Criticality and Anomalous Diffusion Underlying Dynamical and Computational Properties of Deep Learning Networks

Cheng Kevin Qu

School of Physics
Faculty of Science
The University of Sydney

A thesis submitted in partial fulfilment of requirements for the degree of Doctor of Philosophy.
The research reported in this thesis was supported by the award of a Research Training Program Fee-Offset Scholarship.

February 2025

Statement of originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Cheng Kevin Qu, 28/Feb/2025

Authorship attribution statement

Listed below are involved references for Chapters 2–4 in this thesis. The * sign indicates equal contribution.

Chapter 2:
Extended critical regimes of deep neural networks. arXiv e-prints, page arXiv:2203.12967, March 2022.
Cheng Kevin Qu*, Asem Wardak* and Pulin Gong.
Under revision with major updates on the arXiv preprint version.
I was primarily responsible for this work, with an overall contribution of about 50%. I co-designed the study, performed the analysis and wrote the drafts of the manuscript with the co-authors.

Chapter 3:
A Fractional Diffusion implementation of self-attention in Transformers.
Cheng Kevin Qu, Andrew Ly and Pulin Gong.
To appear on arXiv.
I was primarily responsible for this work, with an overall contribution of about 70%. I co-designed the study, performed the analysis and wrote the drafts of the manuscript with the co-authors.

Chapter 4:
Main Text: **Anomalous diffusion dynamics of learning in deep neural networks.** *Neural Networks*, 149, 18–28, 10.1016, Jan 2022.
Guozhang Chen, Cheng Kevin Qu and Pulin Gong.
Published.
I was partially responsible for this work, with an overall contribution of about 10%. I co-designed the study with the co-authors, interpreted the analysis done by G.C. and P.G., and wrote the drafts of the paper.

Appendix: **Dysonian dynamics of learning.**
Cheng Kevin Qu and Pulin Gong.
In preparation.
I was primarily responsible for this work, with an overall contribution of about 80%. I co-designed the study, performed the analysis and wrote the drafts of the manuscript with the co-authors.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Cheng Kevin Qu, 28/Feb/2025

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Pulin Gong, 28/Feb/2025

Abstract

Artificial neural networks, inspired by biological neural systems, harness modern computing power to facilitate learning. The corresponding scaled-up models, known as deep neural networks (DNNs), consistently achieve state-of-the-art performance across various user-defined and scientific applications. Their success is largely driven by the integration of diverse operators, such as linear and convolutional layers, alongside self-attention blocks. The vast number of model parameters enable extensive analysis of network properties, including width and depth, allowing DNNs to be treated as mathematical objects in theoretical studies. Drawing from neuroscience and statistical physics, we impose statistical assumptions on artificial neural networks and their learning processes. In particular, heavy-tailed, power-law distributions prevalent in many complex systems, give rise to heterogeneity in neural connectivity, increasing the likelihood of extreme connection strengths. Leveraging heavy-tailed non-Hermitian random matrix theory, we reveal an extended criticality-like regime between the chaotic and ordered phases that display ideal information propagation and training dynamics, further extending classical Gaussian results. Next, we generalize the diffusion processes underlying the self-attention mechanism of Transformers to the class of Lévy processes, incorporating long-range interactions between tokens on the embedding space manifold. This enhances the self-attention performance and gives rise to fractional diffusion dynamics in its continuous space-time limit. Finally, we analyze the short- and long-term behavior of stochastic gradient descent (SGD) in various settings, revealing non-equilibrium dynamics driven by anomalous diffusion that naturally arise and benefit the deep learning process. The findings presented in this thesis illustrate the importance of developing theoretical tools for understanding shared principles governing artificial and biological systems.

Acknowledgements

I would like to express my deepest gratitude to my PhD advisor, Pulin Gong, for introducing me to my research. I am truly thankful for his invaluable advice, insights, and enthusiasm, which have inspired and motivated me to always pursue a deeper understanding of science.

Thanks are also due to my co-advisors Peter Robinson and Ben Goldys, as well as the Australia Government for supporting my research.

I would also like to thank my friends and collegues Asem Wardak, Guozhang Chen and Andrew Ly who have taught and encouraged me more than they will ever recognize. It has also been a pleasure to share the office with the other group members including Xian Long, Shengcong Ni and Brendan Harris.

Finally, I would like to forward my appreciation to my family, to whom I am forever indebted. My parents and grandparents have been caring, open-minded and everything I could ask for. To my brother, Matthew, thank you too!

Contents

Chapter 1

Introduction

Deep learning [1] has revolutionized the landscape of numerous fields, including vision tasks [2], natural language processing [3], protein-folding [4], etc. Its paradigm is based on two fundamental components: the artificial neural network architecture inspired by the compact yet complex structure of the human brain; and the training process that assimilates human learning experience by capturing relevant knowledge from structured input data to make informed decisions within specific domains. Despite the success of deep learning in various scientific fields, its underlying principles remain largely unclear. Thus, understanding the field not only offers guidance to more efficient approaches but also sheds light on shared mechanisms between artificial and biological neural networks. We shift our focus away from practical applications and employ theoretical methods to investigate the foundations of deep learning.

The success of AlexNet [2] in the ImageNet Large Scale Visual Recognition Challenge [5] has popularized the implementation of deep neural networks (DNNs). Contrary to traditional models, deep networks are overparametrized which incorporate multiple layers to refine as well as extract progressively more abstract features from the input datum. After training, DNNs can facilitate decisions and predictions with higher accuracy in various scenarios, including supervised, unsupervised, and reinforcement learning. In the case of supervised learning, every input is paired with a label, and deep learning refers to minimization of a predefined error function, which measures the discrepancy of a deep network's output to the ground truth. Computational power has been able to continuously upscale the number of trainable parameters. Thus, for the past decade, the focal point of machine learning has gradually shifted towards deep learning where these large networks are particularly of great interest. Despite the success of deep learning, its underlying principles remain largely unclear. With increasing complexity due to the upscaling of trainable parameters, analytically solving the associated dynamics quickly becomes infeasible.

In this thesis, we leverage statistical physics and probability theoretical tools to model the underlying structure of deep learning, displaying an interplay of discrete and continuous dynamics. The precision of these approximation schemes scale with the system size, leading to increasingly accurate results as the system grows. For instance, the large width of hidden layers norms from Gaussian networks can invoke the central limit theorem (CLT), allowing for a precise approximation of the layerwise dynamics in a mean-field limit. Similarly, the characterization of the rate-based model fixed points can be inferred from the spectral density of the underlying Jacobian, a column-scaled connectivity matrix shifted by the identity. In the case of infinite neuron sites, the limiting eigenspectrum can be explicitly computed from non-Hermitian random matrix theory (RMT). We extend these previous findings to study heavy-tailed networks where the general central limit theorem (GCLT) and heavy-tailed non-Hermitian random matrix theory must be incorporated. Large limits also appear in more sophisticated models like Transformers where sequential embeddings are treated as spatial structures. The SoftMax self-attention mechanism requires yield interactions between tokens captured by the dot product, which describes their similarity. In the regime of infinite-sequence length (spatial) and infinitesimal step sizes, the self-attention mapping is described by a (second-order) continuity equation [6]. We further generalize this to the fractional heat equation with dynamics associated to Lévy processes. Last but not least, we unveil non-equilibrium dynamics in learning constituting anomalous diffusion. Motivated by all the previous findings, we consequently outline a new type of formulation of stochastic gradient descent as a matrix-values process which allows one to captured the eigenvalue and eigenvectors dynamics of the weight matrices.

1.1 Random neural networks

Understanding large neural networks, both artificial and biological, is of great significance due to their natural emergence in complex systems. In both settings, randomness is systematically introduced through inputs, environmental factors, learning, and other sources, making the study of random neural networks crucial for gaining insights into their information-processing capabilities. As a starting point, previous studies have assumed Gaussian statistics underlying the connectivities of networks [7, 8, 9, 10, 11], where the edge-of-chaos exists for various network types. However, empirical observations often contradict the common statistical Gaussianity assumption, which has been extensively explored in previous studies [12, 13, 14, 15]. To address this, we extend our analysis to more realistic heterogeneous networks, where the connectivity strengths follow a scale-free distribution governed by a power law. Under this property, a notion of extended criticality emerges for

these networks, capturing inherent variability and complexity found in real-world systems.

1.1.1 Fully-connect networks as Gaussian processes

We begin by considering a multilayer perception (MLP) or fully-connected feed-forward neural network (FFNN) of fixed depth L. For an input $x := x^0 \in \mathbb{R}^{N_0}$, it propagates through the network based on

$$x_i^l = \phi(h_i^l) \ , \quad h_i^l = W_{ij}^l x_j^{l-1} + b_i^l \ , \quad l = 1, \cdots, L \tag{1.1}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the activation function, typcically non-linear and $h^l, x^l \in \mathbb{R}^{N_l}$. Conventionally, the models are studied with (connectivity) weights and biases following $W_{ij}^l \sim \mathcal{N}(0, \sigma_w^2/N_{l-1})$ and $b_j \sim \mathcal{N}(0, \sigma_b^2)$ respectively. In the infinite-width limit, as $N_l \to \infty$ for all l, these models converge to *Gaussian processes* [10]. For MLPs, it is important to understand the network hidden states by examining the covariance matrix $q^l(x, x') := \mathbb{E}\left[h_i^l(x) h_i^l(x')\right]$ constructed by the geometry of two inputs as they propagate through the network layers. In the infinite-width regime where the limits are taken recursively, i.e. $N_1 \to \infty$, then $N_2 \to \infty$, $\cdots$, eventually $N_L \to \infty$, a mean-field limit can be obtained for these expressions for a network with fixed depth

$$q^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z_i^{l-1} \sim \mathcal{GP}\left(0, K^{l-1}\right)} \left[\phi\left(z_i^{l-1}(x)\right) \phi\left(z_i^{l-1}(x')\right)\right] . \tag{1.2}$$

Here z_i^l is drawn from a Gaussian process with zero mean function and covariance kernel q^{l-1}. For bounded activation functions, the variance $q^l(x, x)$ and correlation $c^l(x, x') := q^l(x, x')/(\sqrt{q^l(x, x)}\sqrt{q^l(x', x')})$ converge to some limiting value q^∞ and c^∞ respectively which depend on the phase (σ_w, σ_b) [9]. Other unbounded activation functions, such as ReLU $\phi(x) = \max\{0, x\}$ and its variants, have also been extensively explored [11], resulting in distinct EOC curves, though derived through comparable methods. However, the hyperbolic tangent function emerges naturally in recurrent networks, and we will focus on this case due to its relevant connections.

Prior to training, when the network parameters are randomly initialized, both the hidden states across the network depth and the gradients of trainable parameters with respect to the loss must be well-behaved. Ideally, substantial differences between various input classes are expected, while the variability within the same should remain minimal. Thus, a delicate balance of geometrical compression and expansion is required to appropriately separate different classes and draw the same together. On the other hand, we require suitable values of the gradients as extremely large and small gradients can lead to chaotic and stagnant learning respectively. The hidden states and gradients are thus connected through the

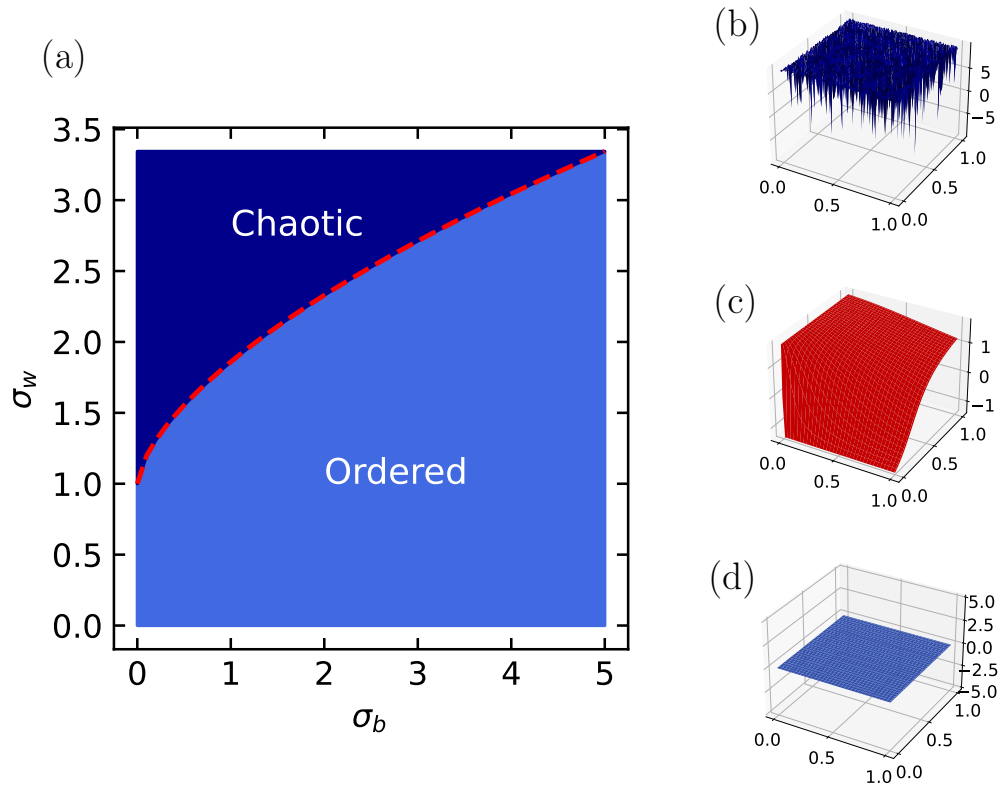

Figure 1.1: Phase transition of Gaussian networks. (a) The phase transition of an infinite-width Gaussian MLP with tanh activation; red dashed line represents the edge-of-chaos (EOC) (Eq. 1.3). (b–d) From top to bottom are outputs of a FFNN at the chaotic phase, EOC and order phase (z axis with scale 10^{-2}) respectively, the network is of depth $L = 300$ and width $N_l = 20$ for all $l \in [0, L]$. Adapted from [10, 11].

duality of forward and backward propagation through the (input-output) Jacobian [16]. The *edge of chaos* (EOC) is the set of values for $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$ that satisfy

$$\chi_1 := \sigma_w^2 \mathbb{E}\left[\phi'\left(\sqrt{q^\infty(\sigma_b, \sigma_w)}Z\right)^2\right] = 1, \tag{1.3}$$

where $Z \sim \mathcal{N}(0,1)$. It is represented by the dashed red line in Fig. 1.1(a). In the *ordered phase* ($\chi_1 < 1$) and *chaotic phase* ($\chi_1 > 1$), the correlation $c^l(x, x')$ of any input $x \neq x'$ converges exponentially fast to 1 and some limiting value $c \in (0,1)$ respectively. Moreover, one can expect the output of the neural network to be constant in the first case and non-continuous almost everywhere in the second (Fig. 1.1(b,d)). At the EOC ($\chi_1 = 1$), the correlation $c^l(x, x')$ also converges to 1, however, at a sub-exponential rate [11]. Any input pair would be indistinguishable at sufficiently large depth, the information of the inputs can nonetheless be retained at much deeper layers compared to the previous phases. The corresponding loss landscape also exhibits smoother properties which allow for better learning (Fig. 1.1(c)). Therefore, initialization of the weights at the EOC would be optimal for training Gaussian networks as deeper information propagation can be achieved.

1.1.2 Rate-based models

The complex systematic dynamics observed in biological networks often necessitate mathematical modeling to provide theoretical explanations and inferences. In contrast, artificial networks are built upon a set of predefined operations inspired by their biological counterparts. In both contexts, randomly connected firing-rate-based models provide a foundational basis for examining the effects of biological network connectivity whilst including sufficient nonlinearity to yield non-trivial dynamics. A leaky version is described by an ordinary differential equation (ODE) which evolves through time continuously [7]:

$$h_i'(t) = -h_i(t) + \sum_{j=1}^{N} W_{ij}\phi\left(h_j(t)\right). \tag{1.4}$$

The nonlinearity ϕ converts the internal neural activity to an output firing rate, and a typical sigmoidal choice $\phi := \tanh$ suppresses instability in neural activity. Similarly, limiting dynamics of the system $N \to \infty$ is of great interest under the statistics $W_{ij} \sim \mathcal{N}(0, g^2/N)$ where the gain parameter g controls the level of fluctuations. At time t, the Jacobian matrix given by $-I + W \operatorname{diag}_j \phi'\left(h_j(t)\right)$ reveals a phase transition with a critical value corresponding to the variance of the connectivity strengths at $g = 1$. For values above this threshold ($g > 1$), a chaotic phase emerges, characterized by a macroscopic number of unstable, nonzero fixed points in the direction of the Jacobian eigenvalues with real parts exceeding zero.

