] Dhruv Batra, Angel X. Chang, Sonia Chernova, Andrew J. Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, Manolis Savva, and Hao Su. Rearrangement: A challenge for embodied AI. *arXiv*, 2020. 1

[6] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv*, 2020. 2, 3

[7] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017. 1, 2

[8] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, 2020. 3

[9] Prithvijit Chattopadhyay, Judy Hoffman, Roozbeh Mottaghi, and Aniruddha Kembhavi. Robustnav: Towards benchmarking robustness in embodied navigation. *ArXiv*, abs/2106.04531, 2021. 3

[10] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *CVPR*, 2021. 3

[11] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. In *ECCV*, 2020. 3

[12] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, 2021. 1

[13] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009. 2

[14] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *CVPR*, 2018. 1, 2

[15] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In *NeurIPS*, 2019. 2

[16] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. 2

[17] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv*, 2019. 2

[18] Unnat Jain, Luca Weihs, Eric Kolve, Ali Farhadi, Svetlana Lazebnik, Aniruddha Kembhavi, and Alexander Schwing. A cordial sync: Going beyond marginal policies for multi-agent embodied tasks. In *ECCV*, 2020. 1

[19] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2018. 2

[20] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 1, 2, 5

[21] Chih kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Pradeep Ravikumar, and Tomas Pfister. On completeness-aware concept-based explanations in deep neural networks. In *NeurIPS*, 2020. 2

[22] Juncheng Li, Xin Wang, Siliang Tang, Haizhou Shi, Fei Wu, Yueting Zhuang, and William Yang Wang. Unsupervised reinforcement learning of transferable meta-skills for embodied navigation. In *CVPR*, 2020. 3

[23] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2020. 1, 2, 4, 11

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017. 2

[25] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019. 1, 2

[26] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual navigation with spatial attention. In *CVPR*, 2021. 3

[27] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. In *ICRA*, 2019. 3

[28] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *NeurIPS*, 2016. 1, 2

[29] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. 2

[30] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv*, 2019. 2

[31] Shivansh Patel, Saim Wani, Unnat Jain, Alexander Schwing, Svetlana Lazebnik, Manolis Savva, and Angel X. Chang. Interpretation of emergent communication in heterogeneous collaborative embodied agents. In *ICCV*, 2021. 2

[32] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *ECCV*, 2020. 3

[33] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *CVPR*, 2020. 2

[34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *KDD*, 2016. 2

[35] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 2

[36] Lloyd S Shapley. *17. A value for n-person games*. 2016. 3, 4

[37] Bokui Shen, Fei Xia, Chengshu Li, Roberto Mart'in-Mart'in, Linxi (Jim) Fan, Guanzhi Wang, S. Buch, Claudia. Pérez D'Arpino, Sanjana Srivastava, Lyne P. Tchapmi, Micael Edmond Tchapmi, Kent Vainio, Li Fei-Fei, and Silvio Savarese. igibson, a simulation environment for interactive tasks in large realistic scenes. In *IROS*, 2021. 1, 2, 3

[38] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017. 2

[39] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv*, 2013. 2

[40] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv*, 2017. 2

[41] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. In *NeurIPS*, 2020. 3

[42] Donglai Wei, Bolei Zhou, Antonio Torrabla, and William Freeman. Understanding intra-class knowledge inside cnn. *arXiv*, 2015. 2

[43] Luca Weihs, Aniruddha Kembhavi, Kiana Ehsani, Sarah Pratt, Winson Han, Alvaro Herrasti, Eric Kolve, Dustin Schwenk, Roozbeh Mottaghi, and Ali Farhadi. Learning generalizable visual representations via interactive gameplay. In *ICLR*, 2021. 2

[44] Luca Weihs, Jordi Salvador, Klemen Kotar, Unnat Jain, Kuo-Hao Zeng, Roozbeh Mottaghi, and Aniruddha Kembhavi. Allenact: A framework for embodied ai research. *arXiv*, 2020. 5

[45] Erik Wijmans, Abhishek Kadian, Ari S. Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *ICLR*, 2020. 3

[46] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *CVPR*, 2019. 3

[47] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. In *ICLR*, 2019. 3

[48] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary tasks and exploration enable objectgoal navigation. In *ICCV*, 2021. 2

[49] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv*, 2015. 1, 2

[50] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[51] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. 2015. 2

[52] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017. 3

4 Discussion and Outlook

4.1 Summary

This thesis aimed to pave the way forward in understanding the representations of the human visual cortex and deep neural networks trained for visual tasks. Our works from Chapter 1 and Chapter 2 demonstrated that DNNs trained on different tasks have the potential to reveal insights into representations of the human visual cortex and DNNs. Our work from Chapter 3 exploited the untapped potential of simulation engines to develop a new method for interpreting representations of hidden neurons of DNNs. We also demonstrated the potential of the proposed new approach in revealing the new insights into representations learned by navigation models.

