Neural Networks 173 (2024) 106179

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Fading memory as inductive bias in residual recurrent networks

Igor Dubinin [a,b,*], Felix Effenberger [a]

[a] Ernst Strüngmann Institute, Deutschordenstraße 46, Frankfurt am Main, 60528, Germany
[b] Frankfurt Institute for Advanced Studies, Ruth-Moufang-Straße 1, Frankfurt am Main, 60438, Germany

ARTICLE INFO

Keywords:
Recurrent neural network
Inductive bias
Residual connection
Memory

ABSTRACT

Residual connections have been proposed as an architecture-based inductive bias to mitigate the problem of exploding and vanishing gradients and increased task performance in both feed-forward and recurrent networks (RNNs) when trained with the backpropagation algorithm. Yet, little is known about how residual connections in RNNs influence their dynamics and fading memory properties. Here, we introduce weakly coupled residual recurrent networks (WCRNNs) in which residual connections result in well-defined Lyapunov exponents and allow for studying properties of fading memory. We investigate how the residual connections of WCRNNs influence their performance, network dynamics, and memory properties on a set of benchmark tasks. We show that several distinct forms of residual connections yield effective inductive biases that result in increased network expressivity. In particular, those are residual connections that (i) result in network dynamics at the proximity of the edge of chaos, (ii) allow networks to capitalize on characteristic spectral properties of the data, and (iii) result in heterogeneous memory properties. In addition, we demonstrate how our results can be extended to non-linear residuals and introduce a weakly coupled residual initialization scheme that can be used for Elman RNNs.

1. Introduction

The power of artificial neural networks in solving tasks lies in their universal approximation abilities, which is commonly referred to as *theoretical expressivity* (Barron, 1994; Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe, & White, 1989). However, the practical solutions to which networks can converge in a reasonable number of training iterations of a typically gradient-based learning scheme such as backpropagation, the *practical expressivity* of a network, have been shown to lag behind their theoretical expressivity (Hanin & Rolnick, 2019). Practical expressivity is determined by a set of inductive biases that take the form of (i) network architecture, (ii) weight initialization methods, (iii) convergence properties and other specifics of the training procedure, and (iv) more generally, comprise anything that influences the space of mappings learnable by a given network in practice (Battaglia et al., 2018; Goyal & Bengio, 2022). As the bias–variance trade-off suggests, the right choice of inductive biases plays a crucial role for model performance because properly informed biases can improve the efficiency of learning, a serious constraint on network expressivity in practice (Kearns & Vazirani, 1994).

A celebrated example of feed-forward networks with an effective inductive bias in the form of a constrained network architecture are convolutional neural networks (LeCun, Bengio, et al., 1995). Another popular form of an architectural inductive bias are so-called *residual connections* (or skip connections) of deep feed-forward architectures. These have been shown to strongly increase performance for many architectures and are used, for example, in the U-Net (Ronneberger, Fischer, & Brox, 2015), ResNet (He, Zhang, Ren, & Sun, 2016) or Transformer (Vaswani et al., 2017) architectures. Such residual connections have been shown to prevent gradients from vanishing in deep feed-forward networks, thereby mitigating one aspect of the well-studied exploding and vanishing gradients problem (EVGP) that appears in practice when training deep networks with the backpropagation algorithm (Glorot & Bengio, 2010).

When considering inductive biases in recurrent neural networks (RNNs), the crucial question is how these biases influence network dynamics and the resulting memory properties of the networks. Training an RNN with the backpropagation through time (BPTT) algorithm involves unrolling the RNN into a deep feed-forward network, so that networks trained on longer time series also face the EVGP (Pascanu, Mikolov, & Bengio, 2013). Historically, RNNs have been studied from a dynamical systems perspective (Bengio, Simard, & Frasconi, 1994; Hochreiter & Schmidhuber, 1997) and many ideas have been proposed to address the EVGP (Chang, Chen, Haber, & Chi, 2019; Erichson, Azencot, Queiruga, Hodgkinson, & Mahoney, 2020; Miller & Hardt, 2018; Schoenholz, Gilmer, Ganguli, & Sohl-Dickstein, 2016). Importantly, the

* Corresponding author at: Ernst Strüngmann Institute, Deutschordenstraße 46, Frankfurt am Main, 60528, Germany.
E-mail addresses: igor.dubinin@esi-frankfurt.de (I. Dubinin), felix.effenberger@esi-frankfurt.de (F. Effenberger).

https://doi.org/10.1016/j.neunet.2024.106179
Received 27 July 2023; Received in revised form 7 February 2024; Accepted 13 February 2024
Available online 15 February 2024

dynamical systems approach showed that RNN dynamics which are close to the point of a transition between stability and instability (the edge of chaos) are characterized by long-term fading memory and therefore efficient gradient propagation (Engelken, Wolf, & Abbott, 2020; Vogt, Touzel, Shlizerman, & Lajoie, 2020).

Although the influence of residual connections on RNN performance has been studied previously (Wang & Tian, 2016; Yue, Fu, & Liang, 2018), dynamics and memory properties of RNNs with residual connections have not been studied in detail. Here, we fill this gap and explore how residual connections in RNNs can result in inductive biases that influence the networks' dynamics and properties of their fading memory. Note that the present study was not primarily motivated by the goal of developing a model that achieves a new state of the art (SOTA) score in a number of benchmark tasks, but to study RNN dynamics and fading memory properties by means of Lyapunov exponents. The main contributions of this work are as follows.

- We extend on the connection between network dynamics, fading memory, and learning dynamics in RNNs discussed in Pascanu et al. (2013), showing that the fading memory properties of RNN dynamics result in temporally modulated learning rates.
- A new RNN architecture, the *weakly coupled residual recurrent network* (WCRNN), is introduced and proven to have stable and easily controllable memory properties by showing the existence of Lyapunov exponents of network dynamics. In particular, we demonstrate how the eigenvalues of the residual matrix control fading memory in WCRNNs. Additionally, we study WCRNNs with dynamics close to the edge of chaos and confirm the theoretically predicted trade-off between the efficiency (i.e. in how many training steps the network converges to a high-performing configuration) and the stability of learning.
- We show that informed residual connections and corresponding inductive biases result in higher practical expressivity of WCRNNs on a set of benchmark problems, assessed by learning efficiency and the best test accuracy achieved. In particular, we show how residuals resulting in weakly subcritical network dynamics allow the networks to benefit from long memory timescales, how residuals with rotational dynamics allow the networks to utilize spectral properties of the data samples, and how heterogeneous residuals allow the networks to capitalize on the resulting diversity of informed memory timescales.
- We show how results from WCRNNs with linear residuals can be generalized to the case of non-linear residuals and to the general case of a standard Elman RNN in the form of a weakly coupled residual initialization scheme.

In summary, our work demonstrates how Lyapunov exponents can be used to characterize fading memory resulting from residual connections and shows how informed residual connections can be used to achieve superior practical expressivity in RNNs.

2. Background and motivation

In this section, we discuss the connection between network dynamics, memory, and learning dynamics resulting from the training by backpropagation through time (BPTT). In Section 2.1, we introduce the concept of Lyapunov exponents and show how they determine memory timescales. In Section 2.2, we show how learning dynamics, mediated by BPTT, is influenced by the properties of fading memory.

2.1. Dynamical systems analysis

The memory of a system, commonly called fading memory in the context of recurrent neural networks, is a well-defined and thoroughly studied concept in the field of dynamical systems. From a dynamical system's perspective, a recurrent network is a non-autonomous, non-linear recurrent discrete map, the dynamics of which are given by

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{S}_t), \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^N$ is the network state at time t, $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ is non-linear function, $\mathbf{S}_t \in \mathbb{R}^N$ is the input to the network at time t and N is the dimensionality of the network state.

First, we consider the autonomous case, where the memory properties of network dynamics can be studied by means of perturbation theory and Lyapunov exponents. If we evolve an infinitesimal perturbation of the P-dimensional volume of the tangent space $\delta\mathbf{P}$, linearize it along this perturbation and apply the chain rule t times, we obtain

$$\delta\mathbf{P}(t+1) = \mathbf{V}_{\mathbf{x}}(\mathbf{f}^t(\mathbf{x}_1))\delta\mathbf{P}(1), \tag{2}$$

where $\mathbf{x}_1$ and $\delta\mathbf{P}(1)$ are the initial state and initial volume of the tangent space, $\mathbf{f}^t = \mathbf{f}\circ\ldots\circ\mathbf{f}$ denotes the t-fold iteration of the map $\mathbf{f}$, and $\mathbf{V}_x(\mathbf{f}^t(\mathbf{x}_1))$ denotes a variational term. In the case of a discrete map $\mathbf{f}$, the variational term $\mathbf{V}_x(\mathbf{f}^t(\mathbf{x}_1))$ takes the form of the product of the instantaneous Jacobians $\mathbf{J}_x$ of $\mathbf{f}$ over the course of the system trajectory (Eckmann & Ruelle, 1985; Sandri, 1996) and can be written as

$$\mathbf{V}_{\mathbf{x}}(\mathbf{f}^t(\mathbf{x}_1)) = \mathbf{J}_{\mathbf{x}}(\mathbf{f}^t(\mathbf{x}_t))\mathbf{J}_{\mathbf{x}}(\mathbf{f}^{t-1}(\mathbf{x}_{t-1}))\ldots\mathbf{J}_{\mathbf{x}}(\mathbf{f}(\mathbf{x}_1)). \tag{3}$$

Finally, according to Oseledets' theorem (Oseledets, 1968), the Lyapunov exponents for the autonomous system (1) are given by the eigenvalues of the matrix

$$\mathbf{M} = \lim_{t\to\infty} \frac{1}{2t} \log\left(\mathbf{V}_x(\mathbf{f}^t(\mathbf{x}_1))\mathbf{V}_x^T(\mathbf{f}^t(\mathbf{x}_1))\right), \tag{4}$$

where $\cdot^T$ indicates the matrix transpose.

Essentially, the Jacobians $\mathbf{J}_x(\mathbf{f}^k)$ define the space of instantaneous local volume transformations that are accessible to the system. The variational term $\mathbf{V}_x(\mathbf{f}^t)$ determines the memory timescales of network dynamics, and every Lyapunov exponent defines the direction with an asymptotically stable rate of memory change. The number of Lyapunov exponents is equal to the dimensionality of the system N and the largest Lyapunov exponent determines the stability of network dynamics, with a negative (positive) Lyapunov exponent indicating its stability (instability). In particular, the largest Lyapunov exponent changes its sign when the system undergoes a transition between chaos and order.

In the general case given in (1), the external input makes the system non-autonomous and the existence of the limit (4) is not guaranteed. Thus, the described analysis cannot be easily performed for most input-driven recurrent networks. In this study, we show how the introduction of weak coupling can mitigate this issue; see Section 3.

2.2. Learning dynamics

Through the lens of the theory of dynamical systems, training an RNN of the form (1) with backpropagation through time creates the following learning dynamics $\mathbf{g}$ on the recurrent weights $\mathbf{w}$,

$$\mathbf{w}_{\tau+1} = \mathbf{g}(\mathbf{w}_\tau) = \mathbf{w}_\tau - \eta\frac{1}{M}\sum_M \nabla_w L, \tag{5}$$

where $\tau \in \mathbb{N}$ denotes the training iteration, $\eta \in \mathbb{R}^+$ is the learning rate hyperparameter, $\nabla_w L$ is the gradient of the loss function L with respect to the weight w, and $M \in \mathbb{N}$ is the batch size. Here, we assume a deterministic version of gradient descent without loss of generality. In practice, training an RNN with BPTT involves unrolling the network over time so that the RNN is transformed into an equivalent feed forward network consisting of D *unrolled recurrent layers* (one for each time point $1 \le t \le D$), and also defines a temporal distance between any two unrolled recurrent layers. To obtain network predictions, all networks are equipped with an affine readout layer that transforms the state vector of the network at the last time point $t = D$ into activations

of a set of output units on which the loss function L is computed (see Fig. 1).

The theoretical considerations derived in this section can be applied to all typical loss functions used for classification and regression problems, as long as the Hessian of the loss function with respect to network predictions is positive semidefinite and its derivative at the optimum has a vanishing mean (Schraudolph, 2002). In our experiments, we used loss functions for which these conditions are met, namely a cross-entropy loss with softmax for classification problems and a root-mean-square (RMS) loss for regression tasks. For a detailed description of the experiments performed, see Section 4.