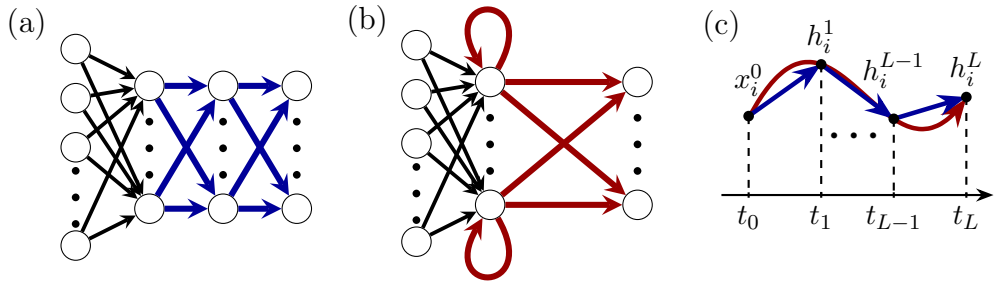

Figure 1.2: Feedforward vs Recurrent. (a–b) Schematic information flow of feedforward and recurrent types. The black arrows between the input and hidden layer typically refer to a linear (affine) transformation followed by a non-linear activation. The red arrows between the hidden and output layers correspond to a continuous-time recurrent evolution as in Eq. 1.4 whereas the blue arrows are similar to that of the black arrows (Eq. 1.1). (c) States of the i^{th} neuron from the hidden layers across the depth of a feedforward network are discretely indexed (blue lines), while recurrent neural network states are continuous in time (red curve).

Conversely, when $g < 1$, the system exhibits an ordered phase, where fixed-point activity converges to zero.

The interplay between discrete and continuous dynamics has received growing interest due to large depths and residual network (ResNet) skip-connections being implemented in networks [17, 18]. For our case, when the bias term in the MLP (Eq. 1.1) is set to zero, it can be viewed as an explicit Euler discretization of the rate-based model above, and the layers across the depth of the network in the former case can now be viewed as a time variable as presented in Fig. 1.2(c). Without coincidence, Gaussian MLPs with zero biases and the same hyperbolic tangent activation demonstrate equivalence to rate-based models in terms of fixed point classification via $\sigma_w \equiv g$. This is due to the same structure of preactivation layerwise Jacobians $\partial h^l / \partial h^{l-1} = W^{l+1} D^l$ where $D^l_{ij} = \phi'\left(h^l_i\right) \delta_{ij}$. Assuming that the input x^0 is chosen that $q^1 = q^\infty$ (Eq. 1.2), then the dynamics begin at the fixed point and the distribution of D^l becomes independent of l, the EOC in Eq. 1.3 can equivalently be expressed as

$$\chi_1 = \frac{1}{N} \left\langle \mathrm{Tr}(DW)^\top DW \right\rangle , \tag{1.5}$$

directly linking criticality to the Jacobians [19]. As a result, MLPs and rate-based models are unified through their shared Jacobian structure when governed by the same framework of random Gaussian connectivity weights.

1.1.3 Heterogeneous weights and extended criticality

Modern applications of DNNs demand heavy computational resources for the learning process, especially for investigations of new scientific domains. On the other hand, training networks from scratch has become less desirable for well-established applications such as image recognition. Transfer learning effectively mitigates this issue, i.e. fine-tuning pretrained networks, models successfully trained on large datasets, on the downstream task is an efficient approach [20]. In deep learning, strong evidence has been suggested against Gaussian randomness induced by the process where the gradients of the loss are found to be heavy-tailed; we defer a detailed discussion on learning to Section 1.4. Similarly, statistical analysis on these large pretrained networks shows that their model weights are generally non-Gaussian, and instead display heterogeneity in these connectivities as shown in Fig. 1.3(a). The distribution of individual weight matrices either fully-connected or convolutional has been found to display (truncated) power-law properties that possess infinite second moments. Furthermore, these power-law properties are also exhibited in the singular value spectrum of the fully-connected or convolutional weight matrix (Fig. 1.3(b)), and the corresponding heavy-tailedness demonstrates strong correlation with the performance of the model [13]. From the perspective of heavy-tailed universality in statistical physics (Fig. 1.3(c)), heavy-tailed singular spectrum generally originates from either strong correlations between entries or random heterogeneity in the coupling strengths. This concept of universality arises in systems with strong correlations, particularly near a critical point or phase transition. It is identified by certain experimental *observables* that display heavy-tailed behavior [12]. In our study, we explore this phenomenon under the assumption of heavy-tailed random connectivity strengths (blue arrow in Fig. 1.3). The analytical tools designed for Gaussian networks become impractical in this context, necessitating further advancements.

Based on recent developments from heavy-tailed rate-based models (Eq. 1.4), a notion of Anderson criticality has been derived for non-Gaussian cases where an extended region is unveiled for heteogeneous connectivity [23]. Due to similar components of the connectivity, the non-linear activation and in particular shared Jacobian structure with MLPs, the same analysis can be performed by utilizing heavy-tailed non-Hermitian random matrix. By characterizing the averaged Jacobian eigenvalues, we unveil a similar type of extended criticality-like phase transition where the range of σ_w for information propagation expands; this phase transition also remains consistent with the homogeneous case in Fig. 1.1. The phenomenon of extended criticality has been observed in graph networks with symmetric adjacency matrices [24, 25]. In contrast, infinite-width Gaussian networks are optimal only at the edge of chaos (EOC) (see Section 1.1.1), a condition that is typically not maintained after training. Therefore, the study of random

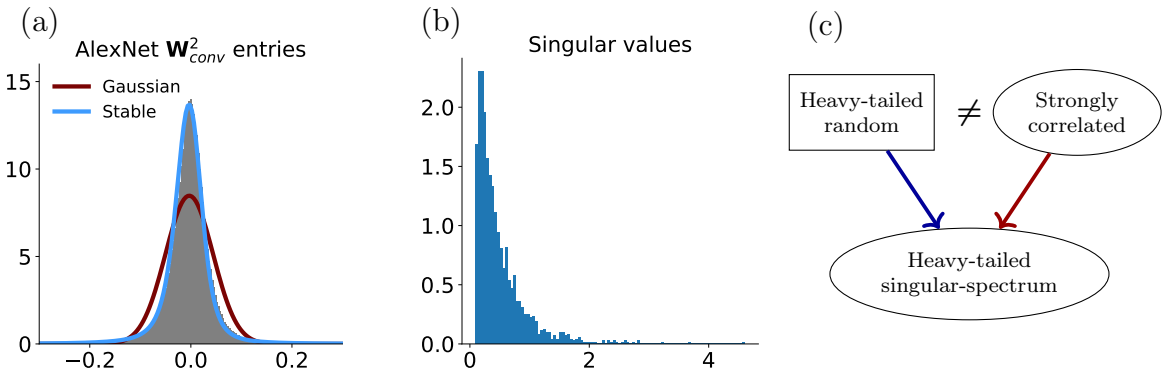

Figure 1.3: Statistics of pretrained network weights. (a) Fitting entries of the second convolution weight matrix $\mathbf{W}^2_{\text{conv}}$ of AlexNet [2] pretrained on the dataset ImageNet1K [5] to a Gaussian distribution and a Lévy α-stable distribution (fitted $\alpha \approx 1.46$). See Chapter 2 for more fitted results. (b) The singular value distribution of the convolution tensor in (a) show power-law decay. The singular value decomposition (SVD) is performed as in [13], similar results were obtained herein. (c) Heavy-tailed Universality in statistical physics [21, 22]. Both heavy-tailed random weights and strong correlations within the weights can result in a heavy-tailed singular spectrum, though these phenomena are not equivalent. In this work, we focus primarily on the former.

heavy-tailed networks offers a compelling theoretical basis for understanding pre-trained models, and we further elaborate on their dynamics under this statistical framework in Chapter 2.

1.2 Stochastic dynamical systems

In the previous sections, we have discussed models at initialization including FFNN (Eq. 1.1) and the rate-based model (Eq. 1.4) where the randomness is quenched and the focus of interest revolves around fixed points of the system. An extra layer of complexity is introduced when the underlying randomness also evolves with respect to time. For example during each training step, the network parameters are influenced by noise terms that vary in nature over time. In neurobiological modeling, a more realistic extension of the rate-based model would involve adopting a dynamic approach, where the total synaptic current accounts for random fluctuations in the second term of Eq. 1.4 [26]. Other phenomena like visual-spatial attention sampling can no longer be accurately described by an ODE due to (time-dependent) random perturbations involved [27]. In these cases, the complexity extends beyond static random variables and leads us to the following.

Stochastic processes are the class of random variables indexed by time, they provide a starting point to model phenomena where random perturbations are induced temporally. Under suitable assumptions, dynamics of the aforementioned examples can be formulated by a stochastic differential equation (SDE):

$$\mathrm{d}X_t = \mu(X_t, t)\mathrm{d}t + \sigma(X_t, t)\mathrm{d}Z_t, \quad X_0 = x. \tag{1.6}$$

Here $\mu : \mathbb{R}^d \times \mathbb{R}_0^+ \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \times \mathbb{R}_0^+ \to \mathbb{R}^{d\times d}$ are the drift and diffusion terms. The intuition behind this equation originates from perturbing an ODE by a diffusion process encapsulated in the second term. The choice for Z is effectively unlimited. However, to allow for a well-defined and rigorous construction with a probabilistic point of view such as the Itô construction of Eq. 1.6, it requires Z_t to be a semimartingale [28], which includes the class of Lévy processes. These processes sufficiently capture the complexity of the noise whilst maintaining the mathematical formulation's well-posedness due to their stationary and independent increments. The the seminal *Lévy-Khitchine formula* [29] reveals that these processes are classified by the characteristic function $\mathbb{E}\left(e^{i(u,X(t))}\right) = e^{t\eta(u)}$ with symbol $\eta(u)$ that satisfies, for all $t \geq 0$ and $u \in \mathbb{R}^d$,

$$\eta(u) = i(b, u) - \frac{1}{2}(u, au) + \int_{\mathbb{R}^d - \{0\}} \left[e^{i(u,y)} - 1 - i(u, y)\chi_{0<|y|<1}(y)\right] \nu(dy). \tag{1.7}$$

Here $b \in \mathbb{R}^d$, a is a positive definite symmetric $d \times d$ matrix and the Lévy measure ν supported on $\mathbb{R}^d \backslash \{0\}$ satisfies $\int_{\mathbb{R}^d\backslash\{0\}} \min\{1, |y|^2\}\, \nu(dy) < \infty$. The corresponding

stochastic process enjoys the celebrated *Lévy-Itô* decomposition

$$X(t) = bt + \sqrt{a}B(t) + \int_{0<|x|<1} x[N(t,dx) - tv(dx)] + \int_{|x|\geq 1} xN(t,dx). \tag{1.8}$$

The independent increments of Lévy processes naturally induce the Markovian property, making the underlying semigroups defined by $T_t\psi(x) := \mathbb{E}[\psi(X_t)|X_0 = x]$ and the infinitesimal generator

$$\mathcal{H}\psi = \lim_{t\downarrow 0} \frac{T_t\psi - \psi}{t} \tag{1.9}$$

very important as it encodes information on the probability density evolution of the underlying process. For instance, the most studied case is when Z is chosen as the Brownian motion W_t where $b = 0$, $a = 1$, and $\nu(\mathbb{R}^d) = 0$. Through the Feynman-Kac formulation, this probabilistic evolution is also encapsulated by a deterministic partial differential equation (PDE) known as the *Fokker-Planck equation*

$$\frac{\partial \rho}{\partial t}(x,t) + \mathcal{H}\rho(x,t) = 0, \quad \mathcal{H}\rho(x,t) = \sum_{i=1}^{d} \mu_i(x,t)\frac{\partial u}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \gamma_{ij}(x,t)\frac{\partial^2 u}{\partial x_i \partial x_j} \tag{1.10}$$

where $\gamma_{ij}(x,t) = \sum_{k=1}^{d} \sigma_{ik}(x,t)\sigma_{jk}(x,t)$ and ρ is the probability density function of the underlying process X_t. In the pure diffusion case where the drift μ is zero in Eq. 1.6, the heat equation is recovered in Eq. 1.10, demonstrating deep connections between stochastic processes and parabolic PDEs. Here, we have provided an overview of multi-dimensional SDEs which remain as powerful tool for modeling in both deep learning and neuroscience. The underlying space can be further generalized to non-Euclidean manifolds such as $\mathbb{S}^{d-1}$ as we will discuss further in Section 1.3.

In several experimental findings, the observations do not match Gaussianity as the conventional literature has assumed. Recent works have shown modeling Z as a Lévy-alpha stable process provides a more accurate depiction of the total synaptic current in a heterogeneous network model [26, 30, 31]. This class of stochastic process is defined by a stability parameter α that controls the heavy-tailedness of each step size through time. In particular, jump sizes for infinitesimal time steps Δt are sampled from the distribution symmetric α stable ($S\alpha S$) distribution, i.e. $\Delta x := Z_{t+\Delta t} - Z_t \sim S_\alpha((\Delta t)^{1/\alpha})$ [32], which asymptotically possess a power-law $p(\Delta x) \sim c_\alpha \Delta t |\Delta x|^{-1-\alpha}$ where $c_\alpha := \Gamma(1+\alpha)\sin(\pi\alpha/2)/\pi$ is a normalization constant. Correspondingly, the symbol of the characteristic function and generator take form

$$\eta(u) = |u|^\alpha, \ \mathcal{H} \equiv -(-\Delta)^{\alpha/2}, \tag{1.11}$$

for $\alpha \in (0, 2)$. A non-local operator *Fractional Laplacian* in turn becomes the corresponding infinitesimal generator of the process, and the classical Laplacian is recovered when $\alpha \to 2$ [33]. Such properties have been observed in many neurobiological systems, including the membrane potential of spiking neural networks [26] and visual spatial sampling in neural circuits [30, 31], we thus incorporate these conditions respectively in the weights of feed-forward networks and the dynamics of self-attention in Transformers (Section 1.3). Additionally, the learning dynamics of deep networks have been modeled by SDEs driven by Lévy processes [34, 15], matching the anomalous diffusion processes empirically found in Chapter 4.

1.3 Mechanisms of sequential processing

Feedforward networks, as discussed earlier, have proven to be highly effective for processing spatial structures like images. Meanwhile, recurrent neural networks (RNNs) have long been the preferred architecture for sequence processing [35]. Sequential data types with intrinsic temporal order commonly appear in speeches, time series, particle diffusion, etc. While some sequential dynamics, like planetary motion, can be derived from first principles of physics, other sequences, such as speech and time series data, are often stochastic and influenced by random noise, making their underlying mechanisms highly complex. Moreover, the corresponding data processing objective is user-dependent – ranging from classification to next-token-prediction. The former involves an encoding component and the latter requires an additional decoding structure that integrates an auto-regression nature. The significance of these applications and the difficulty in handling with these data structures, make this a critical field of study.

Growing interactions between computers and human language have triggered the necessity for programming machines to process and analyze natural language data, hence natural language processing (NLP) has undergone rapid development. Translational tasks, sentiment analysis, topic identification, and text summarization remain as the core applications of NLP. Before any attempts at modeling, the intrinsic semantic component of human language must be represented in a meaningful manner to models. This requires text to be digitalized in a certain format and the state-of-the-art approach relies on *token embedding*. The modern framework *Word2Vec* embeds words into a high-dimensional Euclidean space using unsupervised learning techniques [36]. This approach captures semantic relationships, ensuring that words with similar meanings or those frequently co-occurring in sentences are mapped to nearby points in the embedding space (Fig. 1.4(a)). For a θ-parametrized RNN f_θ, these numeric representations are fed in an autoregressive fashion $(x_t, h_t) \to (y_t, h_{t+1})$ where x_t, h_t and y_t are the input, hidden and output vectors respectively. Here, the hidden states of h_t display parallelisms to *memory*

of neural systems, recording partial information from the input-output pair up to time t. The hidden states are updated over time for improved downstream tasks. Multiple RNN blocks can be aggregated to construct a deep/stacked RNN. For long-sequence processing, past information becomes more difficult to retain within h_t, and through the incorporation of *forget gates*, an updated model Long-short term memory (LSTM) is thus proposed. The model is nonetheless prone to gradient explosion issues during learning [37]. Moreover, LSTM and its variants such as Gated recurrent unit (GRU) [38] suffer from long dependencies despite their improvements, the lack of parallelizability also results in slower processing times.

1.3.1 Self-attention

Recently, an alternative mechanism for sequential processing has taken inspiration from a well-known neuroscience concept – *Attention*, defined by the concentration of awareness on specific phenomena while excluding other stimuli. Analogously, attention in the machine learning context refers to the process of determining the relative importance of each component relative to the others in a sequence. The groundbreaking *Transformer* model guided by attention has emerged as a predominant architecture in deep learning [39]. Contrary to common belief, this model consists of solely simple feed-forward layers such as fully-connected (or convolutional) blocks, non-linear activations, LayerNorms, etc without the need of any recurrent information flow. Transformers have attained benchmarks far surpassing their predecessors and with minimal modifications, the architecture has been successfully extended to other domains such as visual processing in which outstanding results have also been attained [40].