4.1.1 Insights into human visual cortex representations

In Chapter 1, we first investigated the scene-selective regions OPA and PPA by comparing their representations to DNNs trained on scene classification task and scene parsing task and randomly initialized DNNs as a baseline. We first observed that both scene classification and scene-parsing DNNs explained OPA and PPA responses better than randomly initialized DNNs. The results suggest that training on scene classification and scene parsing tasks brings DNN representation closer to representations in OPA and PPA. We further found that the variance of OPA and PPA responses were better explained uniquely by scene-parsing DNNs than scene-classification DNNs, suggesting the representation in OPA and PPA is closer to scene-parsing DNNs. A key difference between scene parsing and scene classification tasks is that scene parsing requires identifying which components are present in the scene and where they are present, while scene classification only requires identifying the type of scene. Therefore, our results suggest that spatial information of where

the objects are located in the scene is also present in scene-selective regions.

In the second part of chapter 1, we focused on the entire visual cortex and increased the number of DNNs to 18 to focus on different aspects of scene understanding (2D, 3D, and semantic). We found a systematic mapping of tasks on the visual cortex with 2D DNNs best explaining the early visual cortex, 3D DNNs best explaining the dorsal regions, and semantic DNNs best explaining the ventral regions. Overall our findings converged with well-established neuroscience theory [89] that proposes two streams in the visual cortex: "What" stream in the ventral regions and "where" stream in the dorsal regions. "What" represents what type of objects are present in the scene and therefore is related to semantic information about the scene. "Where" represents the spatial location of different objects in the scene and therefore is related to 3D information of the scene. While our findings converged with dual-stream theory, using our proposed method, we were able to find new insights into representations of several brain regions that can advance the investigation of the human visual cortex by designing new neuroimaging experiments based on what we found.

4.1.2 Insights into DNN representations

In the first part of Chapter 2, we focused on finding how training on different tasks leads to different representations in the DNNs with identical architecture. We found that earlier layers of all the DNNs have very similar representations irrespective of the tasks they were trained on. On the other hand, the deeper layers were more task-specific and showed more diversity in the representation. The representations of deeper layers of the DNNs trained on related tasks were more similar than DNNs trained on less related tasks. In other words, the representations of DNNs trained on 3D tasks were more similar to other DNNs trained on 3D tasks than DNNs trained on 2D or semantic tasks. We observed a similar pattern in the case of DNNs trained on 2D and semantic DNNs.

We then showed that task similarity obtained using representational similarity analysis is highly related to transfer learning-based task similarity. Using the representational similarity analysis's relation to transfer learning, we showed that the proposed approach could be used for model selection in transfer learning.

In the second part of chapter 2, we defined a new representational similarity

measure called Duality Diagram Similarity that unifies other similarity measures. The duality diagram consists of multiple components. By investigating the different combinations of components, we showed that comparing the feature spaces after z-scoring leads to a very high correlation with transfer learning compared to previous works. We then used DDS to show that early layers of a Resnet50 [90] trained on Imagenet [38] (also Places [91]) classification task have representations similar to 2D DNNs, middle layers have representations similar to 3D DNNs, and deeper layers have representations similar to semantic DNNs. The above observation shows that DDS can be applied to branching location selection for transfer learning.

In the final chapter, we took inspiration from explainability research and developed a new method to find what concept a neuron encodes. We found that DNNs trained on the Objectnav task learn to encode target visibility, obstacles (or reachability), and history of visits to successfully navigate to the target object. Ablating target units significantly dropped the DNN's performance compared to ablating other irrelevant units, validating our approach's effectiveness in finding relevant units. DNNs trained on the Pointnav task learned to encode the agent's progress towards the target and showed less reliability on visual information. Overall, we observed that concepts are sparsely represented in units suggesting that models can be compressed easily by 25% without a noticeable change in performance.