The instantaneous Jacobian $\mathbf{J}_w$ of the learning dynamics (5) is then given by

$$\mathbf{J}_w(\mathbf{g}) = \mathbf{I} - \eta \frac{1}{M} \sum_M \mathbf{H}_w(L), \tag{6}$$

where $\mathbf{H}_w(L)$ denotes the Hessian of the loss function L and $\mathbf{I}$ is the identity matrix. Note that the asymptotic behavior of the product of instantaneous Jacobians given in (6) determines the convergence or divergence of the learning dynamics, in the same way as the variational term $\mathbf{V}_x$ determines the convergence of divergence of the recurrent network dynamics in (1). This agrees with previous studies that have shown that the curvature of the loss landscape defined by the Hessian plays a crucial role in gradient-based learning (Dauphin et al., 2014).

From (6), it also follows that every eigenvalue of the Hessian $\mathbf{H}_w(L)$ defines an *effective learning rate* in the direction of the associated eigenvector in the weight space, modulating the base learning rate η in the direction of this eigenvector. This effective learning rate is equal to $(1-\eta\lambda_{\mathbf{H}}^M)^{-1}$, where $\lambda_{\mathbf{H}}^M$ is the corresponding eigenvalue of the Hessian averaged over a given batch. In order to further investigate these effective learning rates, we can rewrite the Hessian as

$$\mathbf{H}_w(L) = \mathbf{J}_w^T(\mathbf{f}^D(\mathbf{x}_1))\mathbf{H}_{\mathbf{f}^D}(L)\mathbf{J}_w(\mathbf{f}^D(\mathbf{x}_1)) + \sum_{n=1}^{N} \nabla_{\mathbf{f}_n^D}(L)\mathbf{H}_w(\mathbf{f}_n^D(\mathbf{x}_1)), \tag{7}$$

where D denotes the input length, $\mathbf{J}_w(\mathbf{f}^D)$ is the Jacobian of $\mathbf{f}^D$ with respect to w, $\nabla_{\mathbf{f}_n^D}(L)$ is the gradient of the loss function L with respect to the nth coordinate of $\mathbf{f}^D$, and $\mathbf{H}_w(\mathbf{f}^D)_n$ is the Hessian of the nth coordinate of $\mathbf{f}^D$ with respect to w, see Schraudolph (2002). Here, we consider the loss to be a function of the network state at the last time point $L(\mathbf{f}^D)$. As therefore the decoding layer is included in $\mathbf{H}_{\mathbf{f}^D}(L)$, this allows us to directly show the dependence of $\mathbf{H}_w(L)$ on $\mathbf{J}_w(\mathbf{f}^D)$.

The first term in (7) is known as the Generalized Gauss–Newton (GGN) matrix, a popular approximation for the curvature matrix in second-order optimization methods (Thomas et al., 2020). For the loss functions considered here, the Hessian converges to the GGN matrix when training by BPTT because the second term in (7) is proportional to the loss and vanishes as the learning approaches a local minimum of the loss landscape. Therefore, the asymptotic behavior of learning dynamics given in (5) is predominately influenced by the properties of the GGN matrix.

For a recurrent network defined by (1), the GGN matrix depends on the Jacobians $\mathbf{J}_w(\mathbf{f}^t(\mathbf{x}_t)))$ and can be computed by applying the chain rule as

$$\mathbf{J}_w(\mathbf{f}^D(\mathbf{x}_1)) = \sum_{1\le t\le D} \mathbf{V}_x^{D-t}(\mathbf{f}^D(\mathbf{x}_{t+1}))\mathbf{J}_w(\mathbf{f}^t(\mathbf{x}_t)), \tag{8}$$

where $\mathbf{V}_x^{D-t}(\mathbf{f}^D(x_{t+1})) = \mathbf{J}_x(f^D(\mathbf{x}_D))\mathbf{J}_x(\mathbf{f}^{D-1}(\mathbf{x}_{D-1}))\dots\mathbf{J}_x(\mathbf{f}^{t+1}(\mathbf{x}_{t+1}))$ is a truncated version of the variational term in (3), and $\mathbf{V}_x^0 = I$. This shows that the memory properties of network dynamics influence the GGN matrix and thereby the final configuration to which the network converges. Taken together, RNN dynamics thus plays the role of an inductive bias.

Importantly, the second term in (7) can be analyzed further if we apply chain rule to the Hessian t times, and we obtain

$$\sum_{n=1}^{N} \nabla_{\mathbf{f}_n^D}(L)\mathbf{H}_w(\mathbf{f}_n^D(\mathbf{x}_1)) = \sum_{n=1}^{N}\sum_{t=2}^{D} \nabla_{\mathbf{f}_n^t}(L)\mathbf{C}^{\mathbf{f}_n^t} + \sum_{n=1}^{N} \nabla_{\mathbf{f}_n}(L)\mathbf{H}_w(\mathbf{f}_n(\mathbf{x}_1)), \tag{9}$$

where we denote the curvature matrix at time t as $\mathbf{C}^{\mathbf{f}_n^t} = (\mathbf{J}_w(\mathbf{f}^{t-1}))^T\mathbf{H}_{\mathbf{f}^{t-1}}(\mathbf{f}_n^t)(\mathbf{J}_w(\mathbf{f}^{t-1}))$ with total Jacobians evaluated at $\mathbf{x}_1$, and by $\mathbf{H}_{\mathbf{f}^{t-1}}(\mathbf{f}_n^t)$ the Hessian of nth coordinate of $\mathbf{f}^t$ with respect to the previous state $\mathbf{f}^{t-1}$, evaluated as instantaneous partial derivatives.

Due to the presence of an activation function, the curvature matrix $\mathbf{C}^{\mathbf{f}_n^t}$ is (in contrast to the GGN matrix) not necessarily positive semi-definite, but we note that this does not affect the following conclusions. In particular, the application of the chain rule in (9) shows how the Hessian (left-hand side) can be split into terms corresponding to different unrolled recurrent layers (right-hand side). Thus, we can represent the eigenvalues of the Hessian of the full network as a sum of the contributions of each unrolled recurrent layer as $\lambda_{\mathbf{H}} = \sum_{1\le t\le D} \lambda_{\mathbf{H}}^{\mathbf{f}^t}$. Importantly, the magnitudes of these contributions are proportional to the corresponding gradient propagation

$$\lambda_{\mathbf{H}}^{\mathbf{f}^t} \propto \nabla_{\mathbf{f}^t}(L) = (\mathbf{V}_x^{D-t})^T \nabla_{\mathbf{f}^D} L, \tag{10}$$

meaning that the variational term $\mathbf{V}_x$ determines the contribution of each unrolled layer to the overall effective learning rates.

Taken together, this shows that the variational term $\mathbf{V}_x$ not only defines the memory properties of the network dynamics but also temporally modulates the effective learning rates, establishing a connection between fading memory and learning dynamics. This connection will be important for the further analysis of weakly coupled residual recurrent networks (WCRNNs) as defined below.

3. Weakly coupled residual recurrent networks

Residual connections in deep feed-forward networks have been shown to allow for a better backpropagation of errors and are usually implemented by an identity map between subsequent layers (He et al., 2016).

Here, we consider the more general case of *weakly coupled residual recurrent neural networks* (WCRNNs) equipped with an arbitrary fixed residual map $\mathbf{R} : \mathbb{R}^N \to \mathbb{R}^N$. The update equation for such networks takes the form

$$\mathbf{x}_{t+1} = \mathbf{R}(\mathbf{x}_t) + \gamma \cdot \sigma(\mathbf{W}_{xx}\mathbf{x}_t + \mathbf{S}_t), \tag{11}$$

where $\gamma \ll 1$ denotes a weak coupling constant, σ denotes a non-linearity (typically $\tanh$), and $\mathbf{S}_t = \mathbf{W}_{sx}\mathbf{s}_t$ is the input vector that is an affine projection with weights $\mathbf{W}_{sx}$ on some time-varying input data $\mathbf{s}_t$. We also equip the network with an affine readout layer $\mathbf{W}_{xo}$ and perform the readout on the final network state $\mathbf{x}_D$, where D is the length of the input, see Fig. 1. The weights $\mathbf{W}_{sx}, \mathbf{W}_{xx}, \mathbf{W}_{xo}$ are subject to backpropagation learning and include trainable bias terms that are omitted in (11) for simplicity of notation.

The condition of weak coupling ($\gamma \ll 1$) simplifies the analysis of memory properties of such networks because for small values of γ the variational term (3) can be written as

$$\mathbf{V}_x(\mathbf{f}^t(\mathbf{x}_1)) = \mathbf{J}_R(\mathbf{x}_t)\mathbf{J}_R(\mathbf{x}_{t-1})\dots\mathbf{J}_R(\mathbf{x}_1) + O(\gamma), \tag{12}$$

where $\mathbf{J}_R(\mathbf{x}_t)$ denotes the instantaneous Jacobian of the residual at time t, and $O(\cdot)$ is the Landau O.

Eq. (12) shows that the properties of the dynamics of WCRNNs are predominately determined by the instantaneous Jacobians of the residual. Thus, the WCRNN architecture allows for a control of the properties of fading memory of the entire network by choosing an appropriate residual.

We note that the weak coupling condition holds as long as γ is sufficiently smaller than 1. Small values of γ lead to controllable network dynamics in WCRNNs and are the prerequisite for well-defined Lyapunov exponents, as the analysis below shows. However, if γ becomes negligibly small ($\gamma < 10^{-3}$), the external input will be too small in magnitude to allow the networks to perform well, so there is a trade-off between stabilizing the system dynamics and still allowing for a forcing of system dynamics through an external input. In practice, we have seen that values of $\gamma \in [0.001, 0.1]$ work well for the datasets tested.

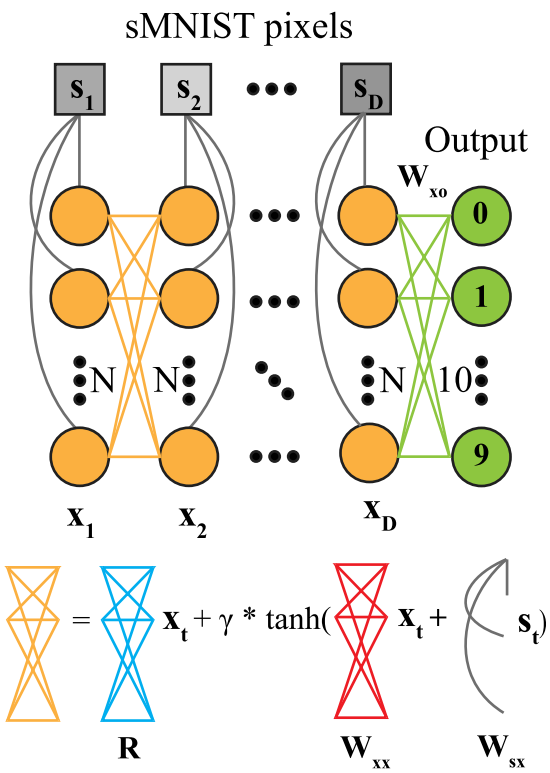

Fig. 1. Schematic representation of the WCRNN model for the sMNIST classification task. Each 28 × 28 pixel MNIST digit is serialized and presented to the network as a time series of length $D = 784$. $\mathbf{s}_t$ and $\mathbf{x}_t$ denote the stimulus and network amplitude configurations at the discrete time step t ($1 \leq t \leq 784$), respectively. Orange circles indicate RNN nodes (unrolled over time) and green circles indicate the output units for the 10 digit classes, respectively. Line colors indicate the input type. The total recurrent input is shown in orange and consists of a residual input as mediated by the residual map R shown in blue, a recurrent input as mediated by the recurrent weight matrix $\mathbf{W}_{xx}$ shown in red, and an external input mediated by an input projection matrix $\mathbf{W}_{sx}$ shown in gray. Note that, in general, the total recurrent input at time t shown in orange does not represent an affine transformation of the previous network state $\mathbf{x}_{t-1}$. The readout weights $\mathbf{W}_{xo}$ of a linear readout performed at $t = 784$ are shown in green. In the case of the ADD datasets, the configuration is analogous, except for adjustments in the input and output layers (2d input, one output unit).