We begin by briefly describing the mechanism of encoding in Transformers which relies on the self-attention is also known as the *Standard Scaled Dot-Product Attention*. In a NLP supervised learning setting such as classification, given a sentence where each word is referred to as a token and indexed according to a pre-defined dictionary, the index automatically maps each token onto a high-dimensional word embedding which consists of trainable weights; positional embeddings share a similar concept and are aggregated into the word embeddings, typically through summation to obtain the overall embeddings X. As outlined in Fig. 1.4(b), the embeddings $X := (x_1, x_2, \cdots, x_n) \in \mathbb{R}^{d\times n}$ are then used to

$$\text{Attention}(Q, K, V) = AV^\top \in \mathbb{R}^{n\times d_v};\ A = \text{softmax}\left(\frac{Q^\top K}{\sqrt{d_k}}\right) \tag{1.12}$$

to create the queries, keys and values $Q = W_Q X$, $K = W_K X$, $V = W_V X$ from linear projections of the embeddings based on $W_Q \in \mathbb{R}^{d_k\times d}$, $W_K \in \mathbb{R}^{d_k\times d}$ and $W_V \in \mathbb{R}^{d_v\times d}$ respectively. The attention weights A is the row normalization of the

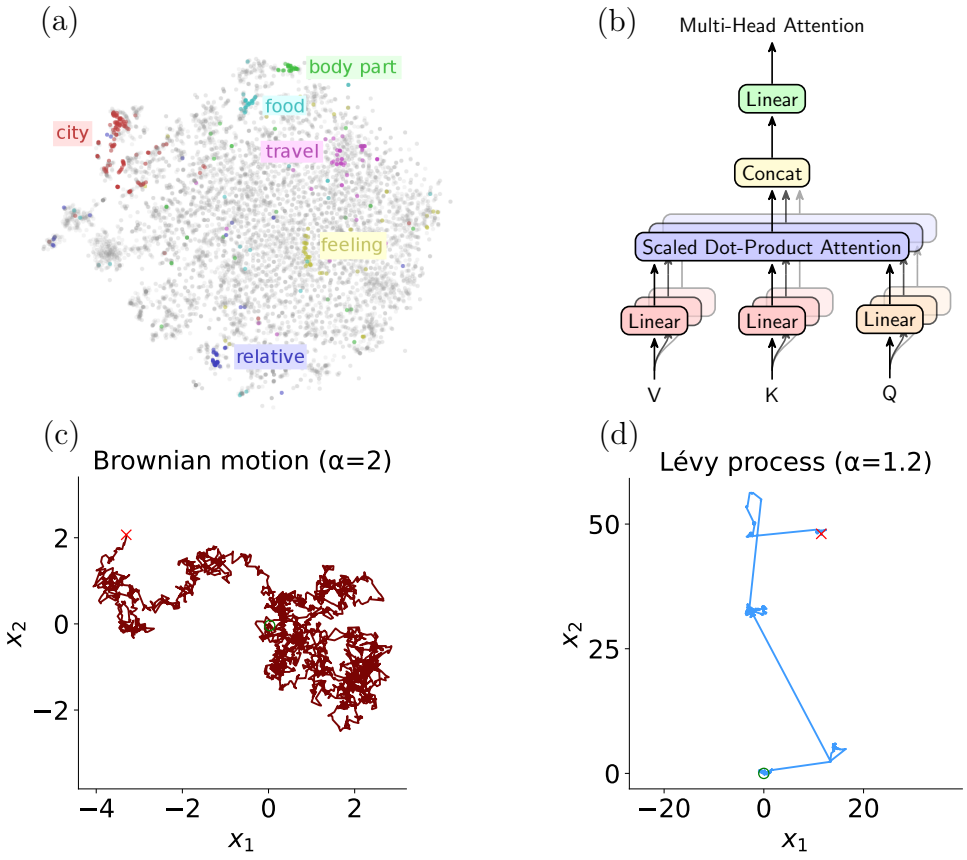

Figure 1.4: Elements of Transformers. (a) The Word2Vec embeddings projected on $\mathbb{R}^2$ [41]. (b) The Multi-head Attention block as described in Eq. 1.12. The shadows of the queries, keys and values along with the scaled dot-product attention are for multiple attention-heads of size $H = 3$. During the encoding process, Q, K and V all share the same embedding inputs; for decoding, inputs of K, V are different to Q. (c) Brownian motion in $\mathbb{R}^2$. (d) Lévy process ($\alpha = 1.2$) in $\mathbb{R}^2$.

graph weights

$$\tilde{C}_{ij} = \exp(x_i^\top W_Q^\top W_K x_j / \sqrt{d_k}). \tag{1.13}$$

and it transforms the values to the output of the self-attention block (Fig. 1.4(b)). For general NLP tasks such as neural machine translation where both a source X and target sequence X' under a different language are present, this necessitates a decoding component, an additional *cross-attention* layer. In particular, the encoded outputs of X would be used to generate the keys and values for the cross-attention, whereas X' is used to construct Q [39], highlighted in the different color shades for Q and K, V in Fig. 1.4(b). Our focus in the following is on the encoding component of self-attention, which is central to deriving the corresponding first- and second-order dynamics.

Transformers process all sequence components in parallel during training, enabling efficient matrix multiplications and exceptional scalability for handling long sequences. Pretrained models have been shown to display remarkable properties such as turing-completeness [42]. However, they suffer from many shortcomings as a general input-output mapping. In particular, the self-attention mapping is not Lipschitz, and consequently, the geometry of embeddings are not well-preserved throughout layers, i.e. two non-overlapping tokens would progressively become more similar across layers. Hence, other types of similarity measures such as the L^2 distance have replaced the original demonstrating more desirable properties [43]. Furthermore, as computation scales in the self-attention block quadratically ($\mathcal{O}(n^2)$) in the sequence length n, large efforts have been devoted to reducing this computational complexity whilst minimizing the reduction in model expressivity. Despite its roots in neuroscience, relatively little focus has been allocated to the neurobiological underpinnings of the self-attention mechanism. In the following, we explore how fractional diffusion-based sampling methods can be integrated into self-attention, offering new perspectives on the model's structure and efficiency.

1.3.2 Attention, Operators, and Diffusions

A core element of the self-attention mapping is the residual network (ResNet) skip-connection, as it prevents the rank-collapse [44]. Similar to Fig. 1.2(c), this provides a continuous-time view of the mapping, without loss of generality, let us focus on the single-head case, alternatively expressing Eq. 1.12 as

$$x_i^{l+1} = x_i^l + \varepsilon \sum_{j=1}^{n} K_{ij} W_V x_j^l \tag{1.14}$$

where $\varepsilon = 1$ is the default step size implementation and $K := \text{softmax}(C)$ is obtained in a row-wise fashion. Entries of C are the pairwise dot-products between queries and keys $C_{ij} := (q_i)^\top k_j = x_i^\top W_Q^\top W_K x_j$, both W_Q and W_K are trainable

weights which undergo training to yield appropriate attentiveness between all tokens. Note that Eq. 1.14 is a specific realization of $x_i^{l+1} = x_i^l + T(x_i^l)$ that corresponds to an Euler discretization with step-size 1 of the ODE $\dot{x}_i = T_\mu(x_i)$ for all i; the measure μ applies to the distribution of x_i's. A recently developed architecture which applies the *Sinkhorn* algorithm iteratively approaching a bi-normalized transition matrix in Eq. 1.14 has second-order dynamics represented by the classical heat diffusion $\partial_t \rho = \Delta \rho$ underlying the transition of the values [6]. Equivalently, the same diffusion type can be established through the Gaussian kernel by replacing the dot-product with the L^2 distance

$$\tilde{C}_{ij} = \exp\left(|x_i - x_j|^2 / 2\varepsilon\right) \tag{1.15}$$

where $|\cdot|$ is the Euclidean distance in the embedding space $\mathbb{R}^m$ and $\varepsilon > 0$ is the kernel bandwidth. Here, we have made extra assumptions that $W_Q \in O(d)$ is orthogonal and $W_Q = W_K$ which are commonly assumed [43]. The (row) degree matrix defined by $D_{ii} := \sum_{j=1}^n \tilde{C}_{ij}$ allows one to arrive at the graph Laplacian $L = D^{-1}\tilde{C} - I$. In the limit of infinite data points, the graph Laplacian manifests a continuous counterpart where the convergence is pointwise [45]

$$\frac{1}{\varepsilon} \lim_{n\to\infty} \sum_{j=1}^{N} L_{ij} \rho(\boldsymbol{x}_j) = \frac{1}{2} \Delta \rho\left(\boldsymbol{x}_i\right) + O\left(\varepsilon^{1/2}\right). \tag{1.16}$$

for a smooth function $\rho : \mathbb{R}^d \to \mathbb{R}$. In infinite sequence ($n \to \infty$) and infinitesimal step size ($\varepsilon \to 0$) limits, both of the above cases represent the infinitesimal generator of a Brownian motion on the manifold $\mathcal{M}$ (Fig. 1.4(c)). Motivated by a *Fractional Neural Sampling* (FNS) theory of spatiotemporal probabilistic computations in neural circuits, we implement a partial realization of with only the diffusion component in self-attention. Through the seminal *Heat Kernel Dichotomy*, we use a counterpart of Eq. 1.15 that extends to a wider class of kernels that have polynomial decay [46]. With appropriate assumptions, we implement the notion of fractional diffusion in the self-attention mapping, enabling the occurrence of long-range interactions in the embedding space. These effects correspond to a discretized counterpart reminiscent of the Lévy process (Fig. 1.4(d)), with the infinitesimal generator yielding the fractional Laplacian (Eq. 1.11) discussed above. This construction also allows for a canonical decomposition of the attention weights K, providing an interpretable view in low-dimensional spaces through the lens of diffusion maps [47, 48].

1.4 Dynamics of stochastic gradient descent

Training algorithms for DNNs are the backbone of deep learning applications. *Stochastic gradient descent* (SGD) has remained the state-of-the-art method due

to its effectiveness and efficiency. The objective function in deep learning can take two forms: a loss function measuring the discrepancy between network predictions and ground truth in supervised learning, or a reward function assessing an agent's performance in reinforcement learning. In the latter case, the loss can be framed as a negative reward. However, our focus will be on the former, which is typical for supervised learning tasks. The loss forms a highly complex landscape with dimensions each corresponding to learnable parameters of the model. Stochasticity introduced through random subsampling on the dataset at each time step has been shown to reveal a non-trivial structure, often non-Gaussian [15]. In Chapter 4, we quantify characteristics of SGD over short and long timescales and show *anomalous diffusive* in various network settings. In the following Appendix 4.A as ongoing work, we propose and outline a novel mathematical framework that models the learning process as a matrix-valued stochastic system, offering deeper insights into the evolution of the model's parameters. We also discuss the implications of this approach and its potential to enhance our understanding of deep learning dynamics.

1.4.1 Anomalous diffusion dynamics

Let us consider a supervised learning setting where a labeled training dataset $\mathcal{D} := \{(\mathbf{x}^i, y^i)\}_{i=1}^{D_1}$ with D training samples is present. Given some network NN_Θ parametrized by a set of trainable weights Θ (represented as a flattened vector), we define the network prediction as $\hat{y}^i = \mathrm{NN}_\Theta(x^i)$. As a first-order optimization method, SGD minimizes the empirical loss averaged L across the dataset

$$L(\Theta) := \frac{1}{D}\sum_{i=1}^{D} f_\Theta(\hat{y}^i, y^i) \tag{1.17}$$

To avoid an instantaneous computation flux based on the entire dataset $\mathcal{D}$, L is is approximated by $\tilde{L}_k := |B_k|^{-1}\sum_{i\in B_k} f_\Theta(\hat{y}^i, y^i)$. At each iteration k, the data points of $|B_k| := b\ (< D)$ are arbitrarily subsampled from $\mathcal{D}$ known as minibatches. Under SGD, the weights undergo an iterative scheme

$$\begin{aligned}\Theta_{k+1} &= \Theta_k - \eta_k \nabla \tilde{L}_k\left(\Theta_k\right), \\ &= \Theta_k - \eta_k \nabla L\left(\Theta_k\right) + Z_k\end{aligned} \tag{1.18}$$

where $Z_k = \eta_k(\nabla L\left(\Theta_k\right) - \nabla \tilde{L}_k\left(\Theta_k\right))$ is the source of the minibatch noise and adaptive learning rate schemes for η_k are also available. Additional terms such as momentum or adaptive step sizes can also be introduced for accelerated learning [49, 50, 51, 52]. The high-dimensional loss (Eq. 1.17) is generally non-convex [53, 54] and filled with local minima and saddle points that have bad generalization [55],

i.e. overfitting to the training dataset whilst yielding significantly worse evaluation performance. Not only gradient descent is computationally inefficient, but it also frequently leads the optimizer to a sub-optimal solution.

The stochasticity induced by the minibatch noise in Eq. 1.18 is considered beneficial, as the randomness provides the optimizer with a higher probability to *escape* from bad local minima. Analysis on the trajectory of the learning process through the time-average mean-squared displacement (TAMSD) unveils non-equilibirum dynamics which are *anomalous diffusive*. In particular, subdiffusive processes emerge early during training whilst superdiffusion occur at longer time-scales, displaying small movements that are intermittently interrupted by long-range jumps.

1.4.2 Continuous-time modeling

The realization of first-order training algorithms for deep neural networks operates in iterative time steps (Eq. 1.18) which can be modeled by discrete-time stochastic processes [56]. Under certain conditions, in the limit of infinitesimal learning rates $\eta_k \to 0$, the dynamics can be approximated by SDEs in the form of Eq. 1.6 [57, 58]. Additionally, provided that the second moment of the random noise Z_k remains finite, the stochastic process governing these dynamics can be modeled as Brownian motion. Similar to the description in Eq. 1.18, almost all studies have *vectorized* or *flattened* the parameters into a vector, ignoring the matrix structure. As an increasing amount of work has applied random matrix theory (RMT) in comprehending the singular values of the weight matrices [13], or furthermore the eigenvalues of the Jacobian matrices [59], these quantities during learning are of great significance. Therefore, we combine both aforementioned tools of stochastic analysis and RMT to motivate a more complete description of deep learning, where we briefly discuss the framework of modeling the weights as a matrix-value diffusion process driven by a Ginibre ensemble [60], that in turn elicits the eigenvalue dynamics during training.

Chapter 2

Dynamical and computational properties of heavy-tailed deep neural networks

Abstract

In deep neural networks (DNNs), the assumption of Gaussian statistics in theoretical studies has been pervasive. However, empirical data has shown that heavy-tailed connectivity is prevalent in commonly used pretrained DNNs. To elucidate the fundamental impact of such heavy-tailed connectivity on the dynamical and computational properties of DNNs, we develop a novel mean field theory that integrates theories of heavy-tailed random matrices and non-equilibrium statistical physics. Our theoretical framework demonstrates that heavy-tailed weights enable the emergence of the extended criticality (edge-of-chaos) in a broad parameter region, leading to improved and faster learning of real-world tasks without the need to fine-tune the weight statistics, compared to networks with Gaussian weights. Notably, we find that heavy-tailed DNNs exhibit multifractal eigenvectors and internal neural representations of input data. Overall, our findings challenge the commonly held assumption of Gaussian statistics in DNNs and offer new insights into the underlying mechanisms of deep learning.

2.1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in a wide range of applications including visual object classification [2], natural language processing [3], speech recognition [61] and advancing general scientific research [4]. These systems pass input data through numerous hidden layers composed of neurons,

resulting in complex activity dynamics to generate highly abstract representations necessary for real-world classification tasks. Understanding the origin of such complex dynamics in DNNs and their computation roles is thus a longstanding topic of interest in artificial intelligence and has implications for gaining insight into similar complex systems such as the brain [62]. Since the deep learning revolution of the past decade [63], a classical theoretical framework has emerged using a variety of physical and mathematical tools including statistical physics as well as random matrix theory in order to understand the dynamics of signal propagation through deep networks [64, 65]. Such works have made use of Gaussian mean-field approaches to establish parameter phases of vanishing and exploding signal propagation [9, 19], corresponding to the classical phases of order and chaos respectively [16]. These important insights have informed initialization strategies which lead to the faster and more successful training of deep neural networks, such as concentrating the singular values of the Jacobian matrix across each layer around 1 [66].

A majority of theoretical studies have generally assumed Gaussian random neural networks or Gaussian initialization schemes. However, empirical evidence suggests that during the training process, deep networks invariably exhibit heterogeneous, heavy-tailed weights, independent of their specific optimization algorithm or architecture [34, 56]. Additionally, sufficiently complex and well-trained DNNs exhibit heavy-tailed mechanistic universality: the singular value spectra of large weight matrices display a powerlaw whose tail exponent is negatively correlated with generalization performance [13]. These empirical observations and the prevailing theoretical practices raise fundamental questions about the impact of heavy-tailed heterogeneity on the dynamical and computational properties of DNNs. Addressing these questions would not only enable us to gain a better understanding of the functional mechanisms of DNNs, but would also inform the design of future systems.