Overall, the studies in this thesis revealed new insights into the representations of the visual cortex and DNNs. While the earlier chapters used an existing well-established method, the last work proposed a new approach to interpreting DNN representations and showed the potential of simulation engines in interpretability research.

4.2 Limitations

In the first chapter, we found representations in the human visual cortex in terms of 2D, 3D and semantic tasks. These tasks are only a fraction of all possible visual functions the human brain can perform. Further, the images used in these chapters were from indoor scenes. So, our findings are limited in the scope of tasks considered and only valid for indoor scene perception. Although we expect the representational insights we found to be consistent with the image domain, we make

no claims whether our findings will be valid for outdoor scenes, videos, and active tasks where humans can interact with the scenes. Another limitation was that the brain responses were collected on only 50 selected images, so we are not sure what these results will look like for a larger sample.

In the second chapter as well, we had the same limitations regarding the number of tasks considered. The number of computer vision tasks considered is only a fraction of all possible computer vision tasks. Although we would like to extend our approach to all possible computer vision tasks, the datasets for different tasks are usually different. This leads to ambiguity about whether the difference in the representation of DNNs is due to the training dataset or task. Further, the task similarities we obtained were for the tasks in indoor scenes, so again, similar to chapter 1, we make no claims whether our results will be consistent for the same task setting in outdoor images.

In the third chapter, the number of concepts we looked at was not exhaustive for navigation. The concepts we chose were what we thought were relevant for performing Objectnav and Pointnav tasks, but the agent might be learning some other concepts and possible biases in the dataset. We only looked at the baseline models and not the state-of-the-art models on these tasks: those could have led to a better understanding of what concepts are relevant for solving these tasks. Again, the number of tasks we investigated here was limited.

4.3 Future directions

Although this work has made progress towards understanding representations in both the human visual cortex and DNNs, it has also opened several new directions to explore. We discuss some of these possible directions below:

4.3.1 New brain datasets

The approach we presented to find the representation of a brain region in terms of the task a DNN was trained for is not limited to indoor scene perception. One can apply a similar approach to find representational insights about the auditory cortex, language regions, hippocampus, and prefrontal cortex. Even in the visual cortex, one can use a larger dataset like BOLD 5000 [92] or NSD [93] and probe the visual

regions using DNNs trained on different tasks from MSCOCO dataset [39]. Another large-scale dataset of fMRI responses to video clips [34] can be used to probe which regions are involved in motion, event understanding, and action recognition. Almost all the fMRI datasets for vision involve passive viewing of images, so there is also a need for a collection of new brain recordings where humans actively engage in the environments, such as playing a video game or performing a navigation task in photorealistic simulation environments.

4.3.2 New DNNs

Recently there is a growing interest in new DNN architectures for vision tasks such as vision transformers [63], swin transformers [94], MLP mixers [95], hybrids of convolutions and transformers [96] and many others. These architectures lead to better performance on several benchmarks, and hence it would be interesting to apply methods such as representational similarity analysis or neuron-to-concept mapping to find out how different the representations are in these architectures. Another hot research area is self-supervised learning [97, 98, 99, 100], where people focus on designing new tasks without the need for human annotations or labels. The self-supervised learning has shown tremendous potential in learning generic representations that outperform DNNs trained in a supervised manner in transfer learning scenarios. What makes these representations more transferable and better than supervised DNNs? One might consider applying the approaches presented in this thesis to answer these questions.

4.3.3 Transfer learning

While in this work, we investigated transfer from the models trained on Taskonomy datasets or Imagenet/Places datasets, the number of new DNN architectures and self-supervised DNNs has grown exponentially. These new DNN architectures and self-supervised DNNs have shown promising results on transfer to downstream tasks. Further, new benchmarks have been created to compare transferability estimates, such as one proposed in Agostinelli et al. [101] investigating transferring from an ensemble of models. Hence, extending the presented methods in this thesis may allow selecting multiple models for transfer learning.

4.3.4 Tapping the full potential of simulation engines

Most DNN interpretability research and research on the human visual cortex focuses on passive image/video viewing. In both cases, either a human or a DNN processes the visual input passively without engaging with the environment. These passive viewing datasets are usually selected from natural images with limited groundtruth annotations. The tasks that one can use for training a DNN are also less related to how humans interact with the world. The simulation engines such as AI2Thor [1] and Habitat [77] provide photorealistic scenes where humans/DNNs can interact with the world to perform more complex tasks such as navigation, following language instructions, rearranging a room, and interacting with objects, that are more related to how humans perform the tasks in the natural world. Therefore, there is a lot of potential in designing new tasks in these simulation environments that are related to human learning and can be explored via new brain recordings to provide wholesome insights into how humans/DNNs solve these complex tasks.