In the general case (11), the Lyapunov exponents depend on x_t due to the non-linearity in the residual, which can complicate proving the existence of the limit in (4). To simplify the analysis, we first consider networks with linear residuals $\mathbf{R}(\mathbf{x}_t) = \mathbf{R}\mathbf{x}_t$, where $\mathbf{R}$ is a $N \times N$ matrix. For the linear case, the instantaneous eigenvalues of the residual are independent of the trajectory of the system, and the variational term simplifies to $\mathbf{V}(\mathbf{f}^t(\mathbf{x}_1)) = \mathbf{R}^t + \mathcal{O}(\gamma)$, where $\cdot^t$ denotes matrix exponentiation. Furthermore, we can easily derive the Lyapunov exponents in this case, because the Oseledets equation (4) in the limit of weak coupling yields $\mathbf{M} = \log \mathbf{A} + \mathcal{O}(\gamma)$, where $\mathbf{A} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ is the diagonal matrix of eigenvalues of $\mathbf{R}$. This means that for WCRNNs the logarithms of the eigenvalues of the residuals $\lambda_{\text{residual}}$ approximate the Lyapunov exponents LE_{net} of the entire network

$$\mathrm{LE}_{\text{net}} \approx \log \lambda_{\text{residual}}, \tag{13}$$

and that the weak coupling limits the range of their finite-size fluctuations. Based on their dynamical stability (distance to the edge of chaos, see Bertschinger & Natschläger, 2004) as determined by the magnitude of the largest eigenvalue $\lambda_{\max}$ of the residual matrix, we can distinguish three classes of WCRNNs: (i) subcritical ($\lambda_{\max} < 1$), (ii) critical ($\lambda_{\max} = 1$), and (iii) supercritical ($\lambda_{\max} > 1$) networks.

Moreover, it follows from (7) that the Hessian of WCRNN dynamics at the point of convergence takes the form

$$\mathbf{H}_w(L) = \sum_{1 \leq k \leq D} \sum_{1 \leq m \leq D} \mathbf{J}_w^T(\mathbf{f}^k(\mathbf{x}_k))(\mathbf{R}^{D-k})^T \mathbf{H}_{\mathbf{f}^D}(L) \mathbf{R}^{D-m} \mathbf{J}_w(\mathbf{f}^m(\mathbf{x}_m)) + \mathcal{O}(\gamma^3). \tag{14}$$

This shows that the contribution of the partial derivatives to the overall curvature at different times is proportional to the corresponding eigenvalues of the residual matrix. This means that the residual matrices determine the inductive biases in the final weight configuration to which the network converges. We thus predict that WCRNNs will achieve different levels of performance depending on the residual initialization, and that the residuals resulting in the best performing networks will be dataset-specific. Moreover, we predict that an optimal residual configuration will depend on the input length.

Similarly to the GGN matrix, the second term of (8) is also affected by the residuals. It follows from (10) that

$$\lambda_{\mathbf{H}}^{\mathbf{f}^t} \propto \nabla_{\mathbf{f}^t}(L) = (\mathbf{R}^{D-t})^T \nabla_{\mathbf{f}^t} L + \mathcal{O}(\gamma), \tag{15}$$

meaning that the magnitudes of eigenvalues of $\mathbf{R}$ determine the contribution of each unrolled recurrent layer to the overall effective learning rates. Based on the propagation of the gradients as described in (15), we therefore predict temporally modulated effective learning rates, where the eigenvalues of $\mathbf{R}$ define the exponential rate with which this contribution decays or increases with time t. On the one hand, an increase in the magnitudes of the eigenvalues of $\mathbf{R}$ induces an improved learning efficiency for information contained in temporally distant recurrent layers. However, such an increase can also result in an instability of learning dynamics if the overall effective learning rates reach high magnitudes as a result. In contrast, a decrease in the magnitudes of the eigenvalues of $\mathbf{R}$ can result in a loss of temporally distant information and reduce the efficiency of learning dynamics. At the same time, the same mechanism can also provide better stability of learning dynamics by reducing the overall magnitudes of effective learning rates.

These properties of WCRNNs make us hypothesize that there exists a trade-off between the stability of learning dynamics and *learning efficiency*, the number of iterations required to achieve a certain value of the loss function (with lower numbers of iterations being better). For more subcritical networks, we expect problems with slow learning and vanishing gradients. For more supercritical networks, we expect faster convergence, but potentially problems with unstable learning trajectories and exploding gradients at the same time. For critical WCRNNs, characterized by network dynamics in proximity to the edge of chaos, we anticipate an optimal balance between learning effiency and stability.

In summary, we design weakly coupled residual recurrent networks (WCRNNs), where the introduction of weak coupling allows us to obtain stable memory timescales as defined by Lyapunov exponents. We show that the properties of fading memory are mainly determined by properties of the residuals and, in the case of linear residuals, by the eigenvalues of the residual matrix. On the basis of our analyses, we predict that WCRNNs close to criticality show the best trade-off between efficiency and stability of learning, and that optimal residual configurations depend on the input length. In conclusion, we have shown here how architecture-based inductive biases in the form of residuals shape the memory properties and learning dynamics of WCRNNs in theory. In the next section, we will test our predictions in practice.

4. Experiments

To empirically validate our theoretical predictions, we performed experiments on several datasets:

- Sequential MNIST (sMNIST), where the 28×28 pixels of MNIST digits (LeCun, Bottou, Bengio, & Haffner, 1998) are presented to the network sequentially in the form of a time series and the task is to solve a digit classification problem. The samples are turned into a time series of length 784 by collecting intensity values in scan-line order from top left to bottom right. We also consider the permuted sequential MNIST (psMNIST) dataset, where a random but fixed permutation is applied to the sMNIST samples. The permutation removes the dominant low-frequency components present in the sMNIST samples and increases the difficulty of the classification problem. For these classification tasks, a cross-entropy loss was used.
- The adding problem (ADD) of lengths 100, 200, 400, 800, as a regression problem. Here, every sample is given by a two-dimensional time series of the specified length (Hochreiter & Schmidhuber, 1997). The first coordinate is given by random numbers drawn from a uniform distribution $[0, 1]$, and the second coordinate constitutes a cue signal taking values 0 (no cue) and 1 (cue). The task of the network is to compute the sum of the input values presented in the first coordinate for two cue points randomly placed in the first half and the second half of the signal, respectively. For these tasks, a root mean square (RMS) loss was used.

As the primary goal of this study is not to provide a model outperforming other SOTA architectures, but rather to study network dynamics in WCRNNs and underlying principles of how network dynamics influence learning dynamics and inductive biases, we chose the MNIST dataset and the adding problem for most of our experiments as these are well established classic benchmarks for RNNs. These datasets can be made more challenging by introducing permutations for MNIST or longer sample lengths for the adding problem. To test our models on a more challenging task, we also performed experiments on the gray-scaled sCIFAR10 dataset, which is part of Long Range Arena benchmark designed for long sequence tasks (Tay et al., 2020). The sCIFAR10 dataset contains 10 different classes of 32×32 pixel images, which are transformed into time series of 1024 gray-scaled pixels, analogously to the sMNIST dataset. As for the sMNIST dataset, a cross-entropy loss function was used for the latter.

The experiments were carried out in PyTorch (Paszke et al., 2019) for network sizes of 50, 100, and 200 units. Training was performed for 200 epochs for the sMNIST and psMNIST datasets and for 150 epochs for the ADD datasets. Stochastic gradient descent (SGD) with a momentum of 0.9 was used as an optimizer for BPTT and training iterations were performed according to a minibatch scheme, using batch sizes of 64, 128, and 256 samples. Qualitatively, results were found to be mostly independent of network and batch size. Thus, we present results for networks of 100 units, trained with a batch size of 128 samples in the following. Results were collected over 5 network instances with random weight initialization, and most plots report mean scores and their standard deviation obtained from these 5 instances. During initialization, weights and biases were randomly sampled from a Kaiming uniform distribution according to the default implementation of the PyTorch `torch.nn.Linear` layer ($U(-1/\sqrt{n_{\text{in}}}, 1/\sqrt{n_{\text{in}}})$, where n_{in} denotes the input dimension of a given layer). As the non-linear activation function, we used $\sigma = \tanh$ in all of our experiments.

In Section 4.1 we present simulation results of WCRNNs with dynamics close to the edge of chaos, showing the validity of our theoretical predictions about their performance for all datasets. In Sections 4.2 and 4.3 we show how rotational and heterogeneous residuals can be beneficial to the performance of WCRNNs. Lastly, in Section 4.4 we show that our results can also be generalized to Elman RNNs by an initialization scheme. The results presented were found to be consistent across all datasets, inputs, and networks with different random weight initializations.

4.1. Critical residuals

First, we studied different WCRNNs with network dynamics in proximity of the edge of chaos. To place networks in this dynamical regime, we introduced linear diagonal residuals of type $\mathbf{R} = r\mathbf{I}$, where the *residual connection strength* r is a scalar hyperparameter, and $\mathbf{I}$ denotes the identity matrix. According to (13), these networks have N Lyapunov exponents with identical values equal to $\log r$, where N is the number of units in the network. By varying r, we can control the Lyapunov exponents, and thereby the distance of the network dynamics to the edge of chaos. We varied the value of r in the interval $r \in [0.91, 1.02]$ to explore the range of subcritical, critical, and supercritical dynamics (see Fig. 2). To numerically compute Lyapunov exponents from network dynamics, we used a method based on QR-decomposition (Sandri, 1996) that ensures the numerical convergence of the eigenvalues of the orthonormalized variational term even for unstable dynamics (see Appendix A). Gradient propagation was measured by the L^∞ gradient norms $\frac{\partial L}{\partial \mathbf{f}^t}$ in the test dataset (Arjovsky, Shah, & Bengio, 2016). All Hessians were computed for a randomly chosen but fixed set of 1000 samples from the test set and we note that the results were found to be consistent across different choices for this set (data not shown). The findings were found to be qualitatively consistent in the range $\gamma \in [0.001, 0.1]$ and here we present results for the value $\gamma = 0.01$. We found that WCRNNs with $\gamma < 0.001$ have difficulties in achieving high task performance as in this case the forcing of the network dynamics by the input becomes too small. If $\gamma > 0.1$, the forcing of the network dynamics can become too strong and tends to lead to unstable dynamics, which is in agreement with the conditions discussed in Section 3. In particular, the coupling constant γ had an effect on learning stability and efficiency, which is similar to the global learning rate, as $\nabla_w L = \mathcal{O}(\gamma)$; see Fig. B.2.

As expected, we found that the eigenvalues of the variational term had converged to the values defined by the residual according to Eq. (13), which confirms the existence of Lyapunov exponents for the WCRNNs (Fig. 2B). The learning curves for all networks are shown in Fig. B.3. Furthermore, we observed that the magnitudes of the eigenvalues of the Hessians increased with an increase in r, see Fig. 2C. This supports the theoretical results on how the proximity of network dynamics of WCRNNs to the edge of chaos affects their effective learning rates, see (15). In addition, we observed that in subcritical networks the magnitudes of eigenvalues of the Hessian tended to increase over training, while in critical and supercritical networks they tended to decrease; see Fig. B.3. The gradient norms $\frac{\partial L}{\partial f^t}$ of the subcritical and supercritical networks were observed to decrease or increase exponentially with a constant rate over time, as predicted by (15), see Fig. 2D. We compare the eigenvalues of the variational term, the eigenvalues of the Hessian, and the gradient propagation before training WCRNNs, because the initial differences between the networks are attributed to the differences in their residuals. In contrast to the eigenvalues of the Hessian and gradient norms, the eigenvalues of variational term remained in the same range during the training period by design, see Fig. B.1.

To evaluate the networks' practical expressivity, we measure not only the best accuracy achieved on a test set during the training period (overall performance), but also the learning efficiency. We assess the learning efficiency by the number of training iterations required for the network to reach a given threshold of minimal performance (MP), chosen differently for each dataset and thus small values of this measure correspond to better efficiency. The threshold of minimal performance was set to 50% accuracy for the sMNIST and psMNIST datasets and to a RMS of 0.05 for the ADD datasets. These threshold values were chosen to capture a non-negligible deviation from chance-level performance. For networks that were unable to reach this threshold, the number of iterations to reach minimal performance was set to the maximum number of iterations in the training period.