Here, we develop a new theory for DNNs with random heavy-tailed connectivity based on Lévy mean-field theory [26] and non-Hermitian random matrix techniques [67]. We use the theory to map out an extended critical phase where DNNs balance order and disorder in a broad region of parameter space, allowing for deep information propagation. Fine-tuning or handcrafted inductive biases is thus no longer required as predicted by conventional theory. While our theory primarily focuses on random fully-connected neural network models, also known as multilayer perceptrons (MLPs), in the infinite width limit, we have also conducted empirical validation to confirm the presence of extended criticality in other network architectures, such as convolutional neural networks (CNNs). Notably, initializing DNNs in the extended criticality regime enhances generalization and accelerates training, thus emerging as a potent initialization method that challenges the conventional practice of employing homogeneous initial weights.

Our theory also reveals the emergence of multifractality in the eigenvectors of the input-output layerwise Jacobians and principal directions of neural representations. Multifractality is a phenomenon characterized by a spectrum of infinitely many exponents that govern the scaling properties of a system [68]. In the context of DNNs, multifractality in the critical phase is distinct from the classical Gaussian network's edge of chaos [16, 9]. It blends aspects of both localization and delocalization, resulting in robust neural representations with lower effective dimension for fully-connected DNNs. These properties persist in large pretrained networks.

Heavy-tailed connectivity has been previously observed in biological neural networks [69], suggesting that our theory of extended criticality may have general applicability to understanding neural systems. This hypothesis thus highlights the potential of multifractality and extended criticality as general governing principles of both artificial and biological intelligence.

2.2 Heavy-tailed statistics in pretrained networks

2.2.1 Distributions of pretrained weights

It has recently been shown that across a wide variety of architectures, the singular spectra of weight matrices in pretrained DNNs are heavy-tailed (i.e. they have power-law tails) [13]. Such heavy-tailed statistics could arise from correlations within the weight matrices, or from the weight entries themselves originating from a heavy-tailed distribution. We investigate the latter hypothesis for pretrained network weight matrices from the PyTorch and TensorFlow libraries.

We individually fit all the entries of each weight matrix $\mathbf{W}^l$ or convolution tensor $\mathbf{W}^l_{\text{conv}}$ (excluding the biases and batch-normalization layers as in [13]) between the $(l-1)^{\text{th}}$ and l^{th} layers individually as a Lévy α-stable distribution (often termed *stable distribution*) [70], which is defined by a characteristic function involving a tuple $(\alpha, \beta, \sigma, \mu)$ containing the stable, skewness, scale and location parameters respectively,

$$\varphi(u; \alpha, \beta, \sigma, \mu) = e^{-|\sigma u|^\alpha (1 - i\beta \text{sgn}(u) \Phi(u;\alpha)) + iu\mu} \tag{2.1}$$

where $\text{sgn}(u)$ is the sign of u and

$$\Phi(u;\alpha) = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right) & \alpha \neq 1 \\ -\frac{2}{\pi}\log|u| & \alpha = 1 \end{cases}.$$

If a random variable X is drawn from a stable distribution, we denote it by $X \sim S_\alpha(\beta, \sigma, \mu)$, or $X \sim S_\alpha(\sigma)$ if it is symmetric with $\beta = \mu = 0$ [32]. Apart from Gaussian distributions for which $\alpha = 2$, all stable distributions with $\alpha < 2$ possess power-law tails, which is why α is also referred to as the tail index; the scale parameter σ acts as a generalized notion of standard deviation.

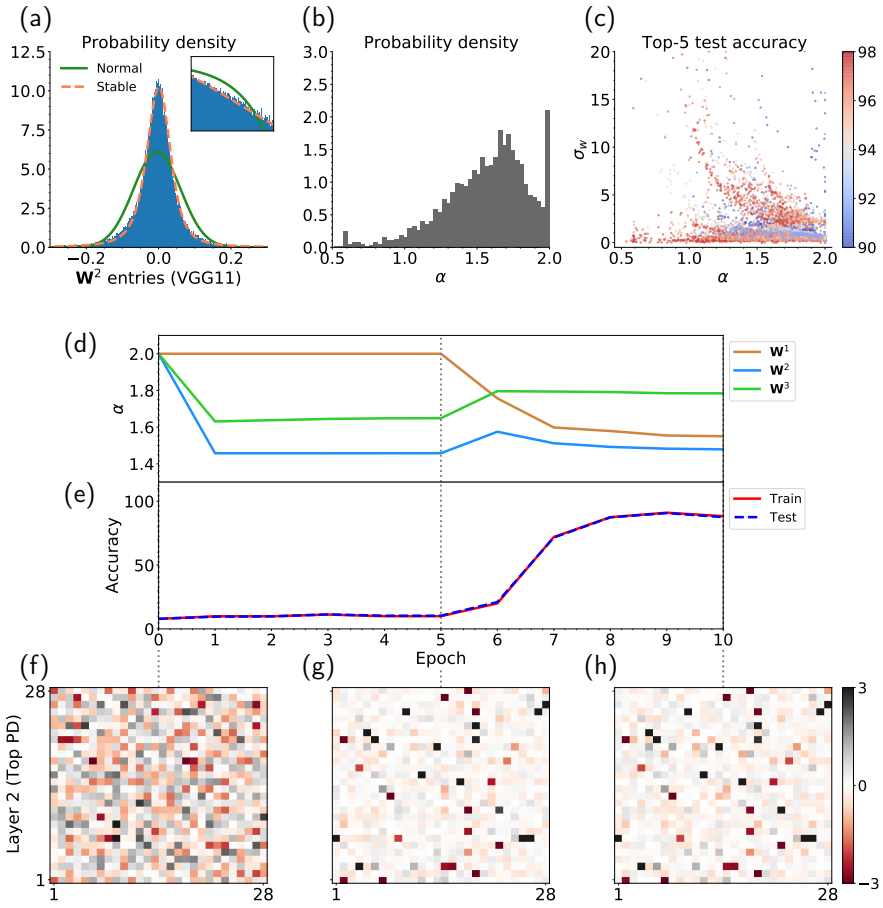

Figure 2.1: Distribution of weight matrix entries. (a–c) Results based on pretrained networks from PyTorch and TensorFlow libraries. (a) All entries of a selected $\mathbf{W}^2_{\text{conv}}$ (over 7.37×10^4 entries) from a pretrained VGG11 fitted to both stable (solid green curve) and Gaussian (dashed orange curve) distributions via maximum likelihood respectively; the stability parameter α is estimated to be 1.38. Inset zooms in on the right tail of the distribution (log-log plot). (b) The distribution of the stability parameter α (with a lower cutoff 0.5) corresponding to each of 5568 weight matrices/convolution tensors from 53 networks, each pretrained on ImageNet1K. (c) Scatter plot of the fitted (α, σ_w) for each pretrained network weight tensor in (b). The colorbar represents the Top-5 test accuracy on ImageNet1K of the network corresponding to the tensor. (d–h) Results based on training FC4 (fully-connected DNN with 4 hidden layers). (d) The evolution of the stability parameter α of FC4 throughout 10 epochs of training for FC4 with the default PyTorch uniform initialization for the fully-connected weight matrices $\mathbf{W}^1$ (orange), $\mathbf{W}^2$ (green) & $\mathbf{W}^3$ (blue). Distribution fitting is performed on the weights at the end of each epoch. Dotted grey lines indicate the epochs of the spatial structures in (f–h). (e) Top-1 train (solid red curve) and test (dashed blue curve) accuracy of FC4 recorded at the final step of each epoch. Dotted grey lines are the same as in (d). (f–h) The top principal direction (PD) of the second preactivation hidden layer corresponding to the output from $\mathbf{W}^2$ (originally 784 reshaped to 28 by 28) at epochs 0, 5 and 10 respectively; colormap represents the value of the coordinates.

As an illustration, Fig. 2.1(a) displays the distribution of the entries in a selected convolution weight matrix $\mathbf{W}^2_{\text{conv}}$ between the first and second hidden layers of the pretrained convolutional architecture VGG11. The distribution can be better fitted by a stable distribution (dashed orange curve) with $\alpha < 2$ than a Gaussian distribution (solid green curve). We perform the same analysis to 5568 weight matrices/tensors (Fig. 2.1(b)) across a sum of 53 networks pretrained on ImageNet1K from both the PyTorch and TensorFlow library and plot the distribution of tail indices α (see **Methods** for fitting details); approximately 88.99% of the tail indices are lower than 1.9. The normalized scale parameters σ_w vary considerably, particularly for lower values of α (Fig. 2.1(c)). The importance of the normalized scale parameter lies in its ability to identify the criticality type of heavy-tailed networks, which can be ordered, critical, or chaotic; this will be further discussed in Section 2.3. For fully-connected weight matrices, $\sigma_w := 2\sqrt{N_w N_h}\sigma^\alpha$ where $\mathbf{W} \in \mathbb{R}^{N_h \times N_w}$; the normalized scale parameter for 2D convolution tensors adopts a different form, this is addressed later in Section 2.3.4. The Shapiro-Wilk tests reveal that 99.58% of the weight matrices reject the null hypothesis of Gaussian weights at the significance level of 2.5%. Additionally, over 67.62% of the p-value ratios of the Kolmogorov–Smirnov (KS) test with respect to the maximum likelihood fit to the stable and normal distribution are strictly greater than 1. While a network's architecture is important for its performance, weight matrices (or tensors) that display greater heterogeneity tend to produce networks with better generalization [13]. This indicates a natural preference for heavy-tailed weights, which is also shown in Figure 2.1(c). Although a perfect fit to the stable distribution is hindered by finite-size effects, it is not necessary for our Lévy mean-field analysis because the theory only requires the assumption of a power-law tail in the large network limit.

2.2.2 Emergence of heavy-tailed weights during training

Similarly, we demonstrate that such heavy-tailed weights can emerge naturally during the training process. We train a fully-connected DNN with the default PyTorch setting of uniform initialization. In particular, this network which consists of 4 hidden layers (FC4) is trained on the MNIST dataset [71] via standard stochastic gradient descent (SGD) for 10 epochs (see **Methods** for network and training setting). Using the same fitting method as in Section 2.2.1 above, we monitor the stability parameters of the first 3 weight matrices during the process (Fig. 2.1(d)). The Top-1 train and test accuracies both improve rapidly (Fig. 2.1(e)) whilst the values of α deviate from 2 and diverge to 1.46, 1.65 and 1.78 for $\mathbf{W}^1$, $\mathbf{W}^2$ and $\mathbf{W}^3$ respectively at epoch 10. The spatial structure of the internal representation for the hidden layers concurrently encounter interesting properties. We perform principal component analysis (PCA) to the preactivation hidden layer

at $\mathbf{W}^2$ and plot the corresponding top principal direction reshaped to a 28 by 28 image for different epochs. Prior to training, the weights remain Gaussian due to the uniform initialization scheme applied and as a result, the top principal direction at layer 2 produces a *dense* (delocalized) spatial structure (Fig. 2.1(f)). After only 5 epochs of training, the spatial structure of the neural representations undergoes significant changes and becomes *sparser*, i.e. more localized, as depicted in Fig. 2.1(g), and this trend persists until the end of training, as shown in Fig. 2.1(h). During this instance of training, not only do the weights deviate from Gaussian statistics, the spatial structures of the neural representations simultaneously change over the course. In Section 2.4, we delve deeper into the ability of heavy-tailed networks to produce multifractal spatial structures in their neural representations (Fig. 2.1(g–h)), this is a distinctive feature which sets them apart from Gaussian networks.

Although the dynamics of DNNs during training may vary depending on the specific instance, architecture, learning setting etc., recent studies [34, 30] have used Lévy processes to model the optimization process; the latter article also showed the anomalous diffusion dynamics in learning which does not commonly appear in systems driven by Brownian motion at large time scales. Furthermore, multiplicative noise combined with heavy-tailed structure also improves the capacity for basin hopping and exploration of loss surfaces of deep learning which are in general non-convex [56]. These theoretical and empirical analyses thus have demonstrated that heavy-tailed weight distributions are common in DNNs.

2.3 Signal propagation through heavy-tailed networks

We now examine the signal propagation characteristics in random DNNs with heavy-tailed weights. The investigation of this property holds crucial significance as it has been shown that the precise training of DNNs relies on the ability of information to traverse through them [16]. In the limit of large layer width, certain properties of the network Jacobian become predictable using tools from mean-field theory and non-Hermitian random matrix theory. Thus, through exploiting these techniques for the layerwise Jacobian spectral density, we develop a theoretical probability measure that enables quantification of signal propagation in fully-connected DNNs with random heavy-tailed coupling. We further confirm the validity of this approach through numerical experiments involving the propagation of circular manifolds through random heavy-tailed networks. Based on the construction, we elucidate an extended criticality transition diagram in the (α, σ_w) phase for $\alpha < 2$ which remains consistent with the Gaussian case ($\alpha = 2$). We then show how this leads to superior information propagation, faster and better training, which not only

provides a different perspective on information propagation, but also serves as a new initialization scheme.

We conduct supplementary experiments to showcase conventional CNNs (without skip connections), such as AlexNet, also manifest remarkable training capabilities when initialized in an extended critical state which closely resembles that of fully-connected DNNs.

2.3.1 A feedforward network model with heavy-tailed weights

Consider a wide fully-connected DNN of depth L, so that each layer l has N_l neurons described by a neural activity vector $\mathbf{x}^l$ along with a weight matrix $\mathbf{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$. Without loss of generality, we consider square weight matrices for all layers where $N_{l-1} = N_l := N$ which allows for computation of the layerwise Jacobian eigenvalues and eigenvectors. The propagation dynamics arising from the input $\mathbf{x}^0 \in \mathbb{R}^N$ is given by

$$\mathbf{x}^l = \phi(\mathbf{h}^l) \ , \qquad \mathbf{h}^l = \mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l \ , \tag{2.2}$$

where the preactivation $\mathbf{h}^l$ is the input at layer l; $\mathbf{b}^l$ is a bias vector and ϕ is an element-wise nonlinear activation function. For ϕ, we only consider sigmoidal activation functions, or more generally, functions that are $o(|x|)$; this is to guarantee the finiteness of the α^{th}-moment of the activation hidden length (see Appendix 2.B). Now consider the network's preactivation input-output Jacobian given by

$$\frac{\partial \mathbf{h}^L}{\partial \mathbf{h}^1} = \prod_{l=1}^{L-1} \mathbf{W}^{l+1} \mathbf{D}^l \tag{2.3}$$

where $\mathbf{D}^l$ is a diagonal matrix with entries $D^l_{ij} = \phi'(\mathbf{h}^l_i)\delta_{ij}$. We are interested in understanding the spectral properties of the layerwise Jacobian $\mathbf{W}^{l+1}\mathbf{D}^l$ for heavy-tailed deep networks with randomly initialized weights and biases (often abbreviated as **WD** in the main text). In particular, we take the weights and biases to be drawn i.i.d. from a zero-mean and symmetrical α-stable distribution with scale parameters $\sigma_w^\alpha/2N$ and $\sigma_b^\alpha/2$ respectively, i.e. $W^l_{ij} \sim S_\alpha(\sigma_w/\left(2N\right)^{1/\alpha})$ and $b^l_i \sim S_\alpha(\sigma_b/2^{1/\alpha})$ for all $l = 1, \cdots, L$. The weight connectivities have asymptotic tails

$$p_{W^l_{ij}}(x) \sim \frac{c_\alpha \sigma_w^\alpha}{2N} |x|^{-1-\alpha} \tag{2.4}$$

for the probability density function $p(x)$, where $c_\alpha := \Gamma(1+\alpha)\sin(\pi\alpha/2)/\pi$ and Γ is the Gamma function. Note that the layerwise Jacobian does not explicitly depend on the bias term, unless otherwise stated, we only consider the DNN system described by Eq. 2.2 where the biases are all set to zero.

2.3.2 Extended criticality identified through Jacobian spectrum

The classical Gaussian mean-field approach identifies the transition to chaos by computing the covariance of two inputs as they propagate through the layers of DNNs [16, 9, 72, 19]. In these studies, chaos is characterized by the separation of nearby points as they pass through network layers, with asymptotic expansions yielding depth scales over which information transmission may amplify the magnitude of a single input or decrease the correlation between two different inputs. However, because variances of such inputs become infinite upon propagation by a single layer for heavy-tailed networks, these covariances are no longer guaranteed to be well-defined. For other systems such as recurrent neural networks (RNNs), analyzing the fixed point requires one to examine the spectral radius of the Jacobian matrix, which shares the same form as the layerwise Jacobian in Eq. 2.3 for fully-connected DNNs. The layerwise Jacobians of Gaussian networks have confined spectral radius due to the finitely supported spectral density (Fig. 2.2(a)), allowing one to classify fixed points based on whether the edge of chaos, $\sigma_w = 1$ is crossed. This is no longer the case for heavy-tailed networks as the Jacobian spectral density has infinite support shown in Fig. 2.2(b), any fixed point is deemed chaotic by conventional dynamical system theory [23]. As a result, the issues observed in fully-connected DNNs with discrete layers and RNNs with continuous dynamics attest the necessity of a new theoretical framework to understand the dynamics of heterogeneous networks.