A List of Figures

B Bibliography

[1] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[2] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, 2020.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, 2012.

[4] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[5] L. Deng and Y. Liu, *Deep learning in natural language processing*. Springer, 2018.

[6] F. Wang, L. P. Casalino, and D. Khullar, "Deep learning in medicine—promise, progress, and challenges," *JAMA internal medicine*, vol. 179, no. 3, pp. 293–294, 2019.

[7] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[8] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, *et al.*, "Magnetic

control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414–419, 2022.

[9] M. Schrimpf, J. Kubilius, M. J. Lee, N. A. R. Murty, R. Ajemian, and J. J. DiCarlo, "Integrative benchmarking to advance neurally mechanistic models of human intelligence," *Neuron*, 2020.

[10] K. Seeliger, L. Ambrogioni, Y. Güçlütürk, L. M. van den Bulk, U. Güçlü, and M. van Gerven, "End-to-end neural system identification with neural information flow," *PLOS Computational Biology*, vol. 17, no. 2, p. e1008558, 2021.

[11] P. Bashivan, K. Kar, and J. J. DiCarlo, "Neural population control via deep image synthesis," *Science*, vol. 364, no. 6439, p. eaav9436, 2019.

[12] C. R. Ponce, W. Xiao, P. F. Schade, T. S. Hartmann, G. Kreiman, and M. S. Livingstone, "Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences," *Cell*, vol. 177, no. 4, pp. 999–1009, 2019.

[13] M. Khosla and L. Wehbe, "High-level visual areas act like domain-general filters with strong selectivity and functional specialization," *bioRxiv*, 2022.

[14] K. Grill-Spector and R. Malach, "The Human Visual Cortex," *Annual Review of Neuroscience*, vol. 27, no. 1, pp. 649–677, 2004. _eprint: https://doi.org/10.1146/annurev.neuro.27.070203.144220.

[15] B. A. Wandell, S. O. Dumoulin, and A. A. Brewer, "Visual field maps in human cortex," *Neuron*, vol. 56, no. 2, pp. 366–383, 2007.

[16] B. A. Wandell and J. Winawer, "Imaging retinotopic maps in the human brain," *Vision research*, vol. 51, no. 7, pp. 718–737, 2011.

[17] K. Grill-Spector, Z. Kourtzi, and N. Kanwisher, "The lateral occipital complex and its role in object recognition," *Vision Research*, vol. 41, pp. 1409–1422, May 2001.

[18] N. Kanwisher and G. Yovel, "The fusiform face area: a cortical region specialized for the perception of faces," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 361, no. 1476, pp. 2109–2128, 2006.

[19] R. Epstein, A. Harris, D. Stanley, and N. Kanwisher, "The parahippocampal place area: Recognition, navigation, or encoding?," *Neuron*, vol. 23, no. 1, pp. 115–125, 1999.

[20] S. V. Astafiev, C. M. Stanley, G. L. Shulman, and M. Corbetta, "Extrastriate body area in human occipital cortex responds to the performance of motor actions," *Nature neuroscience*, vol. 7, no. 5, pp. 542–548, 2004.

[21] E. L. Josephs and T. Konkle, "Large-scale dissociations between views of objects, scenes, and reachable-scale environments in visual cortex," *Proceedings of the National Academy of Sciences*, vol. 117, no. 47, pp. 29354–29362, 2020.

[22] K. Vinken, T. Konkle, and M. Livingstone, "The neural code for'face cells' is not face specific," *bioRxiv*, 2022.

[23] D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8619–8624, 2014.

[24] S.-M. Khaligh-Razavi and N. Kriegeskorte, "Deep supervised, but not unsupervised, models may explain it cortical representation," in *PLoS Computational Biology*, 2014.

[25] U. Guclu and M. A. J. van Gerven, "Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream," *Journal of Neuroscience*, vol. 35, pp. 10005–10014, July 2015.