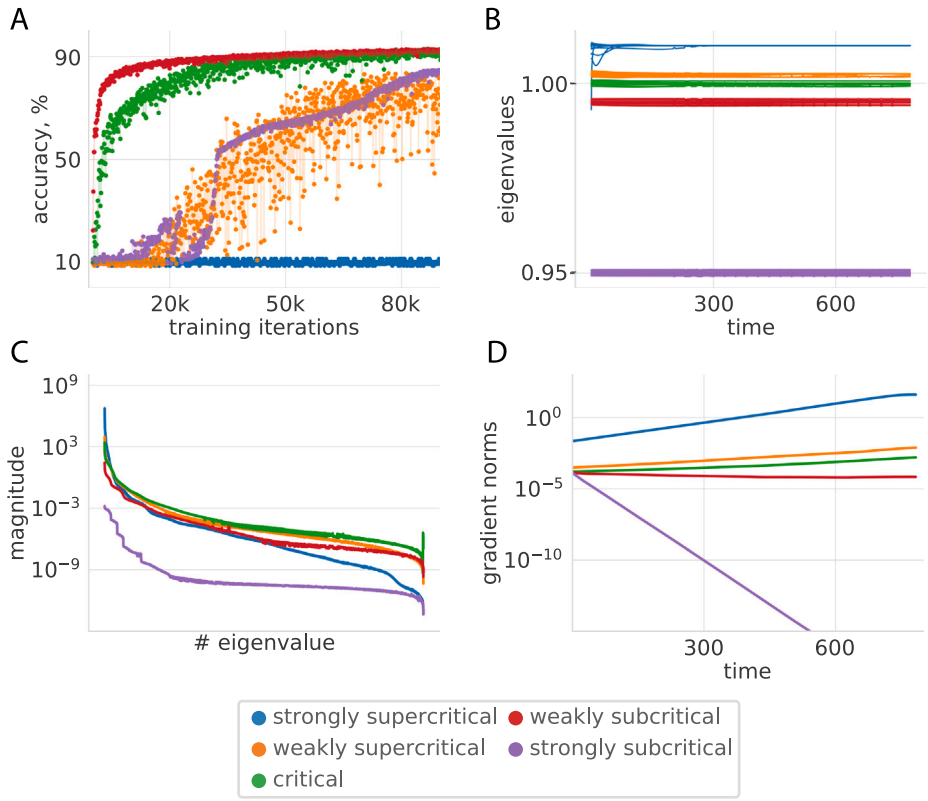

Fig. 2. WCRNN performance and dynamics on sMNIST. Colors indicate network type; strongly subcritical ($r = 0.95$), weakly subcritical ($r = 0.995$), critical ($r = 1$), weakly supercritical ($r = 1.0025$), strongly supercritical ($r = 1.01$). **A.** Test accuracy on sMNIST as a function of training iterations over 200 training epochs. **B.** Dynamics of eigenvalues of variational term $\mathbf{V}_x(f^t)$ before training. Lines show trajectories of 20 randomly chosen eigenvalues over time for a randomly chosen input digit. **C.** Rank plot of the eigenvalue magnitudes of the Hessian of the loss function $\mathbf{H}_w(L)$ before training. Lines show eigenvalues that were computed for a randomly chosen batch of the sMNIST test set. **D.** Evolution of norms of BPTT gradients as a function of time. Lines show gradient norms that were computed over a random input batch before training.

The practical expressivity of WCRNNs was found to be strongly dependent on the parameter r, showing that although residual connections do not limit the theoretical expressivity of the network, they play an important role for network expressivity in practice.

We observed that an increase in distance from the edge of chaos in the subcritical regime resulted in a decrease in learning efficiency for all datasets, see Fig. 3. This is in good agreement with our theoretical predictions presented in Section 3. The observed decay in learning efficiency is caused by vanishing gradients, showing that the EVGP poses an important practical limitation for subcritical networks. We observed that supercritical networks showed unstable learning trajectories, see Fig. 3A, again in line with our theoretical predictions. The fact that the gradients of supercritical networks were informative but exploded in magnitude was supported by the finding that clipping of the gradients resulted in more stable learning trajectories (data not shown). Overall, these results support our hypothesis that proximity to the edge of chaos enables a better learning efficiency at the expense of the stability of learning dynamics.

Interestingly, we observed that supercritical networks showed better performance for the sMNIST and psMNIST datasets, while subcritical networks performed better for the ADD datasets. This can be explained by the fact that the inputs of ADD datasets are dense in the first dimension (input values in the first dimension are rarely zero), whereas the inputs of the sMNIST datasets are more sparse (many input values are zero or close to it). These different characteristics of the input favor exploding and vanishing gradients, respectively. Importantly, we observed that the best performing networks were closer to the edge of chaos for longer ADD datasets. This agrees with the general intuition that longer inputs require longer memory timescales and shows that the best inductive biases are dependent on the characteristics of the inputs as defined by the dataset. We saw that networks with dynamics close to the edge of chaos showed both higher overall performance and better learning efficiency compared to strongly supercritical and strongly subcritical networks. Overall, weakly subcritical networks performed best with a dataset-specific optimal distance to the edge of chaos. We note that our results are in good agreement with previous literature on the role of the interplay between architecture-based inductive biases and characteristic properties of the input (Goyal & Bengio, 2022; Kerg et al., 2022; Liu et al., 2023; Mastrogiuseppe & Ostojic, 2018; Rajan, Abbott, & Sompolinsky, 2010).

In summary, we evaluated the practical expressivity of subcritical, critical, and supercritical WCRNNs by means of their overall performance and learning efficiency on a set of benchmark tasks. We validated the existence of Lyapunov exponents by numerical calculations and confirmed our theoretical predictions about the trade-off between learning efficiency and the stability of learning dynamics. Consistent with the previous literature (Schoenholz et al., 2016), we found that residual networks with dynamics close to the edge of chaos possess a higher practical expressivity compared to strongly subcritical or supercritical networks. Importantly, we observed that the weakly subcritical networks showed the best overall performance, and found that the optimal distance to the edge of chaos was indicative of memory timescales beneficial for the task at hand.

4.2. Rotational residuals

The WCRNNs considered so far only had residual matrices with real eigenvalues, so they were limited to scaling linear transformations and reflections. To study all geometrical transformations represented by the group of square matrices, we have to introduce rotations, which correspond to matrices with eigenvalues having non-vanishing imaginary

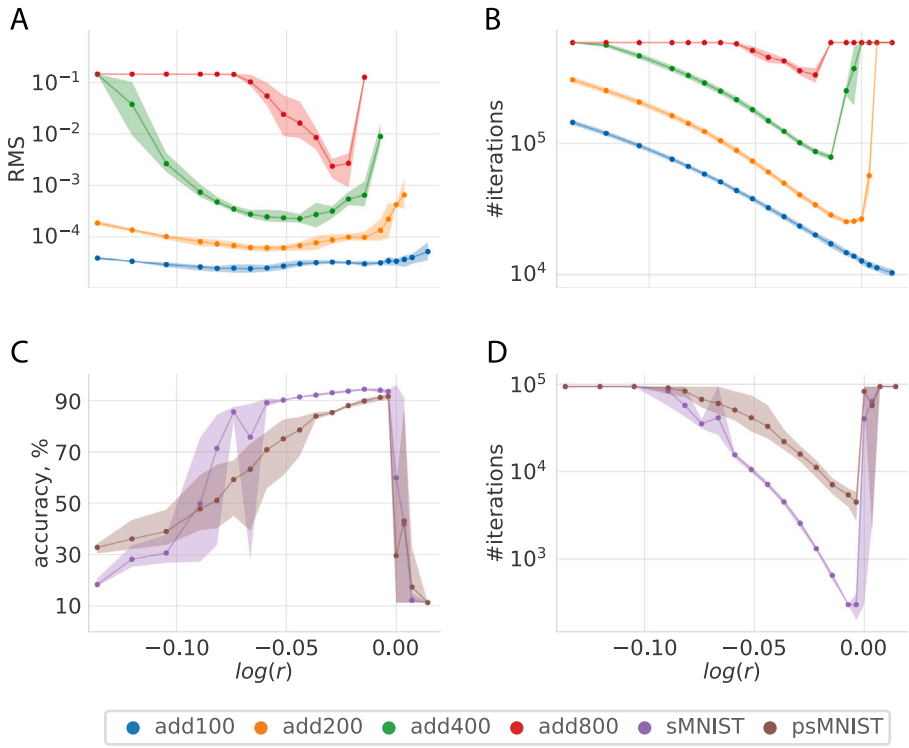

Fig. 3. Practical expressivity of WCRNN networks as a function of the value of the residual connection strength r for the ADD and MNIST datasets. Lyapunov exponents of presented WCRNNs are equal to $\log r$. All networks have a value of $\gamma = 0.01$. Lines show mean values over 5 network instances with random weight initialization, shaded areas show the range between minimal and maximal values. **A.** Best test accuracy on the ADD task as measured by root mean squared error (RMS) attained over 150 training epochs for the ADD datasets. **B.** The number of training iterations to reach a defined minimal performance (MP) of 0.05 RMS error (see main text) for ADD datasets. **C.** Best test accuracy for MNIST dataset over 200 training epochs. **D.** The number of training iterations to reach a MP of 50% test accuracy for MNIST datasets.

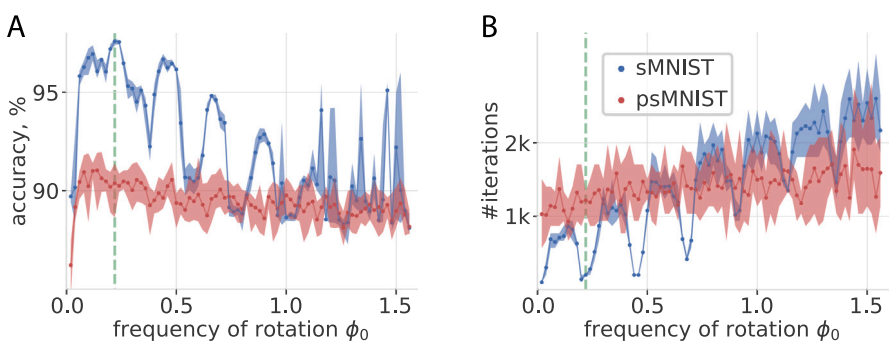

Fig. 4. Performance of WCRNNs with homogeneous rotational residuals on sMNIST and psMNIST. Lines show mean values over 5 network instances with random weight initialization, shaded areas show the range between maximal and minimal values. Colors indicate the training dataset. The green dashed line indicates the characteristic frequency of sMNIST $\phi_c = 2\pi/28 \approx 0.22$. **A.** The best test accuracy attained as a function of angular frequency of rotation ϕ_0 of the homogeneous residual matrix. **B.** The number of training iterations to reach a defined minimal performance (MP) of 50% test accuracy (see main text), as a function of angular frequency of rotation ϕ_0.

part. For this purpose, we consider residual matrices $\mathbf{R}$ taking the form of orthonormal diagonal block matrices

$$\mathbf{R} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{T}_{N/2} \end{bmatrix}, \quad \text{with} \quad \mathbf{T}_i = \begin{bmatrix} \cos\phi_i & -\sin\phi_i \\ \sin\phi_i & \cos\phi_i \end{bmatrix}, \tag{16}$$

where N is the number of units in the network and $\phi_i \in [0, 2\pi]$ denotes an angular frequency of rotation. This type of residual represents rotation matrices with arbitrary combinations of angular frequencies ϕ_i.