For this reason, we will directly compute the signal propagation properties of the full input-output Jacobian in the stationary state, by observing that the layerwise Jacobians are independent and identically distributed. Thus, diagonalizing the layerwise Jacobians by $\mathbf{W}^{l+1}\mathbf{D}^l = \mathbf{Q}_l^{-1}\mathbf{\Lambda}_l\mathbf{Q}_l$, where the columns of $\mathbf{Q}_l$ are the eigenvectors and the diagonal entries of $\mathbf{\Lambda}_l := \operatorname{diag}_k \lambda_k$ are the eigenvalues, and moving into Schröndiger's picture where the eigenvectors remain constant, the full input-output Jacobian is

$$\prod_{l=1}^{L-1}(\mathbf{W}^{l+1}\mathbf{D}^l) = \prod_{l=1}^{L-1}(\mathbf{Q}_l^{-1}\mathbf{\Lambda}_l\mathbf{Q}_l) = \mathbf{Q}^{-1}\left(\prod_{l=1}^{L-1}\mathbf{\Lambda}_l\right)\mathbf{Q} \tag{2.5}$$

Due to the strong law of large numbers, each diagonal element of this product indexed by k converges to the following magnitude almost surely for sufficiently large depth L:

$$\prod_{l=1}^{L-1}|\lambda_k| = \exp\left(\sum_{l=1}^{L-1}\log|\lambda_k|\right) \xrightarrow{\text{a.s.}} \exp\left((L-1)\int_{\mathbb{C}}\log(|z|)\rho(z)\mathrm{d}z\right) \tag{2.6}$$

where $\rho(z)$ is the eigenvalue density of the layerwise Jacobian. Thus, in the stationary state where the eigenvalues no longer have dependence on l, signal

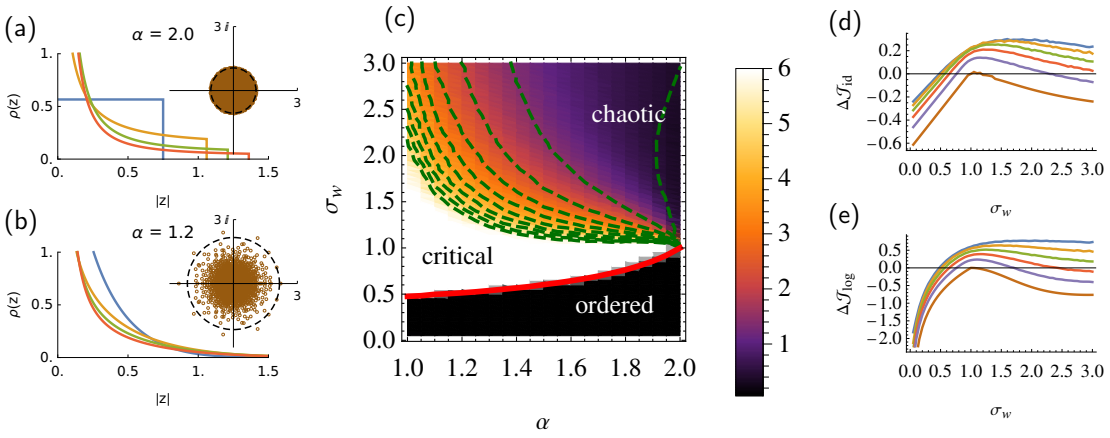

Figure 2.2: Signal propagation measure for the Jacobian spectrum and phase diagram for random deep networks. (a) The density of states of the layerwise Jacobian **WD** for deep random networks with zero bias and $\sigma_w = 0.75, 1.5, 2.25, 3$ (blue, yellow, green, red); $\alpha = 2.0$. Insets for both (a–b) depict the empirical eigenvalue distribution and characteristic spectral radius (dashed circle) beyond which exponential suppression of eigenvalue density dominates for $\sigma_w = 1.5$. Here the characteristic spectral is equivalent to the spectral radius of the Jacobian. (b) Same as in (a) but for $\alpha = 1.2$, the characteristic spectral radius in this case is finite but the spectral radius is infinite. (c) α—σ_w phase transition diagram. The colormap depicts the maximum L such that the ratio of Jacobian averages $\mathcal{J}_{f_1}$ using $f_1(r) = \text{sgn}(\log(r))|\log(r)|^L$ between the given point and the edge-of-chaos ordered transition (red line) at that α remains greater than unity, with a cutoff at 6. Green lines represent the corresponding points at which this ratio equals unity using $f_2(r) = (r-2)^L$ and $L = 1, 11, 21, \ldots, 101$ (starting from the right), with the enclosed region performing better than the edge-of-chaos ordered transition line with respect to the Jacobian average $\mathcal{J}_{f_2}$. Higher cutoffs and different Jacobian averages yield qualitatively similar phase diagrams. (d) Jacobian average $\mathcal{J}_f$ for the monotonically increasing functions $f(x) = x$, as a function of the weight gain parameter σ_w and $\alpha = 1, 1.2, 1.4, 1.6, 1.8, 2$ (blue, yellow, green, red, purple, orange). The difference $\Delta\mathcal{J}_f$ is computed against the value at the ordered transition line. (e) Same as in (d) but with the function $f(x) = \log(x)$.

propagation in the direction of a random eigenvector is dictated by the logarithmic Jacobian average.

In view of Eq. 2.6 and a recent framework established for RNNs in [23], to analyze this link between signal propagation and the (layerwise) Jacobian eigenvalue density $\rho(z)$ at random initialization, we compute averages over the eigenvalues of $\mathbf{W}^{l+1}\mathbf{D}^{l}$

$$\mathcal{J}_f := \langle f(|\lambda_i|)\rangle = \int_{\mathbb{C}} f(|z|)\rho(z)\,\mathrm{d}z \tag{2.7}$$

where f is an increasing function which penalizes small eigenvalues and rewards eigenvalues with large modulus. The local signal propagation ability of the deep network can then be expressed using its Jacobian average $\mathcal{J}_f$ defined in Eq. 2.7. To evaluate $\mathcal{J}_f$, we need to first determine the spectral density of a column-structured non-Hermitian heavy-tailed matrix, given $\mathbf{D}^{l}$ is a diagonal matrix. The first step is to obtain the resolvent for the bipartization of $\mathbf{WD}$, which after permutation forms a quaternionic resolvent (Eq. 2.24). In the limit of large matrix size $N \to \infty$, the strong law of large numbers is invoked to obtain the diagonal entries of the quaternionic resolvent which is a 2×2 matrix (Eq. 2.25) [67]. The off-diagonal entries of Eq. 2.25 allows for recovery of the limiting spectral density as in Eq. 2.26 (see Appendix 2.B for further details on $\rho(z)$ as in Eq. 2.18). On the other hand, information of the eigenvectors can be inferred from the diagonal entries of Eq. 2.25, we will further discuss this in Section 2.4. To compare the local signal propagation abilities between networks with different α, we first compute the Jacobian average at the ordered transition line $(\alpha, \overline{\sigma_w})$ at which the fixed point becomes non-negligible ($q^* \sim 0.01$, Fig. 2.2(c) solid red line). The ordered phase corresponds to the region where the Jacobian eigenvalues with modulus greater than unity are exponentially suppressed in probability. We then compute the dimensionless ratio of Jacobian averages (Eq. 2.7) at parameters (α, σ_w) to their values at the ordered transition $(\alpha, \overline{\sigma_w})$.

Evaluating the Jacobian average for Gaussian DNNs ($\alpha = 2$) shows that it is maximal at the classical edge-of-chaos transition, $\sigma_w = 1$, due to the concentration of eigenvalues around zero in the chaotic regime despite a larger spectral radius. Consequently, our characterization of deep information propagation is consistent with those reported elsewhere for Gaussian DNNs [16]. More importantly, for $\alpha < 2$ we find an extended region in phase space where the Jacobian average is greater than that of the ordered transition (Fig. 2.2(c)), indicating signal propagation through more layers in the network relative to the edge-of-chaos ordered transition point (criticality).

To determine the continuous nature of the transition to chaos, we compute the size of this extended critical region using monotonic averaging functions f which progressively become more discriminatory between small and large eigenvalue moduli with greater depth L, such as $f_1(r) = \mathrm{sgn}(\log(r))|\log(r)|^L$ (Fig. 2.2(c),

colormap) and $f_2(r) = (r-2)^L$ (Fig. 2.2(c), dashed green lines). Employing greater values of L in the Jacobian average serves as a proxy for information being able to penetrate a greater number of layers in the deep network. In heavy-tailed networks, the chaotic phase continuously transitions into a critical regime where the ratio of Jacobian averages remains greater than unity even for large L, predicting superior propagation of information through deep networks compared to the edge-of-chaos transition line. These results indicate that the edge-of-chaos criticality exists in an extended region of non-zero area in the (α, σ_w)–parameter space.

Additionally, our form of extended criticality remains robust to changes in the form of the Jacobian average for alternative increasing functions f. The Jacobian averages $\mathcal{J}_f$ for the functions $f(x) = x$ and $f(x) = \log(x)$ both reach their peak value at $\sigma_w = 1$ for $\alpha = 2$ (Fig. 2.2(d–e) orange lines). Conversely, when $\alpha < 2$, these averages continue to exceed $\mathcal{J}_f$ at $(\alpha, \overline{\sigma_w})$ across a broad spectrum of σ_w values. More precisely, smaller α values result in a wider range of high Jacobian averages, thereby expanding the critical regime for networks possessing a greater degree of heterogeneity in their connectivity strengths (Fig. 2.2(d–e) non-orange lines). Various forms of f universally distinguish $\mathcal{J}_f$ at (α, σ_w) and $(\alpha, \overline{\sigma_w})$. Hence the phase transition diagram is insensitive to the specific form of Jacobian average used in Eq. 2.7 and the extended critical regime closes into the classical edge-of-chaos point $\sigma_w = 1$ (Fig. 2.2(c)) in the Gaussian limit, $\alpha = 2$ (Fig. 2.2(d–e) orange lines).

2.3.3 Layerwise contraction and expansion of neural manifolds

We next consider the propagation of manifold geometry through deep random neural networks. It has been shown that random Gaussian feedforward networks may be trained precisely when information can propagate through them [16], which occurs at the classical edge of chaos [72, 9]. Through quantifying the layerwise geometry of $\mathbf{h}^l(\theta)$ from a simple circular manifold input parameterized by a scalar $\theta \in \mathbb{R}$, the classical ordered and chaotic regimes can be shown to correspond with the uniform compression and nonlinear expansion of internal neural representations respectively [9]. In particular, our findings demonstrate that the parameter L depicted in Fig. 2.2(c) effectively controls the depth of the fully-connected DNNs in situations where pairwise distances across the manifold display significant fluctuations. Furthermore, the extended critical phase overlaps with the ideal balance between contraction and expansion in neural manifolds, leading to improved training performance (Section 2.3.4).

We study how this circular manifold propagates through random heavy-tailed DNNs initialized at different (α, σ_w) without biases (see **Methods** for initialization details of fully-connected weights). For $\alpha = 1$, the ordered and chaotic phases

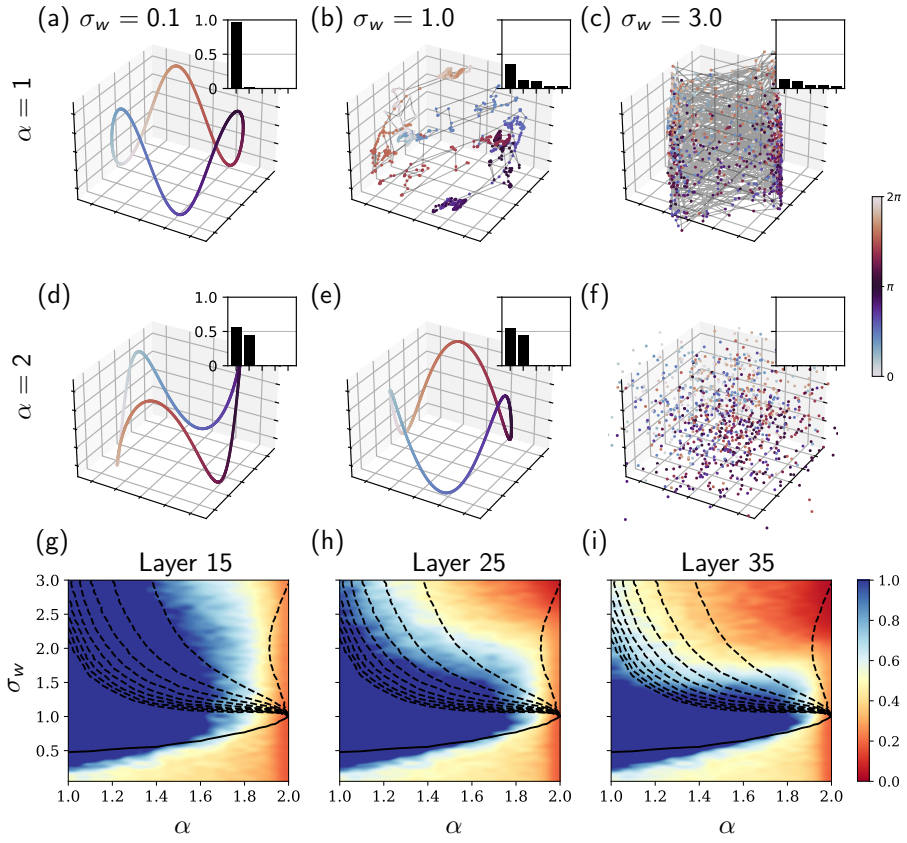

Figure 2.3: Propagation of manifold geometry through deep random networks. Propagation of a great circle through FC30 randomly initialized with $\alpha = 1$ projected to its three principal components (normalized), from (a–c) $\sigma_w = 0.1, 1, 3$ respectively; all 3 axes have cut-offs at ± 0.05. Insets show the fraction of variance explained by the top 5 singular values with respect to the manifold. The cyclic colorbar corresponds to the rotation angle θ of the input manifold which is set between $[0, 2\pi)$. The total variance and top singular value pair from left to right are respectively $(1.19 \times 10^{-17}, 1.16 \times 10^{-17}), (6.78 \times 10^{3}, 2.39 \times 10^{3})$ & $(1.11 \times 10^{5}, 1.58 \times 10^{4})$. (d–f) Same as in (a–c) but for $\alpha = 2$. The total variance and top singular value pair from left to right are respectively $(4.11 \times 10^{-29}, 2.28 \times 10^{-29}), (3.08 \times 10^{1}, 1.70 \times 10^{1})$ & $(4.27 \times 10^{4}, 3.01 \times 10^{2})$. (g–i) A phase transition plot with the colormap representing the coefficient of variation of pairwise distances as in Eq. 2.8 over an initially circular manifold propagated through a deep random network described by Eq. 2.2 with 1000 neurons at each layer, averaged over 50 network ensembles. The transitions are plotted for layer 15, 25 and 35 respectively. Overlaid are the theoretically predicted critical transition lines from Fig. 2.2(c). The classical edge-of-chaos point appears at $(\alpha, \sigma_w) = (2, 1)$.

(Fig. 2.3(a, c) respectively) correspond to regimes of contraction and nonlinear expansion which respectively occur uniformly throughout the manifold. Meanwhile, a combination of contraction and nonlinear expansion of input points is displayed along different parts of the manifold in the critical regime (Fig. 2.3(b)). Through principal component analysis (PCA) performed on the neural representations, the principal components (PCs) show the proportion of signal which has been preserved (Fig. 2.3(a–c) insets). In the chaotic regime, a majority of the signal, represented by the top 5 fraction of variance produces persists. On the other hand, the ordered regime experiences contraction rather than noisy PC dispersion (top singular value is small, Fig. 2.3 caption). Only the critical regime preserves the signal by avoiding contraction while maintaining a significant proportion of data in the first PC. For the Gaussian case $\alpha = 2$, the ordered and critical regime both preserve majority of the signal in the top two PCs but the chaotic regime mixes the signal and noise, leading to de-correlation of the input manifold and hence a much lower proportion of variance explained even for top PCs (Fig. 2.3(d–f)). As σ_w increases, the singular value distribution becomes flatter and the projected curves become more complex and entangled. Nonetheless, smaller values of α can decelerate the reduction of the singular values for deeper layers (vanishing length q^l of Eq. 2.17) in the ordered regime whilst preserving the percentage of the variance explained for the top PCs (i.e. hindering signal-noise mixing) in the chaotic regime.