[26] P. Agrawal, D. Stansbury, J. Malik, and J. L. Gallant, "Pixels to voxels: modeling visual representation in the human brain," *arXiv preprint arXiv:1407.5104*, 2014.

[27] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Comparison of deep neural networks to spatio-temporal cortical dynamics of human

visual object recognition reveals hierarchical correspondence," *Scientific reports*, vol. 6, p. 27755, 2016. Publisher: Nature Publishing Group.

[28] M. Eickenberg, A. Gramfort, G. Varoquaux, and B. Thirion, "Seeing it all: Convolutional network layers map the function of the human visual system," *NeuroImage*, vol. 152, pp. 184–194, 2017.

[29] U. Güçlü and M. A. J. van Gerven, "Modeling the dynamics of human brain activity with recurrent neural networks," vol. 11, no. February, pp. 1–14, 2016.

[30] H. Richard, A. Pinho, B. Thirion, and G. Charpiat, "Optimizing deep video representation to match brain activity," *arXiv preprint arXiv:1809.02440*, 2018.

[31] P. Mineault, S. Bakhtiari, B. Richards, and C. Pack, "Your head is there to move you around: Goal-driven models of the primate dorsal pathway," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[32] S. Bakhtiari, P. Mineault, T. Lillicrap, C. Pack, and B. Richards, "The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[33] R. M. Cichy, G. Roig, A. Andonian, K. Dwivedi, B. Lahner, A. Lascelles, Y. Mohsenzadeh, K. Ramakrishnan, and A. Oliva, "The algonauts project: A platform for communication between the sciences of biological and artificial intelligence," *arXiv preprint arXiv:1905.05675*, 2019.

[34] R. M. Cichy, K. Dwivedi, B. Lahner, A. Lascelles, P. Iamshchinina, M. Graumann, A. Andonian, N. Murty, K. Kay, G. Roig, *et al.*, "The algonauts project 2021 challenge: How the human brain makes sense of a world in motion," *arXiv preprint arXiv:2104.13714*, 2021.

[35] N. Kriegeskorte, M. Mur, D. A. Ruff, R. Kiani, J. Bodurka, H. Esteky, K. Tanaka, and P. A. Bandettini, "Matching categorical object representations in inferior temporal cortex of man and monkey," *Neuron*, vol. 60, no. 6, pp. 1126–1141, 2008.

[36] N. A. Ratan Murty, P. Bashivan, A. Abate, J. J. DiCarlo, and N. Kanwisher, "Computational models of category-selective brain regions enable high-throughput tests of selectivity," *Nature Communications*, vol. 12, no. 1, pp. 1–14, 2021.

[37] Z. Gu, K. W. Jamison, M. Khosla, E. J. Allen, Y. Wu, T. Naselaris, K. Kay, M. R. Sabuncu, and A. Kuceyeski, "Neurogen: activation optimized image synthesis for discovery neuroscience," *NeuroImage*, vol. 247, p. 118812, 2022.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[39] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[41] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303–338, June 2010.

[42] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[43] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[44] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," *arXiv preprint arXiv:2203.05482*, 2022.

[45] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*, pp. 239–248, IEEE, 2016.

[46] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

[47] M. Schrimpf, J. Kubilius, H. Hong, N. J. Majaj, R. Rajalingham, E. B. Issa, K. Kar, P. Bashivan, J. Prescott-Roy, K. Schmidt, *et al.*, "Brain-score: Which artificial neural network for object recognition is most brain-like?," *BioRxiv*, p. 407007, 2018.

[48] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[49] N. Kriegeskorte, M. Mur, and P. A. Bandettini, "Representational similarity analysis-connecting the branches of systems neuroscience," *Frontiers in systems neuroscience*, vol. 2, p. 4, 2008. Publisher: Frontiers.

[50] A. Nguyen, J. Yosinski, and J. Clune, "Understanding neural networks via feature visualization: A survey," in *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 55–76, Springer, 2019.

[51] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[52] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai—explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, p. eaay7120, 2019.

[53] P. Robert and Y. Escoufier, "A unifying tool for linear multivariate statistical methods: the rv-coefficient," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 25, no. 3, pp. 257–265, 1976.

[54] N. Kriegeskorte, M. C. Mur, and P. A. Bandettini, "Representational similarity analysis – connecting the branches of systems neuroscience," *Frontiers in Systems Neuroscience*, vol. 2, pp. 1480 – 1494, 2008.