First, we studied WCRNNs with rotational residuals of the form (16) for which all ϕ_i have a constant value ϕ_0, referred to as *homogeneous rotational residuals* from now on. We trained WCRNNs with homogeneous rotational residuals for different values of ϕ_0 on sMNIST and psMNIST for 50 epochs, see Fig. 4. Here, we chose a shorter training period to better demonstrate the differences in performance between such networks that arise due to differences in their learning efficiency, but we also trained these networks for 200 epochs, see supplementary Fig. B.4 where the results were similar. We found that these networks converged fastest and expressed highest performance on sMNIST when their angular frequency ϕ_0 was close to $2\pi/28 \approx 0.22$. Interestingly, this frequency coincides with the maximal peak of the average power spectra of the samples in sMNIST, thus we call it the *characteristic frequency* ϕ_c of the dataset (Effenberger, Carvalho, Dubinin, & Singer, 2023). We found that homogeneous rotational residuals with angular frequencies that were close to integer multiplies of ϕ_c (i.e., ϕ_c and its harmonics) result in networks that show the highest performance and the best learning efficiency, see Fig. 4. As follows from Section 2.2, we attribute this enhanced performance to the fact that

the instantaneous Jacobians of the networks with rotational residuals modulate the effective learning rates in an oscillatory manner. Such instantaneous Jacobians allow the networks to accumulate derivatives that are in-phase and cancel out derivatives that are out of phase, see Fig. B.4B. Taken together, this shows that oscillatory learning rates can be beneficial for a network's practical expressivity when they align with characteristic spectral properties of the dataset.

In contrast, we did not find a strong effect on practical expressivity when varying the angular frequencies of the rotational residuals for the psMNIST, see Fig. 4. This can be explained by the fact that, in contrast to the sMNIST dataset, the samples of the psMNIST dataset do not possess a prominent characteristic frequency that can be exploited for classification.

Furthermore, we also studied the case where residuals are given by random orthonormal matrices with a uniform distribution of angular frequencies of rotations (constructed with the function `scipy.stats.ortho_group.rvs` from the SciPy package (Virtanen et al., 2020)). The latter were found to result in networks with a performance similar to that of those with residuals with heterogeneous angular frequencies discussed below, see Fig. B.4D.

Overall, we found that choosing residuals informed by characteristic spectral properties of the samples in the dataset can result in WCRNNs with higher practical expressivity. In particular, we show for sMNIST that rotational residuals can significantly improve the efficiency of learning. This agrees with the previous findings of increased performance of networks composed of oscillatory units (Effenberger et al., 2023; Norcliffe, Bodnar, Day, Simidjievski, & Liò, 2020; Rusch & Mishra, 2020).

4.3. Heterogeneous residuals

In this section, we study how introducing heterogeneous residuals influences practical expressivity in WCRNNs. We first considered networks for which the residual matrix takes the form

$$\mathbf{R} = \mathrm{diag}(\mathbf{r}), \tag{17}$$

with each coordinate r_i of $\mathbf{r}$ sampled from a uniform distribution $\mathrm{U}(r_0 - \delta r/2, r_0 + \delta r/2)$. For our experiments, we fix r_0 which defines a baseline distance to edge of chaos and gradually increase the level of heterogeneity controlled by the term δr. We found that subcritical networks $r_0 < 1$ with moderate heterogeneity ($\delta r/2$ is less than the distance to the edge of chaos) showed better performance and efficiency of learning, and only strong heterogeneity ($\delta r/2$ is greater than the distance to the edge of chaos) yield WCRNNs with unstable learning dynamics, see Fig. 5A. This can be explained by the fact that the heterogeneity of the residual matrix increased the diversity of memory timescales in the Lyapunov spectrum and brought the network dynamics closer to the edge of chaos, resulting in longer memory timescales, see Fig. 5B. Moreover, the heterogeneity in subcritical networks improved gradient propagation and increased the number of non-vanishing eigenvalues of the Hessian, making learning dynamics richer, see Fig. 5C and D. In contrast, for networks with dynamics close to the edge of chaos, the benefit of having heterogeneous residuals reduced, because heterogeneity increased the risk of exploding gradients (data not shown).

Next, we considered WCRNNs with residual matrices that are a product of an orthonormal matrix of the form (16) and a diagonal matrix of the form (17). Furthermore, we allowed unit-specific heterogeneity of the weak coupling parameter by substituting the scalar coupling constant from (11) by a vector $\boldsymbol{\gamma}$. Informed by previous experiments, we sampled $\mathbf{r}_i$ from $U([0.99, 1])$, ϕ_i from $U([0, \pi/4])$, and $\boldsymbol{\gamma}_i$ from $U([0.005, 0.05])$ independently for each network unit. We found that WCRNNs with such informed heterogeneity performed on par with the best homogeneous configuration of WCRNNs on all datasets considered, see Table 1. Our results show that the variety of memory timescales present in WCRNNs with heterogeneous residuals allows them to generalize well over different datasets and therefore obtain increased practical expressivity compared to networks with homogeneous residuals. In particular, it follows that informed heterogeneity can be used to avoid a computationally expensive search for the best performing residual configuration.

4.4. Non-linear residuals

So far, we have only studied WCRNNs for which the residual was given by a linear map. In this section, we study two non-linear variants of WCRNNs of the form

$$\mathbf{x}_{t+1} = \sigma(\mathbf{R}\mathbf{x}_t) + \gamma\sigma(\mathbf{W}\mathbf{x}_t + \mathbf{S}_t). \tag{18}$$

and

$$\mathbf{x}_{t+1} = \sigma(\mathbf{R}\mathbf{x}_t + \gamma(\mathbf{W}\mathbf{x}_t + \mathbf{S}_t)). \tag{19}$$

For ease of notation, we will refer to networks defined by (18) as type A and to networks defined by (19) as type B. Note that the observations from Section 3 also hold for the non-linear case presented here, meaning that the weak coupling ensures that the memory properties of the networks (18) and (19) are still mostly determined by the residual connections. Moreover, when the non-linearity is not in a strongly saturating regime, the eigenvalues of the residuals matrices of both network types A, B still influence the network dynamics to a large extent. This is why we continue to distinguish subcritical, critical, and supercritical nonlinear WCRNNs, as before. We note that due to the presence of the non-linearity σ, this classification is less strict than for WCRNNs with linear residuals. To investigate these networks, we performed the same set of experiments on the MNIST datasets as in Sections 4.1, 4.2, and 4.3.

When training networks of both types, we found the same trade-off between efficiency and stability of learning dynamics as observed in networks with linear residuals. Similarly to linear networks, the non-linear WCRNNs with the highest performance and best learning efficiency were the networks that have eigenvalues of residual matrices close to critical value 1, see Fig. 6 A. Subcritical networks of type A showed the best performance and the fastest convergence, in contrast to type B networks, for which the best practical expressivity was achieved by weakly supercritical networks. For both studied non-linear variants of WCRNNs, we also found peaks in the learning efficiency and performance when trained with rotational residuals on sMNIST, see in Fig. 6 B. Interestingly, we found that the performance of networks of type B showed stronger sensitivity to angular frequencies of rotational residuals compared to networks of type A. These differences can be explained by the fact that one common non-linearity better prevents chaotic dynamics, but makes biases from initialization have a stronger effect on performance. We also observed that both network types were able to achieve similar performance levels as their highest performing homogeneous variants when equipped with informed heterogeneity from Section 4.3 (data not shown).

Notably, the network of type B can be considered as a weakly coupled residual initialization of an Elman RNN, because the residual matrix and other weight matrices are subject to the same non-linearity. In agreement with previous studies (Arjovsky et al., 2016), learning long tasks was found to be challenging for classic Elman networks even after applying standard techniques such as identity residuals and gradient clipping (Table 2, $\text{Elman}_{\text{res+clip}}$ and Elman_{wc}), emphasizing the advantage of the weakly coupled residual initialization scheme presented in (19).

We furthermore compared the presented variants of WCRNNs with other architectures that are known to be well-suited for working with long sequences such as LSTM (Gu, Gulcehre, Paine, Hoffman, & Pascanu, 2020) and S4 (Gu, Goel, & Ré, 2021), see Table 2. This comparison also includes the more challenging grayscale sequential CIFAR dataset (sCIFAR10) which is part of Long Range Arena benchmark designed to evaluate model performance on long sequences (Tay et al.,

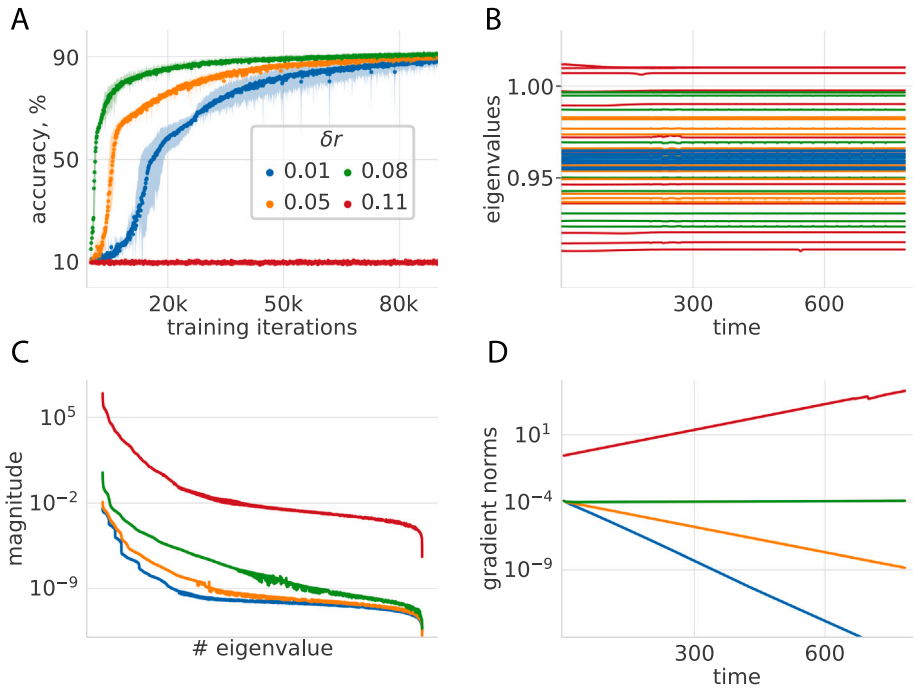

Fig. 5. Performance of heterogeneous subcritical WCRNNs on sMNIST with $r_0 = 0.96$ and different levels of heterogeneity δr (color-coded). **A**. Test accuracy as a function of training iterations over 200 training epochs. The lines show mean test accuracy over 5 network instances with random weight initialization, shaded areas show the range between minimal and maximal values of test accuracy. **B**. Dynamics of eigenvalues of variational term $\mathbf{V}_x(f^t)$ for WCRNNs with different levels of heterogeneity. All eigenvalues computed using the same randomly chosen input. Lines show trajectories of 15 randomly chosen eigenvalues over time. **C**. Rank plot of the eigenvalue magnitudes of the Hessian of the loss function $\mathbf{H}_w(L)$ before training. Lines show eigenvalues that were computed for a randomly chosen batch on sMNIST test dataset. **D**. Evolution of norms of BPTT gradients as a function of time. Lines show gradient norms that were computed over a random input batch before training.

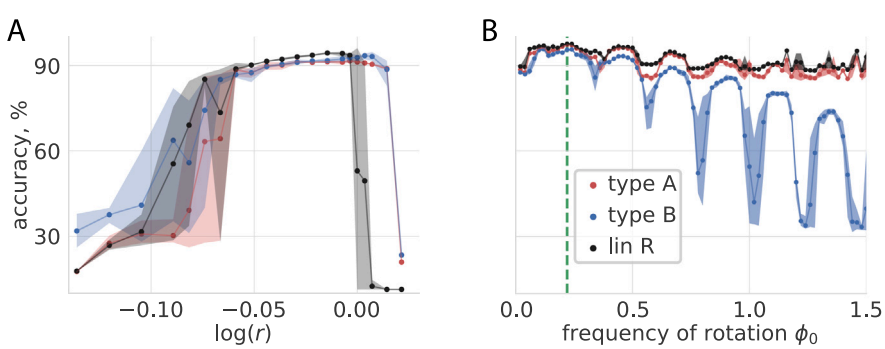

Fig. 6. Performance of WCRNNs with non-linear residuals of type A and type B on sMNIST in comparison with their linear analogs from previous experiments. The lines show the mean test accuracy over 5 network instances with random weight initialization, shaded areas show the range between maximal and minimal values of the test accuracy. **A**. The best test accuracy as a function of the maximal eigenvalue of the residual matrix (as controlled by the scalar parameter r). **B**. The best test accuracy as a function of angular frequency of rotation ϕ_0 of the homogeneous residual matrix. The green dashed line indicates the characteristic frequency of sMNIST $\phi_c = 2\pi/28 \approx 0.22$.