To further quantify the fluctuations of pairwise distances, we use the dimensionless coefficient of variation

$$\mathrm{CV} = \frac{\mathrm{std}_\theta(\Delta h^l(\theta))}{\langle \Delta h^l(\theta) \rangle_\theta} \tag{2.8}$$

where $\Delta h^l(\theta) := \left\| \mathbf{h}^l(\theta + \Delta\theta) - \mathbf{h}^l(\theta) \right\|$ is the change in Euclidean distance of the l^{th} hidden layer as its input angle is perturbed by a small $\Delta\theta$; the average and standard deviation are taken over the angles $\theta \in [0, 2\pi)$. As shown in Fig. 2.3(g–i), both the classical ordered and chaotic regimes have low fluctuations of pairwise distances, as each regime contracts and nonlinearly expands neural inputs uniformly across the ambient space and thus the manifold. However, the extended critical regime displays large fluctuations of pairwise distances across the manifold and thus input space, demonstrating that the network qualitatively balances the contraction and expansion of neural inputs across space. Such balanced contraction and expansion does not exist beforehand (Fig. 2.C.1(a–d)) but emerges upon propagation by a number of layers in the network.

As the circular manifold propagates deeper through the network, the region exhibiting simultaneous contraction and expansion evolves in a manner roughly following the analytically predicted continuous phase transition parameterized by L (Fig. 2.3(g–i), black lines). It is important to note that the extended critical regime

is consistent with that of mixed contraction and expansion. The symmetry breaking caused by the simultaneous balancing of contraction and expansion throughout the neural manifold enables the propagation of information through many layers when networks are trained in this critical regime, extending the findings in [16] to heavy-tailed deep networks. The observed fluctuations of the classical critical point at $(\alpha, \sigma_w) = (2, 1)$ are smaller than those deep in the extended critical phase (Fig. 2.3(i)); this phenomenon potentially arises from a combination of finite-size effects and the localization properties of the corresponding eigenvectors in each regime.

2.3.4 Extended heavy-tailed initialization for deep neural networks

To achieve successful training of a DNN, it is essential to initialize its weights appropriately. Previous research has focused on investigating various techniques to prevent gradient explosion or vanishing [5, 66, 73]. One of the best current practices involves initializing networks with pretrained weights [74], which typically have heavy-tailed distributions as shown in Fig. 2.1. Additionally, the relationship between the signal propagation of the network and its dynamic critical regime suggests that initializing deep networks within the extended critical phase enhances their trainability. To verify this, we uniformly select parameters from a (α, σ_w)-grid and train FC10 (Eq. 2.2, $L = 10$) with tanh activation for 650 epochs using standard SGD (Fig. 2.4(a–c)); a small constant learning rate is applied to mitigate the instability of the learning algorithm (see **Methods** for further training details). For this section and onwards, we make the differentiation and term networks initialized with weights $\alpha < 2$ as heavy-tailed (initialized) neural networks and Gaussian (initialized) neural networks for $\alpha = 2$. DNNs are known to suffer from the exploding or vanishing gradient problem under Gaussian initializations far from the edge of chaos [16, 37]. To understand the advantage of pretrained initialization, we show that such network architectures can be successfully trained when initialized in the extended critical regime (Fig. 2.4), displaying a superior test accuracy and loss after under 50 epochs (Fig. 2.C.2). The interval of successful training for Gaussian initializations with $\alpha = 2$ spans between $0.7 \leq \sigma_w \leq 1.5$ due to finite-size effects of the 784-neuron layers as well as statistical fluctuations from training, with the interval closing into the classical edge-of-chaos point at $(\alpha, \sigma_w) = (2, 1)$ in the large network limit. By determining the epoch at which the Top-1 testing accuracy reaches the threshold of 93% in Fig. 2.4(c), we observe that the network achieves convergence to the desired weight configuration much more rapidly when initialized in the extended critical regime. The difference in training time due to variations in the initialization parameters spans multiple orders of magnitude: networks initialized deep in the critical regime can be successfully

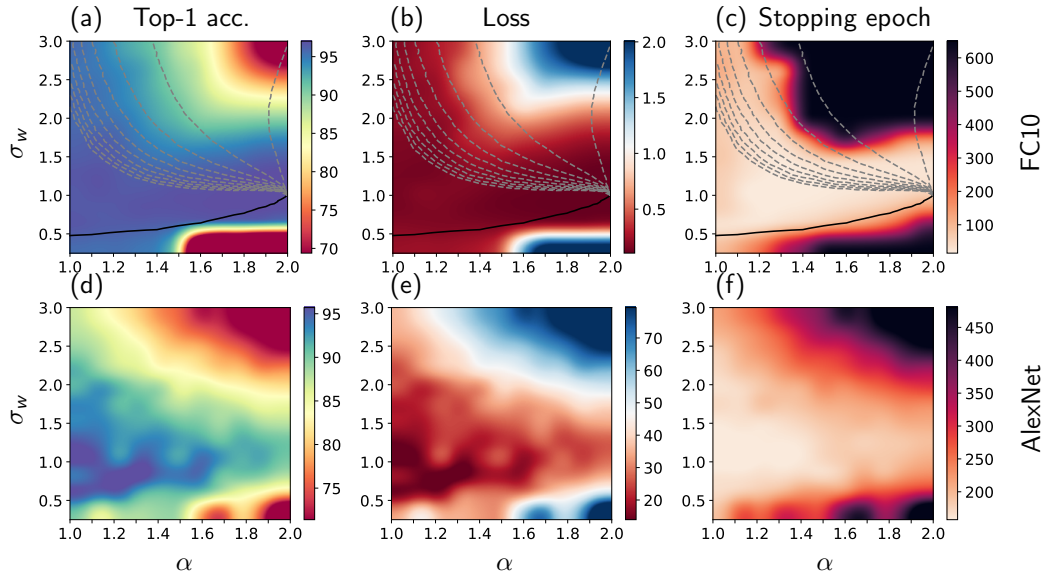

Figure 2.4: Extended criticality displays superior generalization and trainability for DNNs. (a–c) Results for FC10 trained on MNIST. (a) Top-1 test accuracy of FC10 at epoch 650 with quadric interpolation. The (α, σ_w) initializations are realized for $\alpha = 1, 1.1, 1.2, \cdots, 2$ and $\sigma_w = 0.25, 0.5, 0.75, \cdots, 3$. Every network was trained for 650 epochs with a batch size of 1024 and learning rate of 0.001 with the vanilla SGD algorithm. The colorbar has lower and upper cutoff at the 5% and 95% percentile of the testing accuracies at epoch 650. (b) The same as in (a) but with the colormap representing the test loss. (c) The same as in (a) but with the colormap representing the earliest stopping epoch reaching the Top-1 testing accuracy threshold of 93%. The colorbar has lower and upper cutoff at the 5% and 95% percentile of the earliest epochs. (d–f) Results for AlexNet trained on CIFAR10. (d) Top-1 train accuracy of AlexNet at epoch 500 with quadric interpolation. The realizations of the training initialization and colorbar bounds (5% and 95% percentile) are the same as in (a). Every network was trained for 500 epochs with a batch size 256 and learning rate of 0.001 with the vanilla SGD algorithm. (e) The same as in (d) but the colormap representing the train loss. (f) The same as in (d) but with the colormap representing the earliest epoch reaching the Top-1 train accuracy threshold of 70%.

trained in under 50 epochs, while deeply chaotic initialization prevents the network from reaching a testing threshold of 93% even after training is stopped at 650 epochs. For the phase transition of the accuracy and loss at different epochs, please see Fig. 2.C.2.

We further investigate the effect of batch sizes on networks initialized with Gaussian and heavy-tailed weights. For our controlled experiment, we adhere to vanilla SGD with the same learning rate to regulate the setting in Fig. 2.4, while manipulating the batch size in exponential increments. To reduce discrepancies between batch sizes, we plot the test accuracy after an extensive training time for $\alpha = 1$ and 2 each for 3 values of σ_w (Fig. 2.C.3), which represent the ordered, critical and chaotic phase respectively. For $\alpha = 1$, all networks are able to attain very close to perfect accuracy for the large range of batch sizes ($[16, 1024]$). This observation continues to hold for the Gaussian critical phase; but for the chaotic phase there is already a non-negligible decline in test accuracy for larger batch sizes and an even more consequential drop-off for the ordered regime (starting from batch size 32). As expected, heavy-tailed networks maintain their robustness of the Top-1 test accuracy across varying initialization regimes and batch sizes.

While CNNs utilize convolutional layers to process input data [75, 76, 2], they display remarkably similar dynamics to Eq. 2.2, given a fundamental network structure and the weights are Gaussian. In particular, the fixed point analysis for the covariance matrix of different sites (within the same hidden layer at the same depth) fundamentally shares the same mathematical structure with fully-connected DNNs if the evolution equation for the layerwise deviation from the fixed point is moved into Fourier space [77]. In view of the prominence of heterogeneous coupling (Fig. 2.1) as well as similarities with fully-connected DNNs, we now further demonstrate empirically the existence of the extended critical regime for CNN architectures.

Firstly, note that convolution kernels can be of any arbitrary dimension, we restrict our simulations to CNNs without biases where the layer tensors take form of 2D convolutions tensors, i.e. $\mathbf{W}^l_{\text{conv2D}} \in \mathbb{R}^{k_1 \times k_2 \times c_{\text{in}} \times c_{\text{out}}}$. Here k_1 and k_2 are respectively the width and height of the convolution filters; c_{in} and c_{out} are respectively the number of input and output channels. In a similar fashion, entries of each individual convolution tensor from a CNN initialized at (α, σ_w) are independently sampled from the distribution $S_\alpha\left(\frac{\sigma_w}{(2k_1k_2c_{\text{out}})^{1/\alpha}}\right)$; the same scaling as in [77] is recovered as $\alpha \to 2$. In order to fully contrast the effect of convolution weights at different initialization distributions, the fully-connected weight matrix entries in the network are uniformly sampled (see **Methods** more details).

We demonstrate our results for AlexNet [2] trained on a more complex CIFAR10 dataset [78]; each realized parameter pair (α, σ_w) of network initialization in

Fig. 2.4(d–f) is the same as (a–c). In parallel to fully-connected DNNs, an extended critical regime is also displayed for CNNs; these results suggest that the extended criticality is a universal property of heterogeneous DNNs (additional epochs can be found in Fig. 2.C.2).

The parameter space σ_w for high Top-1 accuracy significantly extends over a larger region for lower values for α, underscoring the preference for heterogeneous and heavy-tailed connectivities in DNNs concerning weight initialization and subsequent performance after training (Fig. 2.1). The enhanced training performance of the extended criticality regime in Fig. 2.4 coincides with very similar regions of high CV (Fig. 2.3(g–i)), indicating the consistency between the duality of learning and information propagation. Although the theory developed in Section 2.3 is founded on fully-connected DNNs, it is worth noting that AlexNet yields similar qualitative performance with effectively the same weight initialization scheme. Thus our results provide a crucial theoretical and computational guideline for the successful training and generalization of heterogeneous DNNs from the perspective of extended criticality.

2.4 Multifractality in heavy-tailed networks

We have now investigated the impact of layerwise Jacobian eigenvalues on signal propagation in DNNs, which is directly linked to the off-diagonal elements of Eq. 2.25 described by Eq. 2.26. We next study the dynamics of the Jacobian eigenvectors, which can be inferred from the diagonal elements of Eq. 2.25. In a recent study [23], it has been demonstrated that the eigenvectors of the Jacobian for heavy-tailed recurrent neural networks are multifractal. We illustrate that this property extends to heavy-tailed DNNs. Moreover, this feature is present in the top principal directions of neural representations at any depth. We demonstrate that the delicate mixture of localized and delocalized properties of multifractality generates more robust neural representations for fully-connected DNNs and benefits pretrained network models.

2.4.1 Multifractal Jacobian eigenmodes

Spatial multifractality emerges in various complex systems, such as the distribution of turbulent flow dissipation [79, 68]. It is a generalization of monofractality, where a single exponent is insufficient to describe the underlying scaling properties of complex systems; an entire spectrum of infinitely many exponents is required to characterize the system's effects.

Multifractals describe distributions of intensities or densities of quantities on (measure) supports. Consider a probability measure of general quantities μ_i where

i represents some spatial position. We impose that μ_i is normalized over an entire system of size M, i.e. $\sum_i \mu_i = 1$; hence μ_i can be interpreted as the probability of finding the quantity at position i. Given a box measure $\mu_{b(m)} := \sum_{i \in b(m)} \mu_i$, here the box $b(m)$ has size $m \ll M$ and forms a subset of the system. The q^{th} moment of the coarse-grained measure by an infinitely small scale $m \to 0$ satisfies the relation

$$\langle \mu_m^q \rangle := \sum_b \mu_{b(m)}^q \propto \left(\frac{m}{M}\right)^{\tau(q)} \tag{2.9}$$

where $\tau(q)$ is defined as the mass exponent and M/m is the level of coarse-graining [80]. For monofractals with fractal dimension D_f,

$$\tau(q) = (q-1)D_f, \tag{2.10}$$

there is a linear dependence of the mass exponent $\tau(q)$ on q. From a box-counting perspective, this indicates that the quantities appearing in a box $b(m)$ of size m does not change meaningfully with respect to the location of the box. However, this relation is violated in multifractals as discussed in the following.

Recent findings in the context of recurrent neural networks [23] have shown that the right eigenvectors of its Jacobian for sigmoidal activation functions are spatially multifractal over neural sites with a mixture of localized and delocalized properties. Since the (preactivation) layerwise Jacobians in Eq. 2.3 share a similar structure to that of RNNs where the connectivity matrix is multiplied by a diagonal matrix with entries as the derivative of the activation function, these results can be applied to fully-connected DNNs with the same techniques for deriving localization properties of the left and right eigenvectors in terms of the inverse participation ratio $\text{IPR}_q(v) = \sum_{i=1}^N |v_i|^{2q}$ for any normalized $v \in \mathbb{R}^N$ (see Appendix 2.B.1). Such multifractal localization over neurons is defined by a nontrivial dependence on q of the (generalized) fractal dimension D_q appearing in the inverse participation ratio [81]

$$\text{IPR}_q(v) \sim N^{(1-q)D_q} \tag{2.11}$$

for large system size N. Note that $D_q = 0$ (1) corresponds to localized (delocalized) spatial profiles over neurons. Intuitively, a delocalized (eigen)vector has $v_i \sim 1/\sqrt{N}$ where the magnitude for all components are (roughly) the same. On the other hand, if the vector is significantly different from zero only on one or a very small number of sites, it is localized. Based on Eq. (2.11), the fractal dimension corresponding to an eigenvector can be estimated via the asymptotic relation $D_q \sim \log_N \text{IPR}_q(v)/(1-q)$. Similar to the Jacobian spectral density, the IPR of the left and right Jacobian eigenvectors can be inferred from Eq. 2.25 and have the respective analytical expression as in Eq. 2.27.