[55] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*, pp. 3519–3529, 2019.

[56] I. I. Groen, M. R. Greene, C. Baldassano, L. Fei-Fei, D. M. Beck, and C. I. Baker, "Distinct contributions of functional and deep neural network features to representational similarity of scenes in human brain and behavior," *Elife*, vol. 7, p. e32962, 2018.

[57] M. F. Bonner and R. A. Epstein, "Coding of navigational affordances in the human visual system," *Proceedings of the National Academy of Sciences*, vol. 114, no. 18, pp. 4793–4798, 2017. Publisher: National Acad Sciences.

[58] M. Tsantani, N. Kriegeskorte, K. Storrs, A. L. Williams, C. McGettigan, and L. Garrido, "Ffa and ofa encode distinct types of face identity information," *Journal of Neuroscience*, vol. 41, no. 9, pp. 1952–1969, 2021.

[59] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, pp. 6076–6085, 2017.

[60] J. Mehrer, C. J. Spoerer, N. Kriegeskorte, and T. C. Kietzmann, "Individual differences among deep neural network models," *Nature communications*, vol. 11, no. 1, pp. 1–12, 2020.

[61] T. Nguyen, M. Raghu, and S. Kornblith, "Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth," *arXiv preprint arXiv:2010.15327*, 2020.

[62] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[63] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[64] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.

[65] L. Wang, R. E. B. Mruczek, M. J. Arcaro, and S. Kastner, "Probabilistic Maps of Visual Topography in Human Cortex," *Cerebral Cortex*, vol. 25, pp. 3911–3931, Oct. 2015. Publisher: Oxford Academic.

[66] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, p. 574, 1959.

[67] C. G. Gross, "Genealogy of the "grandmother cell"," *The Neuroscientist*, vol. 8, no. 5, pp. 512–518, 2002.

[68] R. Q. Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried, "Invariant visual representation by single neurons in the human brain," *Nature*, vol. 435, no. 7045, pp. 1102–1107, 2005.

[69] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[70] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv*, 2015.

[71] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," in *NeurIPS*, 2016.

[72] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4467–4477, 2017.

[73] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," 2015.

[74] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *CVPR*, 2017.

[75] E. Hernandez, S. Schwettmann, D. Bau, T. Bagashvili, A. Torralba, and J. Andreas, "Natural language descriptions of deep visual features," *arXiv preprint arXiv:2201.11114*, 2022.

[76] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *CVPR*, 2018.

[77] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *ICCV*, 2019.

[78] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

[79] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv*, 2017.

[80] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models," *arXiv*, 2019.

[81] S.-A. Rebuffi, R. Fong, X. Ji, and A. Vedaldi, "There and back again: Revisiting backpropagation saliency methods," in *CVPR*, 2020.

[82] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, 2015.

[83] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *KDD*, 2016.

[84] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NeurIPS*, 2017.

[85] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *ICML*, 2017.

[86] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *ICML*, 2018.

[87] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards automatic concept-based explanations," in *NeurIPS*, 2019.

[88] "Treeinterpreter." `https://github.com/andosa/treeinterpreter`. [Accessed Nov, 2021].

[89] M. Mishkin and L. G. Ungerleider, "Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys," *Behavioural Brain Research*, vol. 6, pp. 57–77, Sept. 1982.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[91] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[92] N. Chang, J. A. Pyles, A. Marcus, A. Gupta, M. J. Tarr, and E. M. Aminoff, "Bold5000, a public fmri dataset while viewing 5000 visual images," *Scientific data*, vol. 6, no. 1, pp. 1–18, 2019.

[93] E. J. Allen, G. St-Yves, Y. Wu, J. L. Breedlove, J. S. Prince, L. T. Dowdle, M. Nau, B. Caron, F. Pestilli, I. Charest, *et al.*, "A massive 7t fmri dataset to bridge cognitive neuroscience and artificial intelligence," *Nature neuroscience*, vol. 25, no. 1, pp. 116–126, 2022.

[94] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

[95] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "Mlp-mixer: An all-mlp

architecture for vision," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[96] Z. Dai, H. Liu, Q. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[97] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[98] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[99] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9912–9924, 2020.

[100] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

[101] A. Agostinelli, J. Uijlings, T. Mensink, and V. Ferrari, "Transferability metrics for selecting source model ensembles," *arXiv preprint arXiv:2111.13011*, 2021.