Table 1
Practical expressivity of informed heterogeneous WCRNNs compared to homogeneous WCRNNs, assessed by best test accuracy and the number of training iterations to reach a minimal performance (MP).

	Homogeneous	Heterogeneous	Homogeneous	Heterogeneous
	accuracy, %	accuracy, %	iter. to MP	iter. to MP
sMNIST	96.05	98.22	200	200
spMNIST	92.66	95.37	2076	1038
	RMS	RMS	iter. to MP	iter. to MP
ADD100	$2.0 \cdot 10^{-5}$	$4.6 \cdot 10^{-5}$	9298	8698
ADD200	$5.1 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$	$2.4 \cdot 10^4$	$3.1 \cdot 10^4$
ADD400	$1.5 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$7.4 \cdot 10^4$	$9.5 \cdot 10^4$
ADD800	$9.1 \cdot 10^{-4}$	$19.1 \cdot 10^{-4}$	$2.9 \cdot 10^5$	$4.1 \cdot 10^5$

Table 2
Best performance of various RNN architectures on a number of benchmark tasks. $\text{Elman}_{\text{res+clip}}$ denotes the Elman network with an identity residual and gradient clipping. Elman_{wc} denotes the Elman network with weakly coupled residual initialization (type B network). $\text{WCRNN}_{\text{best}}$ denotes the informed heterogeneous WCRNN. $\text{WCRNN}_{\text{no tanh}}$ denotes a WCRNN with the same residual and γ as $\text{WCRNN}_{\text{best}}$, but without the tanh non-linearity. LSTM denotes the default PyTorch implementation of an LSTM network. Elman and WCRNN networks have 100 units, LSTM networks 47 units (resulting in each model having around 10k trainable parameters). State-of-the-art results are given in the form of UR-LSTM (around 250k parameters for ADD, around 4M parameters for sMNIST/psMNIST, around 32M parameters for sCIFAR10) and S4 networks (around 300k parameters). Note the significantly increased model sizes of UR-LSTM and S4 in with respect to the other models. A dash (-) indicates that no data is available.

	sMNIST acc. %	psMNIST acc. %	ADD100, 200, 400, 800 RMS	sCIFAR10 acc. %
$\text{Elman}_{\text{res+clip}}$	36.53	78.59	$1.5 \cdot 10^{-5}$, $3.8 \cdot 10^{-5}$, fail, fail	18.7
Elman_{wc}	97.33	92.98	$1.1 \cdot 10^{-3}$, $2.9 \cdot 10^{-3}$, $8.0 \cdot 10^{-2}$, fail	38.49
$\text{WCRNN}_{\text{best}}$	98.22	95.37	$4.6 \cdot 10^{-5}$, $3.3 \cdot 10^{-5}$, $3.1 \cdot 10^{-4}$, $1.9 \cdot 10^{-3}$	47.21
$\text{WCRNN}_{\text{no tanh}}$	91.15	90.11	fail, fail, fail, fail	29.79
LSTM	93.3	90.29	$1.0 \cdot 10^{-6}$, $1.0 \cdot 10^{-6}$, $2.0 \cdot 10^{-6}$, $6.0 \cdot 10^{-6}$	58.13
UR-LSTM (Gu et al., 2020)	99.28	96.96	$1 \cdot 10^{-10}$ (ADD2000)	71.0
S4 (Gu et al., 2021)	99.63	98.7	–	91.8

2020). We also compared these networks with a variant of WCRNN that lacks the non-linearity ($\text{WCRNN}_{\text{no tanh}}$), showing the importance of the non-linear activation function. We found that heterogeneous WCRNNs with informed memory properties are able to achieve performance levels that are competitive with the much bigger and complex SOTA models for some datasets.

5. Discussion

In this work, we introduced weakly coupled residual recurrent networks (WCRNNs) and studied how their recurrent residual connections influence network dynamics, properties of fading memory, and practical expressivity. A dynamical systems analysis of WCRNNs allowed us to uncover a connection between network dynamics and the weight dynamics resulting from BPTT training. Based on this analysis, we predicted a trade-off between the stability of the learning dynamics and the learning efficiency of the backpropagation algorithm. In line with this prediction, simulation results showed that WCRNNs with dynamics close to the edge of chaos achieve greater practical expressivity than more subcritical or supercritical networks. Moreover, we found that several classes of informed residual connections could yield effective inductive biases for WCRNNs. In particular, we found that (i) rotational residuals are beneficial when they match the characteristic spectral properties of the data, (ii) residuals resulting in subcritical fading memory are favorable when temporally distant dependencies are present in the data, and (iii) heterogeneous residuals can increase the networks' practical expressivity by providing an informed range of memory timescales. In addition, heterogeneity can help avoid the computationally expensive search required to find the best-performing configuration for a homogeneous network.

Over the years, many approaches have been proposed to overcome the EVGP encountered when training RNNs with BPTT. Among those are gated architectures, such as long-short-term memory networks (LSTM) (Hochreiter & Schmidhuber, 1997) and gated recurrent units (GRU) (Cho et al., 2014), which in many cases still require gradient clipping to achieve high practical expressivity (Pascanu et al., 2013). Furthermore, models that place constraints on weight matrices such as orthogonal (Helfrich, Willmott, & Ye, 2018), unitary (Arjovsky et al., 2016) or antisymmetric weight matrices (Chang et al., 2019) were proposed to mitigate the EVGP. In contrast to these models, the WCRNNs proposed here do not impose strict conditions on weight matrices, as this can limit the practical expressivity of networks (Vorontsov, Trabelsi, Kadoury, & Pal, 2017). In that sense, WCRNNs are similar to RNNs with a specific initialization of their recurrent weights (for example, using the identity or orthogonal matrices (Le, Jaitly, & Hinton, 2015; Mishkin & Matas, 2015)), but differ in that the weak coupling in WCRNNs ensures the stability of the gradient properties throughout the learning process.

We also note that in our experiments we only used normal residual matrices and that it is known that non-normal initialization can lead to higher levels of practical expressivity (Kerg et al., 2019). Thus, the thorough study of how normality, diagonalizability, and other properties of residual matrices affect the performance of WCRNNs could be a potential direction for future research.

Although residual networks were first introduced without dynamical systems theory in mind, it was later shown that residual networks possess efficient gradient propagation properties in the infinite-width approximation when their dynamics are close to the edge of chaos (Yang & Schoenholz, 2017). Moreover, this approximation has been used to show the role of initialization schemes in shaping "lazy" or "rich" regimes of learning dynamics (Chizat, Oyallon, & Bach, 2019; Flesch, Juechems, Dumbalska, Saxe, & Summerfield, 2021; Geiger, Spigler, Jacot, & Wyart, 2020; Liu et al., 2023). Interestingly, the presence of weak coupling in WCRNNs resembles the infinite-width approximation as the influence of recurrent weights is scaled down by the weak coupling factor γ. We found that over training, the weights of WCRNNs often changed by an order of magnitude with respect to their initialization values, indicating a "rich" learning regime. However, the unique property of WCRNN is that the weak coupling allows for "rich" learning while maintaining stable memory properties.

Residual networks also sparked new interest in a dynamical systems approach due to the fact that in the limit of an infinite-depth approximation they can be understood as a system of differential equations, the NeuralODE approach (Chen, Rubanova, Bettencourt, & Duvenaud, 2018). However, in contrast to WCRNNs, architectures based on the NeuralODE approach are limited to describe continuous-like dynamics. Notably, recent research on various weight initialization methods in NeuralODE approach has revealed the crucial role of the properties of network dynamics for the training of these networks (Christodoulou, Vogels, & Agnes, 2022; Jarne, 2023; Jarne & Laje, 2023), in line with the findings of the present study.

Furthermore, recently introduced methods grounded in geometric principles and similarity metrics (Ostrow, Eisen, Kozachkov, & Fiete, 2023; Schuessler, Mastrogiuseppe, Ostojic, & Barak, 2023) allow for the detection of dynamical structures in recurrent neural networks. The application of these methods to WCRNNs is left for a future study.

Our results are consistent with previous studies on the benefits of residual networks with dynamics at the edge of chaos (Schoenholz et al., 2016), and also provide an additional perspective on the previously shown computational advantage of recurrent networks consisting of oscillatory units over non-oscillating architectures (Effenberger et al., 2023; Norcliffe et al., 2020; Rusch & Mishra, 2020). Our findings also agree with studies in the field of neuroscience, indicating that the brain seems to operate close to criticality but in a slightly subcritical regime, as this has computational advantages (Wilting et al., 2018; Wilting & Priesemann, 2019). In addition, our results also agree with recent studies that suggest the functional role of neural heterogeneity (Effenberger et al., 2023; Perez-Nieves, Leung, Dragotti, & Goodman, 2021; Sánchez-Puig, Zapata, Pineda, Iñiguez, & Gershenson, 2023).

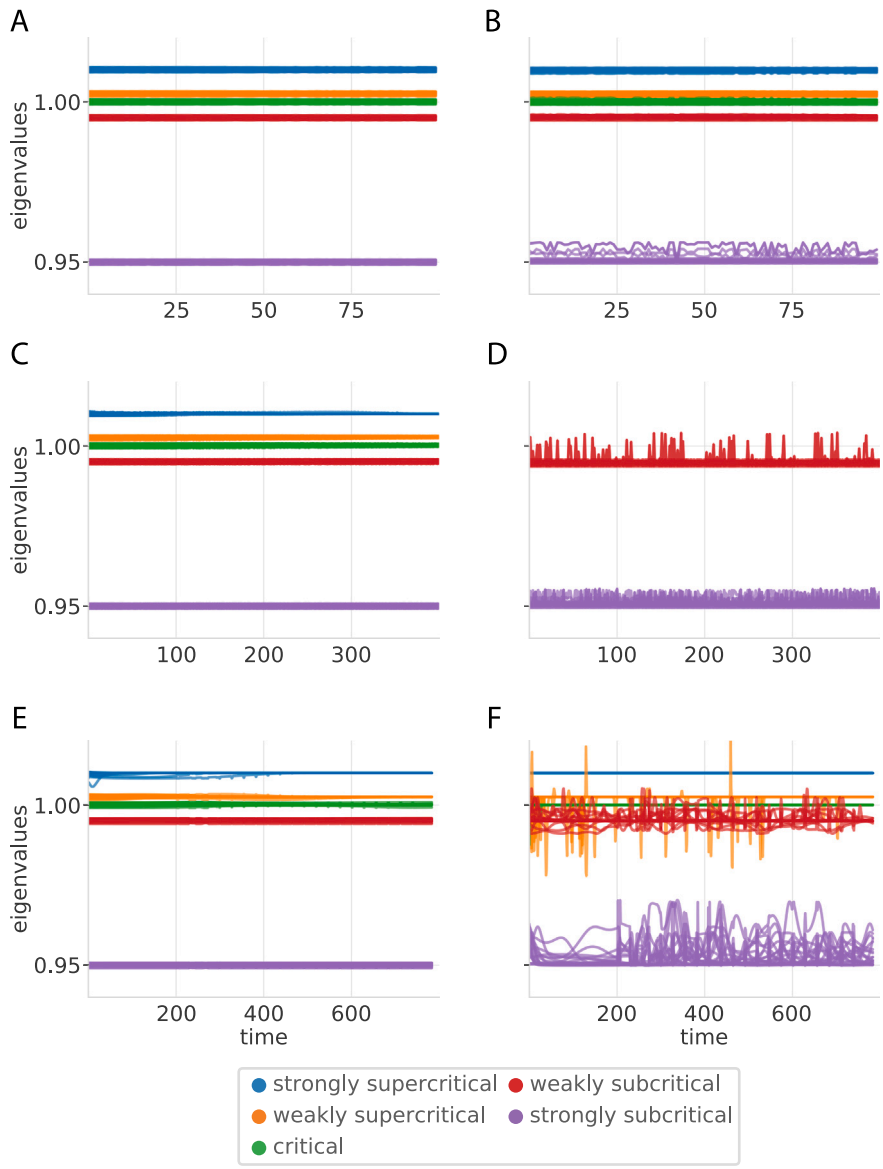

Fig. B.1. Dynamics of eigenvalues of variational term $\mathbf{V}_x(f^t)$ before training and after 200 training epochs. Lines show trajectories of 20 randomly chosen eigenvalues over time for a randomly chosen input digit. Colors indicate network type, strongly supercritical have $r = 1.01$, weakly supercritical have $r = 1.0025$, critical have $r = 1$, weakly subcritical have $r = 0.995$, strongly subcritical have $r = 0.95$. A. ADD100 before training; B. ADD100 after training; C. ADD400 before training; D. ADD400 after training; E. psMNIST before training; F. psMNIST after training;.