To demonstrate such multifractality in DNNs, we first examine quantities as discussed above for the right eigenvectors of the layerwise Jacobians corresponding to

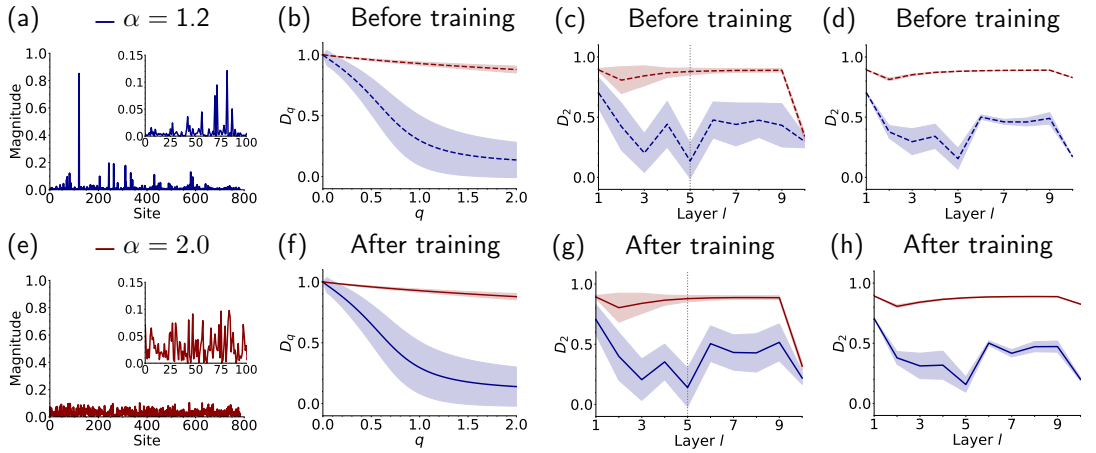

Figure 2.5: Localization properties of preactivation layerwise Jacobian eigenvectors. (a) Absolute site magnitudes of the right eigenvector of the layerwise Jacobian matrix $\mathbf{W}^6\mathbf{D}^5$ with the largest eigenvalue modulus; there are a total of 784 sites. Here $\mathbf{D}^5$ is computed for a single arbitrarily selected MNIST image input. The network FC10 is randomly initialized at $(\alpha, \sigma_w) = (1.2, 1)$. The inset zooms in on the the first 100 sites showing a mixture of localization and delocalization across neuron sites. (b) The mean fractal dimension $D_q \sim \log_N \mathrm{IPR}_q(v)/(1-q)$ across 784 right eigenvectors of $\mathbf{W}^6\mathbf{D}^5$ as in (a) plotted for $\alpha = 1.2$ (dashed blue curve) and $\alpha = 2.0$ (dashed red curve) with $\sigma_w = 1$ before training in both cases. The one standard deviation is represented by the shaded area for $\alpha = 1.2$ (red) and $\alpha = 2.0$ (blue). (c) The average correlation dimension D_2 of the preactivation Jacobian $\mathbf{W}^{l+1}\mathbf{D}^l$ eigenvectors vs layer l, i.e. for $l = 1, \cdots, 10$ (Eq. 2.2). The same image input as in (a) & (b) is used for computing each $\mathbf{D}^l$. The average of D_2 is taken over 784 eigenvectors of $\mathbf{W}^{l+1}\mathbf{D}^l$ at each depth l; the one standard deviation is represented in the manner same as (b) & (f). The dotted grey line coincides with D_2 of the featured layer $l = 5$ plotted in (b) before training. (d) The average $\overline{D}_2$ plotted across layer l. Here for each MNIST image sample, $\overline{D}_2$ is averaged over the eigenvectors. At each layer, the average and standard deviation of $\overline{D}_2$ is taken across 1000 randomly selected training samples. (e) Same as (a) but for $\alpha = 2.0$. The activity fluctuations spread evenly over the sites and remain delocalized. (f) Same as in (b) but after training the corresponding networks for 650 epochs (solid curves). The training algorithm setting for FC10 can be found in **Methods**. (g) Same as in (c) but after training for 650 epochs. The dotted grey line coincides with D_2 of the featured layer $l = 5$ plotted in (f) after training. (h) Same as in (d) but after training for 650 epochs.

a single input. Random heavy-tailed networks (before training) inherently possess multifractal layerwise Jacobian eigenvectors (Fig. 2.5(a)) where the magnitudes of neuron sites exhibit large fluctuations and the geometrical scaling varies from site

to site (figure inset). Comparatively, the associated eigenvectors from Gaussian networks contain evenly distributed magnitudes over the neurons (Fig. 2.5(e) and inset). We compute D_q for both cases at the critical initialization where $\sigma_w = 1$. At $\alpha = 1.2$, the eigenvectors of $\mathbf{W}^6\mathbf{D}^5$ exhibit multifractality with high variance (Fig. 2.5(b) dashed blue curve and shaded region), which persists even after training (Fig. 2.5(f) solid blud curve). In contrast, for the Gaussian network, the eigenvectors of $\mathbf{W}^6\mathbf{D}^5$ remain delocalized even after the completion of training. This is illustrated by the minimal difference between the dashed and solid red curves in Fig. 2.5(b, f). Although a perfect representation of delocalization, i.e. $\langle D_q \rangle = 1 \ \forall q$ for the Gaussian case (Fig. 2.5(b, e) red) is set back by finite size effects and can only be achieved when the system size approaches infinity, the difference between multifractality ($\alpha = 1.2$) and delocalization ($\alpha = 2.0$) is fully displayed: the fractal dimension D_q corresponding to the heavy-tailed network, is a non-trivial function which experiences a much more significant decrease with respect to q relative to the Gaussian case.

Since D_q is a non-increasing function of q, the correlation dimension D_2 (fractal dimension evaluated at $q = 2$) can be directly used as a measure of delocalization/localization of the associated vectors regardless of the finite size effects. At $\sigma_w = 1$, Gaussian networks exhibit a notable prevalence of delocalized eigenvectors throughout each layer, which again is retained from initialization and persist after training (Fig. 2.5(c, g) red lines). Conversely, the Jacobian eigenvectors of heavy-tailed networks demonstrate varying levels of multifractality throughout the layers (Fig. 2.5(c, g) blue lines). Heavy-tailed networks demonstrate visibly greater standard deviation at each layer l, indicating larger fluctuations in eigenvector localization properties compared to Gaussian networks (Fig. 2.5(c, g) blue and red shaded areas). The sudden decline of D_2 at layer 10 is attributed to the fact that the final readout layer comprises only 10 neurons in both cases. For other regimes with non-unitary σ_w, the microscopic details of D_2 vary from case to case (Fig. 2.C.4). In the ordered regime $\sigma_2 = 0.25$, the difference between eigenvector localization properties of Gaussian and heavy-tailed networks decreases; for larger values $\sigma_w = 1.5, 3$, we observe similar qualitative phenomena as in the critical regime ($\sigma_w = 1$), but the Gaussian networks demonstrate larger fluctuations of D_2 within each layer due to the increase in the variance of associated weights (Fig. 2.C.4 red shaded area). We expand our investigation to examine the dynamics of large samples of image inputs based on $\overline{D}_2$. In this case, $\overline{D}_2$ is averaged across all the corresponding eigenvectors for a single input (see Fig. 2.C.4(b, h)). Similar patterns as in Fig. 2.C.4(c, g) are observed, albeit with smaller fluctuations across the training samples. The overall shape of $\overline{D}_2$ remains similar to that for a single input (Fig. 2.C.4(c)) and is preserved even after training, indicating that the localization properties of Jacobian eigenvectors are persistently maintained across the entire dataset.

Heavy-tailed networks consistently reveal layerwise Jacobian eigenvectors with a greater degree of localization, regardless of the value of σ_w. Our results presented in Fig. 2.4 illustrate that improved training and generalization can be achieved with smaller values of α. These heavy-tailed initialized networks display a more prominent multifractal structure, which demonstrates the crucial role of these unique characteristics induced by connectivity heterogeneity in facilitating information propagation. In particular, an eigenvalue of a given eigenvector corresponding to successive layers exhibit larger modulus than those at the ordered transition line, quantified by the high Jacobian average (Eq. 2.7). This causes a subset of spatially multifractal directions in neural space to consistently appear above the critical line (Fig. 2.2), allowing the eigenvectors to experience expansion in those directions while contracting others, a form of symmetry breaking. This balance between contraction in some directions and expansion in others allows neural networks to prioritize different parts of the input when training on the problem dataset. However, deep networks with Gaussian statistics have spatially delocalized layerwise Jacobians eigenvectors, so that each direction appears equivalent to the network even after quenching regardless of the problem data. Consequently, Gaussian deep networks can only contract or nonlinearly expand and fold Jacobian eigenvectors in all directions simultaneously without precision, corresponding to the classical ordered and chaotic regimes respectively [7].

2.4.2 Multifractality emerging in neural representations

We continue to analyze the neural representations initialized under different regimes before and after training on realistic datasets such as MNIST [71]. To comprehend the activity patterns of the hidden layers $\mathbf{h}^l \in \mathbb{R}^N$ produced by the network upon presentation of the visual dataset, we compile the population responses to each of the P images from the dataset into a matrix $\boldsymbol{X} \in \mathbb{R}^{N\times P}$. One useful measure to characterize the dimensionality of $\boldsymbol{X}$ is via its effective dimension (ED) [82]:

$$\mathrm{ED}(\boldsymbol{X}) = \frac{\mathrm{Tr}(\boldsymbol{C})^2}{\mathrm{Tr}\left(\boldsymbol{C}^2\right)} = \frac{\left(\sum_{i=1}^N \lambda_i\right)^2}{\sum_{i=1}^N \lambda_i^2} \tag{2.12}$$

where λ_i for $i = 1, \cdots, N$ are the eigenvalues of the covariance matrix $\boldsymbol{C} := \frac{1}{P}\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{x_0}\boldsymbol{x_0}^\top$; $\boldsymbol{x_0} \in \mathbb{R}^N$ is the centered manifold corresponding to $\boldsymbol{X}$. Assuming the covariance matrix is diagonalizable, it can also be expressed as

$$\boldsymbol{C} = \frac{1}{P}\sum_{i=1}^N R_i^2 \boldsymbol{u_i}\boldsymbol{u_i}^\top. \tag{2.13}$$

The eigenvectors of the covariance matrix, denoted by $\boldsymbol{u_i}$, are referred to as principal directions or principal axes and the associated principal component (PC)

is simply the projection $\boldsymbol{X}\boldsymbol{u}_i$, i.e. the entries $\mathbf{u}_{ij}$ describes the weighting of the i^{th} on the j^{th} feature of $\boldsymbol{X}$. Their corresponding eigenvalues $R_i^2/P := \lambda_i$, are arranged in descending order. The effective dimension is essentially the participation ratio of the covariance matrix eigenvalues and can be interpreted as approximately the number of PCs required to maximally explain the variance of the data [83]. In the following, we explore the properties of these geometrical quantities for Gaussian and heavy-tailed initialized networks.

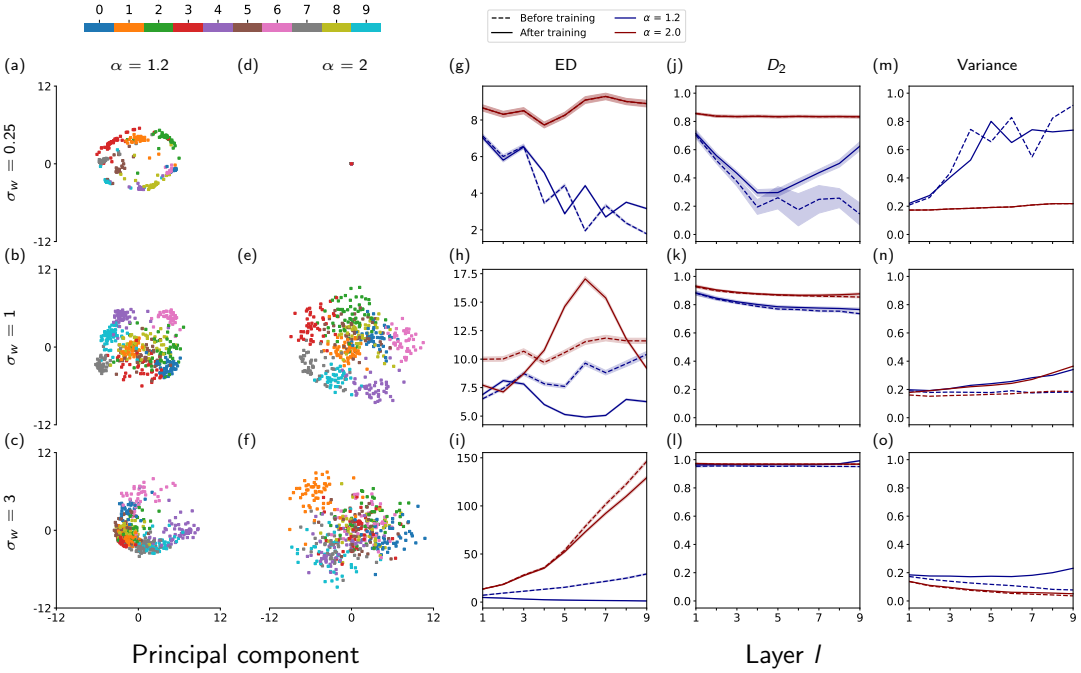

Figure 2.6: Localization properties and effective dimensions of DNN activations. (a–c) The top 2 PCs of the final activation layer $\mathbf{x}^l$ ($l = 9$) of a trained FC10 at epoch 650, the networks are initialized at $\alpha = 1.2$ with $\sigma_w = 0.25, 1$ and 3 respectively. The colorbar indicates the associated image classes $\{0, 1, \cdots, 9\}$ of the MNIST test dataset. (d–f) The same as in (a–c) but for $\alpha = 2$. (g–i) The ED for each activation layer is computed based on Eq. 2.12 for $\sigma_w = 0.25$, 1 and 3 respectively. The lines indicate the mean values and the shaded area corresponds to the standard deviation, both averaged over the batches of input images. The blue and red curves represent the network initialized at $\alpha = 1.2$ and $\alpha = 2.0$ respectively. Dashed curves are computed based on FC10 before training; solid curves after the networks have been trained for 650 epochs. (j–l) The average correlation dimension D_2 of the top 5 principal directions is plotted over depth; the neural representations are obtained from the full-batch MNIST train dataset. (m–o) Percentage of the cumulative variance explained by the top 5 PCs plotted against layer l. Neural representations are obtained as in (j–l).

Based on the training results in Fig. 2.4 and 2.C.3, it is important to understand how the neural representations in the critical, ordered and chaotic regime distinguish heavy-tailed and Gaussian initialized networks during training for real-world datasets. Even though the overall train/test accuracy of FC10 is very similar between Gaussian and heavy-tailed networks for $\sigma_w \in [0.7, 1.5]$ which are close to the critical regime (Fig. 2.4), it is important to note that the neural representations across the network exhibit distinct properties. To show this, we perform PCA on the hidden layers of relevant networks and study their underlying dynamics. Upon projecting the final activation layer of the MNIST test dataset onto the top two principal components as shown in Fig. 2.6, the separability of different input classes in heavy-tailed DNN activations persists across a wider range of σ_2 (Fig. 2.6(a–c)). However, the particular value of σ_w becomes extremely important for Gaussian networks. Initialization within the ordered regime inhibits learning of Gaussian networks and simultaneously prevents dimensionality expansion (Fig. 2.6(d)). On the other hand, the chaotic initialization causes the input classes to be mixed back together, causing difficulties for classification (Fig. 2.6(f)). Only with appropriate σ_w at the edge of chaos [9, 82] can Gaussian networks be trained optimally (Fig. 2.6(e)). This finding highlights the importance of understanding the dynamics of heavy-tailed networks, as they exhibit superior performance for more extreme values of σ_w, even when trained using the same methods as Gaussian networks.

At the criticality point, Gaussian networks expand dimensionality (ED) larger than that of the original input to allow for better separation between different image classes. Previous studies have shown trained DNNs often possess a *hunchback* shape for their EDs across the network depth [84] which depicts a decoding and encoding process, this finding matches with the red solid curve ($\alpha = 2$) in Fig. 2.6(h). In comparison, the ED of heavy-tailed networks is significantly reduced across all layers as shown in Fig. 2.6(h) by the blue curves, indicating a different input class separation mechanism with a higher degree of efficiency. For the ordered regime $\sigma_w = 0.25$, Gaussian networks suffer from gradient vanishing causing the weights to barely evolve [66], leading to very similar values of ED before and after training across all layers (Fig. 2.6(g), solid and dashed red curve overlap significantly); heavy-tailed networks continue to maintain their trainability by suppressing dimensionality (Fig. 2.6(g) solid blue curve). Furthermore, this observation remains valid even in the chaotic regime for heavy-tailed networks. In this case, Gaussian networks face the problem of input signal mixing with noise due to gradient explosion [66]. Consequently, the effective dimensionality is needlessly increased, which makes decoding significantly more difficult (Fig. 2.6(i)).

Heavy-tailed networks effectively combat negative impacts from various settings by compactly reducing the dimensionality of the neural manifolds. For all three cases of initialization regimes discussed above, we find that the top principal

direction of neural representations from heavy-tailed network (computed based on Eq. 2.13) tend to be more localized than those of Gaussian networks due to the lower values of D_2 (Fig. 2.6(j–l)). Moreover, with the same number of top principal direction, those from heavy-tailed networks account for similar levels of variance in the data at $\sigma_w = 1$ (Fig. 2.6(n)) and a greater amount at $\sigma_w = 0.25$ and $\sigma_w = 3$ (Fig. 2.6(m, o)) respectively. Unlike the layerwise Jacobian eigenvectors (Appendix 2.B.1), majority of the principal direction for heavy-tailed DNNs of the activations are relatively delocalized depending on the layer and σ_w (Fig. 2.C.5 blue curves). However, the Gaussian networks possess very delocalized principal directions across all PCs (Fig. 2.C.5 red curves) and the associated cumulative variance percentage of the PCs recovers to 1 at a similar rate for $\sigma_w = 1$ and slower rate for $\sigma_w = 0.25, 3$ (Fig. 2.C.6). For preactivations, the localization degree of the Jacobian eigenvectors generally become more intense as their rank based on the corresponding eigenvalues increases (Fig. 2.C.8 blue curves). Consequently, fewer top PCs can explain a higher proportion of the variance in the neural representations for non-critical initialization (Fig. 2.C.9 blue curves). Besides the concept manifold separability, the preactivation layers overall share qualitatively similar characteristics as the activations (Fig. 2.C.7–2.C.9).

For fully-connected DNNs initialized with heavy-tailed weights, the leading principal directions with multifractal characteristics reveal their heterogeneity in feature selection, which remains robust against the specific values of σ_w. Employing a similar reasoning as discussed in Section 2.4.1 regarding Jacobian eigenvectors, a precarious combination of delocalized principal directions is required for Gaussian DNNs for precise extraction of local input features from the internal neural space. However, for heavy-tailed DNNs generally producing a lower effective dimension, only a few yet robust multifractal components are necessary. In light of these observations, heavy-tailed weights enable the effective balance of dimensions for an extensive range of the scale parameters (Fig. 2.1(c)).