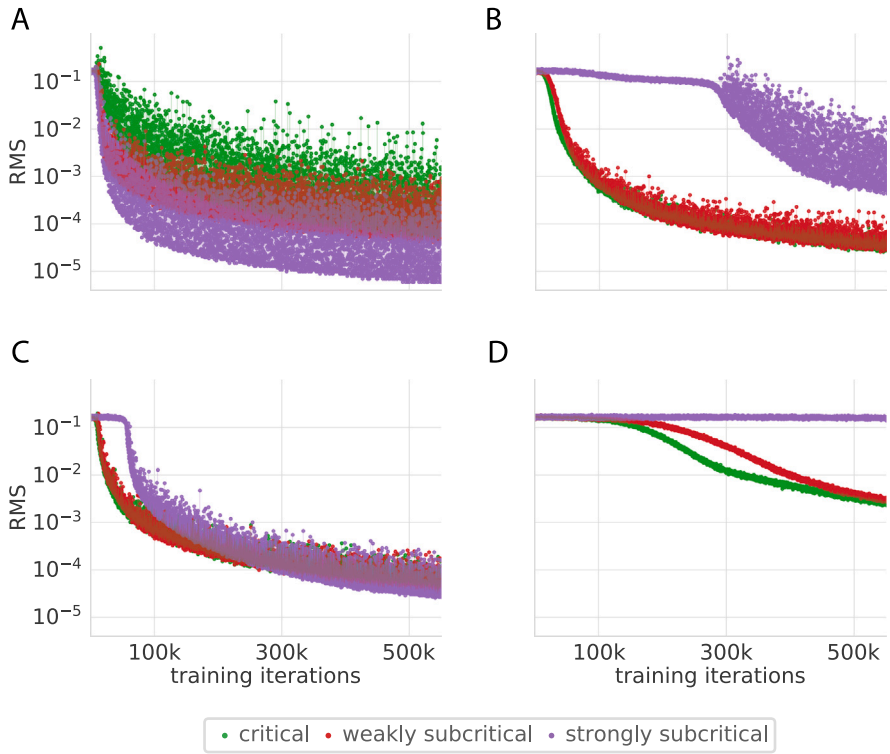

Fig. B.2. Learning trajectories for WCRNNs subject to different learning rates and coupling constants γ, trained on the ADD100 dataset. Lines show test accuracy measured in RMS as a function of training iterations over 150 training epochs. A. Learning rate: $\eta = 0.1$, coupling constant: $\gamma = 0.01$. Note that the learning dynamics are unstable. B. Learning rate: $\eta = 0.1$, coupling constant: $\gamma = 0.001$. Note that the decrease in γ results in more stable but slower learning dynamics compared to A. C. Learning rate: $\eta = 0.01$, coupling constant: $\gamma = 0.01$. Note that the decrease in learning rate results in more stable but also slower learning dynamics. D. Learning rate: $\eta = 0.01$, coupling constant: $\gamma = 0.001$. Note the very stable and also very slow learning dynamics.

We also anticipate that our results will be relevant in the context of continuous learning, where effective memory is known to be essential to avoid catastrophic forgetting (Hadsell, Rao, Rusu, & Pascanu, 2020). Furthermore, we hypothesize that our approach to incorporate informed biases into residuals could find its application in feedforward architectures, because residual connections are widely used in a range of modern architectures (He et al., 2016; Ronneberger et al., 2015; Vaswani et al., 2017).

6. Conclusion

In a broader context, we believe that the findings presented here are in line with a recent surge in interest in RNNs, seeking to overcome the EVGP and training inefficiencies (Orvieto et al., 2023; Zucchet, Meier, Schug, Mujika, & Sacramento, 2023). These developments show that the careful design of RNNs can result in state-of-art performance on long range memory tasks. In some cases, RNNs were shown to surpass the performance of feed-forward Transformer-based architectures (Tay, Dehghani, Bahri, & Metzler, 2022), while overcoming the drawbacks of their dot-product attention, where memory and computational complexity exhibit quadratic scaling with sequence length (Peng et al., 2023). In this work, we have defined weakly coupled recurrent networks (WCRNNs) that possess well-defined Lyapunov exponents and have shown how the practical expressivity and training stability of WCRNNs are influenced by their dynamics, and how heterogeneous and informed residuals can increase further practical expressivity without increasing system size. We believe that this is an important step forward in the understanding of residual RNNs on both a theoretical and practical level.

In summary, our results show how the properties of fading memory resulting from RNN dynamics play a crucial role in shaping the backpropagation-induced learning dynamics, and how the implementation of informed memory properties by means of residual connections can improve the practical expressivity of RNNs.

CRediT authorship contribution statement

Igor Dubinin: Conceptualization, Formal analysis, Investigation, Software, Visualization, Writing – original draft, Writing – review & editing. **Felix Effenberger:** Conceptualization, Software, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by the Ernst Strüngmann Institute for Neuroscience in Cooperation with Max Planck Society.

Appendix A. Algorithm for computation of Lyapunov exponents

Here, we explain how we compute the Lyapunov exponents for WCRNNs defined by Eq. (1). In theory, Laypunov exponents can be computed directly from the Oseledts equation (4), but in practice the P-dimensional volume of the tangent space tends to align with the eigenvector associated with the largest eigenvalue, resulting in degenerate matrices and therefore numerical problems. To overcome such problems, an orthonormalization procedure was introduced; see, for example, (Sandri, 1996). This procedure orthonormalizes the tangent space along a system trajectory that gives more stable estimates of the

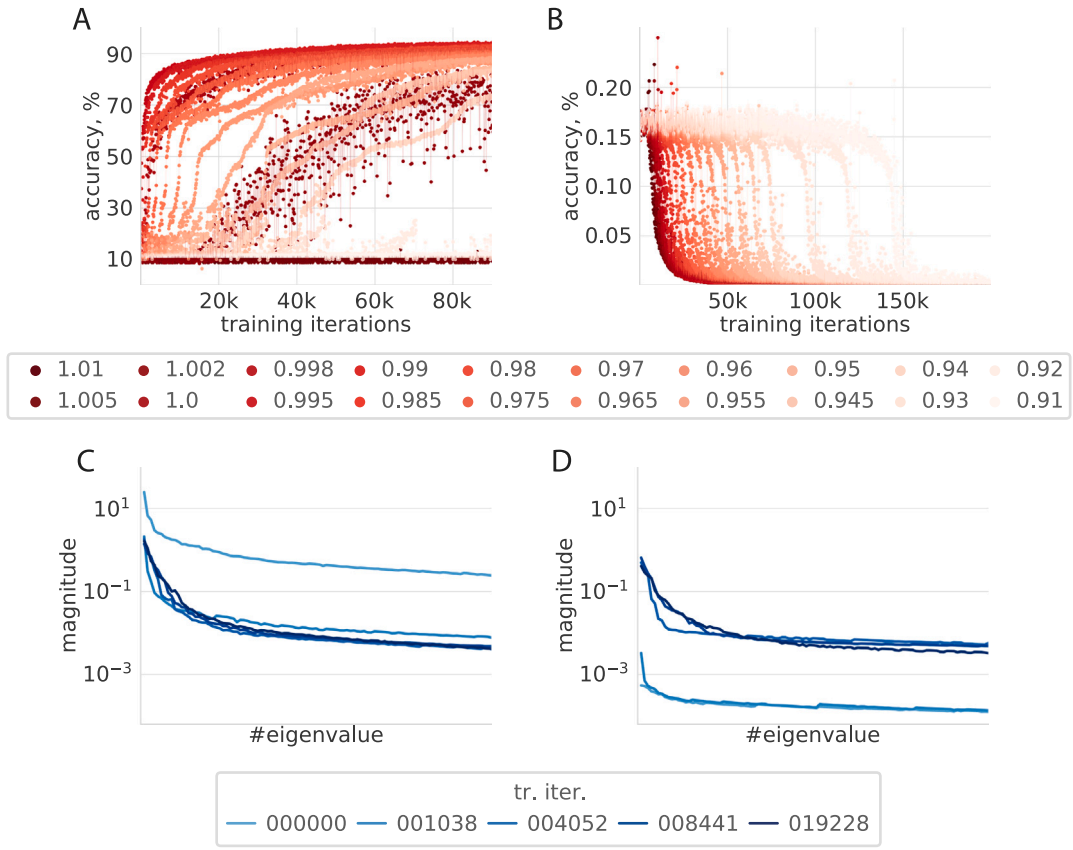

Fig. B.3. Learning dynamics for WCRNNs. Plot A and B show learning trajectories for the studied range of the residual connection strength $r \in [0.91, 1.02]$. Rank plots C and D show the eigenvalue magnitudes of the Hessian of the loss function $\mathbf{H}_\omega(L)$ during training. The eigenvalues were computed for a randomly chosen batch from the sMNIST test set. A. Lines show test accuracy as a function of training iterations for sMNIST dataset. B. Lines show test error measured in RMS as a function of training iterations for the ADD100 dataset. C. Lines show eigenvalues for a critical WCRNN. Note the decrease in magnitudes over learning. D. Lines show the eigenvalues for a strongly subcritical WCRNN. Note the increase in magnitudes over learning.

scaling of the P dimensional volume. For the case of a discrete system (1), we iterate it alongside with its variational equation

$$\delta\mathbf{P}(t+1) = \mathbf{J}_\mathbf{x}(\mathbf{f}^t(\mathbf{x}_1))\delta\mathbf{P}(t). \tag{A.1}$$

If we apply the chain rule for t times, we obtain Eq. (2) from the main text

$$\delta\mathbf{P}(t+1) = \mathbf{V}_\mathbf{x}(\mathbf{f}^t(\mathbf{x}_1))\delta\mathbf{P}(1), \tag{A.2}$$

where $\mathbf{V}_\mathbf{x}(\mathbf{f}^t(x_1)) = \mathbf{J}_\mathbf{x}(\mathbf{f}^t(x_1))\mathbf{J}_\mathbf{x}(\mathbf{f}^{t-1}(\mathbf{x}_1))...\mathbf{J}_\mathbf{x}(\mathbf{f}(\mathbf{x}_1))$. To compute the Lyapunov exponents, we need to estimate the eigenvalues of the variational term $\mathbf{V}_\mathbf{x}(\mathbf{f}^t(x_1))$ with the initial condition of $\delta\mathbf{P}(1) = \mathbf{I}$.

The orthonormalization procedure is performed as follows. After every iteration of the system (1), we compute its Jacobian and evolve the variational equation (A.1), where the P dimensional volume of the tangent space is initialized with the identity matrix. Next, we perform a QR decomposition on the resulting volume of the tangent space $\mathbf{Q}_1$. The QR decomposition of $\mathbf{Q}_1$ produces two matrices $\mathbf{Q}_2$ and $\mathbf{R}_2$, where the orthonormal matrix $\mathbf{Q}_2$ defines the rotational component, and the diagonal elements of $\mathbf{R}_2$ define the volume scaling. Finally, we collect the eigenvalues of $\mathbf{R}$, which describe the instantaneous transformation of the volume, and perform the next step with a new volume from the tangent space $\mathbf{Q}_2$. Furthermore, if we repeat this procedure for t times, the final variational term is

$$\mathbf{V}_\mathbf{x}(\mathbf{f}^t(\mathbf{x}_1)) = \mathbf{Q}_t\mathbf{R}_t \dots \mathbf{R}_1. \tag{A.3}$$

From (A.3) it follows that if the logarithms of the eigenvalues $\mathbf{R}_t$ converge, they define unique Lyapunov exponents according to the Oseledets equation (4). Therefore, we show the convergence of eigenvalues of the variational term in all of our figures. We also want to note that this method is usually applied to autonomous systems, but the design of the residuals of the system (11) allows us to study non-autonomous dynamics.

The pseudocode for the orthonormalization algorithm is as follows:

Algorithm 1 Collect eigenvalues of the variational term

```
Initialize: x, Q, I
while t ≤ L do
    x ← f(x, I)
    J ← df/dx
    Q ← JQ
    Q, R ← qr(Q)
    λ ← append(diag(R))
end while
```

Appendix B. Supplementary figures

See Figs. B.1–B.4.

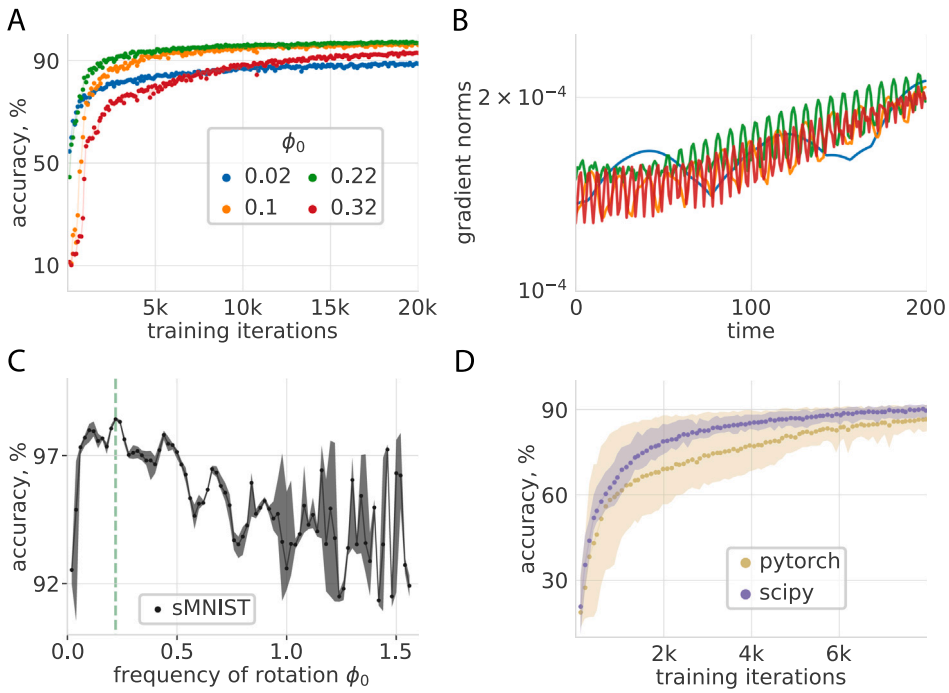

Fig. B.4. A. Learning trajectories for WCRNNs with different angular frequencies of the rotational residuals ϕ_0, trained on the sMNIST dataset. Lines show the test accuracy as a function of training iterations. B. Evolution of the norms of the gradients $\frac{\partial L}{\partial f^t}$ as a function of inference time for homogeneous WCRNNs with different angular frequencies of the rotational residuals ϕ_0. Lines show gradient norms computed on a random batch of the sMNIST test set before training. C. Best test accuracy of rotational WCRNNs trained on sMNIST over 200 epochs as a function of the angular frequency of the rotational residual ϕ_0. D. Comparison between different implementations for the construction of heterogeneous orthonormal residual matrices, comparing `scipy.stats.ortho_group.rvs` implementation (red) and our `pytorch.rand` implementation (blue). Test accuracy for WCRNNs with different residual matrices as a function of training iterations for first 7000 training iterations. Lines show average accuracy over 5 networks with random weight initialization, shaded area indicates the range between minimal and maximal values.

References

Arjovsky, M., Shah, A., & Bengio, Y. (2016). Unitary evolution recurrent neural networks. In *International conference on machine learning* (pp. 1120–1128). PMLR.

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, *14*(1), 115–133.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., et al. (2018). Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, *16*(7), 1413–1436.

Chang, B., Chen, M., Haber, E., & Chi, E. H. (2019). AntisymmetricRNN: A dynamical system view on recurrent neural networks. arXiv preprint arXiv:1902.09689.

Chen, R. T., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. arXiv preprint arXiv:1806.07366.

Chizat, L., Oyallon, E., & Bach, F. (2019). On lazy training in differentiable programming. In *Advances in neural information processing systems*: *vol. 32*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Christodoulou, G., Vogels, T. P., & Agnes, E. J. (2022). Regimes and mechanisms of transient amplification in abstract and biological neural networks. *PLoS Computational Biology*, *18*(8), Article e1010365.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*: *vol. 27*.

Eckmann, J.-P., & Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *The Theory of Chaotic Attractors*, 273–312.

Effenberger, F., Carvalho, P., Dubinin, I., & Singer, W. (2023). *The functional role of oscillatory dynamics in neocortical circuits: A computational perspective*. bioRxiv, Cold Spring Harbor Laboratory, http://dx.doi.org/10.1101/2022.11.29.518360.

Engelken, R., Wolf, F., & Abbott, L. F. (2020). Lyapunov spectra of chaotic recurrent neural networks. arXiv preprint arXiv:2006.02427.

Erichson, N. B., Azencot, O., Queiruga, A., Hodgkinson, L., & Mahoney, M. W. (2020). Lipschitz recurrent neural networks. arXiv preprint arXiv:2006.12070.

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., & Summerfield, C. (2021). *Rich and lazy learning of task representations in brains and neural networks*. http://dx.doi.org/10.1101/2021.04.23.441128.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*(3), 183–192.

Geiger, M., Spigler, S., Jacot, A., & Wyart, M. (2020). Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2020*(11), Article 113301.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). JMLR Workshop and Conference Proceedings.

Goyal, A., & Bengio, Y. (2022). Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society of London, Series A (Mathematical and Physical Sciences)*, *478*(2266), Article 20210068.

Gu, A., Goel, K., & Ré, C. (2021). Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396.

Gu, A., Gulcehre, C., Paine, T., Hoffman, M., & Pascanu, R. (2020). Improving the gating mechanism of recurrent neural networks. In *International conference on machine learning* (pp. 3800–3809). PMLR.

Hadsell, R., Rao, D., Rusu, A. A., & Pascanu, R. (2020). Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, *24*(12), 1028–1040.

Hanin, B., & Rolnick, D. (2019). Complexity of linear regions in deep networks. In *International conference on machine learning* (pp. 2596–2604). PMLR.

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

Helfrich, K., Willmott, D., & Ye, Q. (2018). Orthogonal recurrent neural networks with scaled Cayley transform. In *International conference on machine learning* (pp. 1969–1978). PMLR.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Jarne, C. (2023). Different eigenvalue distributions encode the same temporal tasks in recurrent neural networks. *Cognitive Neurodynamics*, *17*(1), 257–275.

Jarne, C., & Laje, R. (2023). Exploring weight initialization, diversity of solutions, and degradation in recurrent neural networks trained for temporal and decision-making tasks. *Journal of Computational Neuroscience*, 1–25.

Kearns, M. J., & Vazirani, U. (1994). *An introduction to computational learning theory*. MIT Press.

Kerg, G., Goyette, K., Puelma Touzel, M., Gidel, G., Vorontsov, E., Bengio, Y., et al. (2019). Non-normal recurrent neural network (nnrnn): Learning long time dependencies while improving expressivity with transient dynamics. In *Advances in neural information processing systems*: *vol. 32*.

Kerg, G., Mittal, S., Rolnick, D., Bengio, Y., Richards, B., & Lajoie, G. (2022). On neural architecture inductive biases for relational tasks. arXiv preprint arXiv:2206.05056.

Le, Q. V., Jaitly, N., & Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. arXiv preprint arXiv:1504.00941.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Liu, Y. H., Baratin, A., Cornford, J., Mihalas, S., Shea-Brown, E., & Lajoie, G. (2023). How connectivity structure shapes rich and lazy learning in neural circuits. ArXiv.

Mastrogiuseppe, F., & Ostojic, S. (2018). Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, *99*(3), 609–623.

Miller, J., & Hardt, M. (2018). Stable recurrent models. arXiv preprint arXiv:1805.10369.

Mishkin, D., & Matas, J. (2015). All you need is a good init. arXiv preprint arXiv:1511.06422.

Norcliffe, A., Bodnar, C., Day, B., Simidjievski, N., & Liò, P. (2020). On second order behaviour in augmented neural odes. In *Advances in neural information processing systems*: *vol. 33*, (pp. 5911–5921).

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., et al. (2023). Resurrecting recurrent neural networks for long sequences. arXiv preprint arXiv:2303.06349.

Oseledets, V. I. (1968). A multiplicative ergodic theorem. Characteristic ljapunov, exponents of dynamical systems. *Trudy Moskovskogo Matematicheskogo Obshchestva*, *19*, 179–210.

Ostrow, M., Eisen, A., Kozachkov, L., & Fiete, I. (2023). Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis. arXiv preprint arXiv:2306.10168.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318). PMLR.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*: *vol. 32*.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., et al. (2023). RWKV: Reinventing RNNs for the transformer era. arXiv preprint arXiv:2305.13048.

Perez-Nieves, N., Leung, V. C., Dragotti, P. L., & Goodman, D. F. (2021). Neural heterogeneity promotes robust learning. *Nature Communications*, *12*(1), 1–9.

Rajan, K., Abbott, L., & Sompolinsky, H. (2010). Stimulus-dependent suppression of chaos in recurrent neural networks. *Physical Review E*, *82*(1), Article 011903.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–mICCAI 2015: 18th international conference, munich, Germany, October 5-9, 2015, proceedings, part III 18* (pp. 234–241). Springer.

Rusch, T. K., & Mishra, S. (2020). Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. arXiv preprint arXiv:2010.00951.

Sánchez-Puig, F., Zapata, O., Pineda, O. K., Iñiguez, G., & Gershenson, C. (2023). Heterogeneity extends criticality. *Frontiers in Complex Systems*, *1*, Article 1111486.

Sandri, M. (1996). Numerical calculation of Lyapunov exponents. *Mathematica Journal*, *6*(3), 78–84.

Schoenholz, S. S., Gilmer, J., Ganguli, S., & Sohl-Dickstein, J. (2016). Deep information propagation. arXiv preprint arXiv:1611.01232.

Schraudolph, N. N. (2002). Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, *14*(7), 1723–1738.

Schuessler, F., Mastrogiuseppe, F., Ostojic, S., & Barak, O. (2023). Aligned and oblique dynamics in recurrent neural networks. arXiv preprint arXiv:2307.07654.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., et al. (2020). Long range arena: A benchmark for efficient transformers. arXiv preprint arXiv:2011.04006.

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys*, *55*(6), 1–28.

Thomas, V., Pedregosa, F., Merriënboer, B., Manzagol, P.-A., Bengio, Y., & Le Roux, N. (2020). On the interplay between noise and curvature and its effect on optimization and generalization. In *International conference on artificial intelligence and statistics* (pp. 3503–3513). PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Vogt, R., Touzel, M. P., Shlizerman, E., & Lajoie, G. (2020). On Lyapunov exponents for RNNs: Understanding information propagation using dynamical systems tools. arXiv preprint arXiv:2006.14123.

Vorontsov, E., Trabelsi, C., Kadoury, S., & Pal, C. (2017). On orthogonality and learning recurrent networks with long term dependencies. In *International conference on machine learning* (pp. 3570–3578). PMLR.

Wang, Y., & Tian, F. (2016). Recurrent residual learning for sequence classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 938–943).

Wilting, J., Dehning, J., Pinheiro Neto, J., Rudelt, L., Wibral, M., Zierenberg, J., et al. (2018). Operating in a reverberating regime enables rapid tuning of network states to task requirements. *Frontiers in Systems Neuroscience*, *12*, 55.

Wilting, J., & Priesemann, V. (2019). 25 Years of criticality in neuroscience—established results, open controversies, novel concepts. *Current Opinion in Neurobiology*, *58*, 105–111.

Yang, G., & Schoenholz, S. (2017). Mean field residual networks: On the edge of chaos. In *Advances in neural information processing systems*: *vol. 30*.

Yue, B., Fu, J., & Liang, J. (2018). Residual recurrent neural networks for learning sequential representations. *Information*, *9*(3), 56.

Zucchet, N., Meier, R., Schug, S., Mujika, A., & Sacramento, J. (2023). Online learning of long range dependencies. arXiv preprint arXiv:2305.15947.