2.5 Discussion

In this study, we have developed a theoretical framework to study the widely observed phenomenon of heavy-tailed universality in deep neural networks. Our framework identifies an extended critical regime of enhanced signal propagation which does not require fine-tuning of parameters, providing an initialization method that enhances generalization and accelerates training. Furthermore, we have discovered the emergence of multifractality in the eigenvectors of the input-output layerwise Jacobians as well as principal directions of neural representations. This multifractal property enhances robust neural representations for fully-connected DNNs.

The properties of the extended critical phase generalizes the current analysis of singular values [13] to a complete spectral theory of deep networks encompassing eigenvectors, and thus the spatial properties of the system dynamics, as well as eigenvalues [23]. We have obtained this more general formulation through the use of dynamical systems theory, bypassing the singular values and opening up analysis on the spatially local properties of signal propagation via the eigenvectors of **WD**. Thus, we have established strong connections between the improved performance of DNNs, spectral properties of layerwise Jacobian matrices and extended criticality. The theory is also universal in the sense that the precise distributions of weights are not required to be known apart from their tail asymptotics. Additionally, the derived mean-field theory in Eq. (2.17) applies to all sigmoidal activation functions, or more generally, functions that are $o(|x|)$.

Our results on heavy-tailed DNNs indicate a change of view from the critical point or line requiring fine-tuning of parameters, towards a novel, extended critical regime. Previous studies have demonstrated that criticality between the ordered and chaotic phases enables effective information propagation across layers [16, 9], and prevents gradient vanishing and explosion from back-propagation, thus facilitating training. Aside from this functional advantage entailed by the edge of chaos, our work has demonstrated a new type of initialization scheme aligned with the extended criticality regime which supports superior information propagation. This new notion of initialization improves trainablity and generalizability for both fully-connected DNNs as well as a certain class of CNNs. In particular, multifractality emerges as a consequence of heavy-tailed, heterogeneous weights which does not appear in Gaussian networks. Because of multifractality which balances localization and delocalization inherent in eigenvectors of heavy-tailed networks, the trainability as well as generalizability of DNNs can be significantly improved. Gaussian deep networks can only contract or nonlinearly expand and fold neural representations in all directions simultaneously without precision, corresponding to the classical ordered and chaotic regimes respectively [7].

The functional advantages of extended criticality thus illustrate a robust mechanism by which DNNs can possess remarkable performance in solving real-world problems. Our theory supplies the extended critical regime with multifractal properties as a key guide for performing heavy-tailed initialization, thus providing a principled explanation of the empirical observation that DNNs with more heavy-tailed singular spectra of weight matrices generalize better [13]. As heavy-tailed weights emerge during the training process from Lévy distributed gradients and gradient noise [56, 30], our theory also provides a framework for understanding complex learning dynamics [30, 85], which may result in more refined training algorithms for different learning tasks.

The emergence of multifractal eigenvectors in the layerwise Jacobians and principal direction of neural representations distinguishes the extended criticality of

our theory from other schemes such as hierarchical modular networks with Griffiths phases [24]. Moreover, Griffiths phases are clearly separated from the inactive and active phases by a first-order phase transition at the critical spreading rate; the extended critical regime instead reveals a second-order continuous transition with the active chaotic phase such that the crossover region in phase space does not diminish completely with increasing network size. A continuous transition also exists between the extended critical and ordered phases, parameterized by the cutoff of exponential suppression in the eigenvalue density.

Finally, criticality underlies a wide range of biological systems ranging from families of proteins, networks of neurons to flocks of birds with crucial optimal computational capabilities and large dynamical repertoires [86]. By linking heavy-tailed statistical physics with machine learning via random matrix theory, our new formalism on the concept of extended criticality implies prevalent applicability to understanding these systems, suggesting that extended criticality with complex dynamics can likely be a general governing principle of biological and artificial intelligence.

Supplementary Materials

In this section, we provide details on simulations and theory. Additional computational results are also included to complement the main text.

2.A Methods

Lévy alpha-stable distribution fitting. The fitting of the weights to a stable and Gaussian distribution only include fully-connected weight matrices and convolution tensors; the method of maximum likelihood estimation (MLE) is used for finding the parameters associated to the aforementioned distributions, i.e. the tuple $(\alpha, \beta, \sigma, \mu)$ for stable distributions and the pair (μ_N, σ_N) for Gaussian distributions. Weight entries with absolute values less or equal to 1×10^{-5} are pruned. A total of 33 networks are analyzed from Pytorch 1.9.0 (2068 weight matrices/convolution tensors) and 20 from Tensorflow 2.9.2 (3776 weight matrices/convolution tensors), all pretrained on ImageNet1K [5]. Shared network architectures by both libraries including ResNet, DenseNet & VGG are only investigated from the Pytorch library. Fig. 2.1(b–c) inclues a total of 5568 weight matrices/tensors out of 5844 since matrices/tensors with fewer than 1000 entries are excluded from the analysis.

After obtaining the stability and scale parameter, α and σ after the stable distribution fitting, we implement the following scale parameter normalization scheme

$$\sigma_w = (2\sqrt{N_h N_w})^{1/\alpha}\sigma \quad \text{(fully-connected)} \tag{2.14}$$

$$\sigma_w = (2k_1 k_2 c_{\text{out}})^{1/\alpha}\sigma \quad \text{(convolutional 2D)} \tag{2.15}$$

respectively for fully-connected weight matrices $\mathbf{W} \in \mathbb{R}^{N_h \times N_w}$ and convolutional tensors $\mathbf{W}_{\text{conv2D}} \in \mathbb{R}^{k_1 \times k_2 \times c_{\text{in}} \times c_{\text{out}}}$. Note that thescale normalization schemes in Fig. 2.1(c) depend on both α and σ.

Fully-connected DNN architecture and weight initialization. The architecture of the fully-connected DNNs trained in Fig. 2.1(d–h) (FC4) and Fig. 2.4(a–c)

(FC10) follow as in Eq. 2.2 where $\mathbf{x}^l = \tanh(\mathbf{h}^l)$ for $l = 1, \cdots, L-1$ and $\mathbf{x}^L = \mathbf{h}^L$ where a direct readout is used for the final layer. Biases for all layers are set to zero. Cross-entropy loss is the only type of loss function used for all fully-connected DNNs in this report. All fully-connected DNNs in the text are trained on MNIST, in particular the input and hidden layers all have 784-neurons, except for the (final) output layer which only has 10 neurons equivalent to the number of input classes in the dataset.

As the biases in all fully-connected DNNs are excluded, only the weights need to be initialized. A fully-connected DNN initialized at (α, σ_w) has all weight connectivities W_{ij}^l at each layer l independently sampled from the centered and symmetrical stable distribution $S_\alpha(\sigma_w/(2\sqrt{N_l N_{l-1}})^{1/\alpha})$ where $\mathbf{W}^l \in \mathbb{R}^{N_l \times N_{l-1}}$ (inverse of Eq. 2.14).

Propagation of circular manifold. A full batch of inputs of $\mathbf{x}(\theta) = \sqrt{N}[\mathbf{u}^0 \cos\theta + \mathbf{u}^1 \sin\theta]$, $\theta \in [0, 2\pi)$ where a pair of random vectors $\mathbf{u}^0$ and $\mathbf{u}^1$ form an orthonormal basis for a 2 dimensional subspace of $\mathbb{R}^N$ is propagated through fully-connected DNNs as in Eq. 2.2 (depth $L = 35$) randomly initialized based on the phase transition diagram in Fig. 2.2(c). The colormap represents the coefficient of variation for the preactivation layers computed via Eq. 2.8 for $l = 15, 25, 35$.

Optimization settings for fully-connected DNNs. A standard stochastic gradient descent (SGD) algorithm is used for the training of FC4 in Fig. 2.1(d–h) with constant learning rate 0.5 and batch size 256.

A total of 132 FC10's are trained also using a similar SGD setting: constant learning rate 0.001 and batch size 1024, a small learning rate is chosen in this case to mitigate the effect of the instability of the algorithm. We initialize the weights based on the phase transition diagram in Fig. 2.2 with grid point realizations at $\alpha = 1.0, 1.1, \cdots, 2.0$ increments of 0.1 and $\sigma_w = 0.25, 0.5, \cdots, 3.0$ increments of 0.25.

Initialization and optimization settings for AlexNet. We train a downsized version of AlexNet on the CIFAR10 dataset [78], please see **Code availability** for the exact architecture. For one AlexNet initialized at (α, σ_w), the entries of 2D convolutions tensors $\mathbf{W}^l_{\text{conv2D}} \in \mathbb{R}^{k_1 \times k_2 \times c_{\text{in}} \times c_{\text{out}}}$ are independently sampled from the distribution $S_\alpha\left(\frac{\sigma_w}{(2k_1 k_2 c_{\text{out}})^{1/\alpha}}\right)$ (inverse of Eq. 2.15). All fully-connected weight matrix entries in the network are initialized using the default uniform distribution in Pytorch, i.e. for any $\mathbf{W} \in \mathbb{R}^{N_h \times N_w}$, the entries are independently drawn from $\text{Unif}(-1/\sqrt{N_w}, 1/\sqrt{N_w})$.

We adopt sigmoidal activation tanh and no biases are applied for consistency. All networks are trained under the vanilla SGD algorithm with batch size 256 and

learning rate of 0.001.

Multifractality of layerwise preactivation Jacobians for fully-connected DNNs. The right eigenvectors of the preactivation layerwise Jacobian $\frac{\partial \mathbf{h}^{l+1}}{\partial \mathbf{h}^{l}} = \mathbf{W}^{l+1}\mathbf{D}^{l}$ and principal directions of the neural representations at different layers are explored with their localization properties computed based on Eq. 2.11 for FC10. For Fig. 2.5(a–c, e–g) and Fig. 2.C.4, the microscopic statistics of the layerwise Jacobian for one single input is examined. In Fig. 2.5(d, h), the layerwise Jacobians corresponding to 1000 arbitrarily selected inputs are analyzed. The correlation dimension D_2 is used as a measure for the degree of multifractality throughout the text (Fig. 2.5).

Localization properties of neural representation principal directions. For FC10, the activity manifolds of the preactivation layers $\mathbf{h}^l$ are recorded. Principal directions of the covariance matrix for each $\mathbf{h}^l$ are obtained as in Eq. 2.13, then their respective correlation dimensions D_2 are computed in order to determine the localization properties (Fig. 2.6).

Data availability. Weights of the networks pretrained on ImageNet1K can be downloaded from the PyTorch and TensorFlow libraries. We used the MNIST [71] and CIFAR10 [78] datasets for the training of fully-connected DNNs and AlexNet respectively.

Code availability. The code used for generating all data, figures and network training is available at the following GitHub repository: `https://github.com/CKQu1/extended-criticality-dnn`.

2.B Lévy mean-field theory

Applying the generalized central limit theorem [32, 26] to Eq. 2.2 in the limit $N \to \infty$ yields the convergence of inputs $\mathbf{h}_i^l$ over neurons i to a stable random variable in distribution,

$$\begin{aligned} \mathbf{h}_i^l &\overset{i}{\sim} S_\alpha \left[\left(\frac{\sigma_w^\alpha}{2N} \sum_{j=1}^{N} \left| \phi\left(\mathbf{h}_j^{l-1}\right) \right|^\alpha + \frac{\sigma_b^\alpha}{2} \right)^{1/\alpha} \right] \\ &= S_\alpha \left[\left(\frac{\sigma_w^\alpha}{2} \int |\phi(z)|^\alpha p_{\mathbf{h}^{l-1}}(z) \mathrm{d}z + \frac{\sigma_b^\alpha}{2} \right)^{1/\alpha} \right] . \end{aligned} \tag{2.16}$$

Parameterizing the distribution of the neural input $\mathbf{h}_i^l \sim S_\alpha((q^l/2)^{1/\alpha})$ at layer l by q^l, we obtain a Lévy mean-field iterative map for q^l from Eq. 2.16 by repeatedly

applying the above derivation of stable distributions,

$$q^l = \sigma_w^\alpha \int |\phi(z)|^\alpha p_{S_\alpha\left((q^{l-1}/2)^{1/\alpha}\right)}(z)\mathrm{d}z + \sigma_b^\alpha \tag{2.17}$$

for $l = 2, \ldots, L$, where the initial condition is $q^1 = \sigma_w^\alpha q^0 + \sigma_b^\alpha$ and $q^0 = \sum_i |\mathbf{x}_i^0|^\alpha / N$ is the α^{th} moment of the initial activity layer assuming that it is finite. The parameter q^l characterizes the fluctuations of the neural input distribution at layer l and reduces to the classical normalized squared length when $\alpha = 2$ [9].

Eq. 2.17 constitutes the Lévy mean-field equation for fully-connected DNNs. The fixed point is then obtained by setting $q^{l-1} = q^l = q^*$ for large l; in order to guarantee the finiteness of the right-hand side of the equation, we assume that $\phi(|x|) = o(|x|)$ using the little-o notation, which includes all sigmoidal functions.

Leveraging the statistical properties of the eigenvectors and eigenvalues of the Jacobian as well as its constituent preactivation layerwise Jacobians $\mathbf{W}^{l+1}\mathbf{D}^l$ gives us a method to consistently characterize the onset of edge-of-chaos criticality.

2.B.1 Statistics of the Jacobian of heavy-tailed DNNs

The Jacobian operators of Gaussian random neural networks satisfy a circular law with a finite spectral radius [87, 88] and delocalized eigenvectors [81] that spread evenly across the neural sites. Because the maximum singular value of heavy-tailed matrices is infinite [67], we develop a criterion for criticality using the entire eigenvalue distribution of the Jacobian matrix. Since the Jacobian has the form of a column-structured non-Hermitian heavy-tailed random matrix, we employ recent results [23] on the statistics of matrices of the form $\chi_i W_{ij}$.

Ref. [26] demonstrates the key properties of a time-varying Jacobian operator around the stationary state in a recurrent neural network (RNN) context, which is equal to the layerwise Jacobian with form $\mathbf{WD}$ of our feedforward network around the fixed point due to Eq. 2.2. Exploiting the locally treelike properties of heavy-tailed random matrices, a cavity approach is applied [89, 90, 23] to find that its eigenvalue density $\rho(z)$ has infinite support with an exponential cutoff at large modulus [67], such that

$$\rho(z) = \frac{y_*^2 - 2\|z\|^2 y_* \partial_{\|z\|^2} y_*}{\pi} \left\langle \frac{|\chi_i|^2 SS'}{(\|z\|^2 + |\chi_i|^2 y_*^2 SS')^2} \right\rangle_i \tag{2.18}$$

where $\langle .. \rangle_i$ denotes averaging over i and any relevant random variables, $\chi_i = \sigma_w \phi'(h_i)$ varies over neurons, $S, S' \sim S_{\alpha/2}(1, (c_\alpha/4c_{\alpha/2})^{2/\alpha}, 0)$ are independent, skewed stable random samples, and y_* is found by solving the equation

$$1 = \left\langle \left(\frac{|\chi_i|^2 S}{|z|^2 + y_*^2 |\chi_i|^2 SS'} \right)^{\alpha/2} \right\rangle_i . \tag{2.19}$$

Conventional approaches for the edge-of-chaos transition would thus conclude that heavy-tailed networks are always chaotic, ignoring the effect of exponential cutoff in practice.

For self-containment of the report, the key steps and quantities of the derivation for obtaining the spectral density of the heavy-tailed Jacobian (Eq. 2.18) from [23] are applied to the layerwise Jacobian as in Eq. 2.3 which takes form $A_{ij} := W_{ij}\chi_j$, i.e. a column-structured non-Hermitian heavy-tailed random matrix. We require the finiteness $\langle|\chi_k|^\alpha\rangle$ which is guaranteed by the sigmoidal activation function $\phi = \tanh$ and assume additionally that χ_j is independent of W_{ij}. We define $G_A(z) := (A - zI)^{-1}$ as the resolvent of the corresponding square matrix A.

Matrix resolvent

The (empirical) spectral density of non-Hermitian matrix $\hat{\rho}_A(z) := \langle\delta(z - \lambda_k(A))\rangle_k$ can be recovered via the relationship

$$\rho_A(z) = -\lim_{\eta\to 0}\frac{1}{n\pi}\partial_{\bar{z}}\sum_{i=1}^{n} H^{-1}(z,\eta)_{(i+n),i}. \tag{2.20}$$

The matrix H is a hermitization of A with the regulator η:

$$H(z,\eta) = \begin{pmatrix} -\eta I_n & A - zI_n \\ A^\dagger - \bar{z}I_n & -\eta I_n \end{pmatrix}. \tag{2.21}$$

with inverse:

$$H^{-1}(z,\eta) = G_{H(z,0)}(\eta) = \begin{pmatrix} \eta X & X(A - zI_N) \\ \left(A^\dagger - \bar{z}I_N\right)X & \eta Y \end{pmatrix} \tag{2.22}$$

where $X = \left((A - z)\left(A^\dagger - \bar{z}\right) - \eta^2\right)^{-1}$ and $Y = \left(\left(A^\dagger - \bar{z}\right)(A - z) - \eta^2\right)^{-1}$.

Quaternionic resolvent and Jacobian eigenvector localization properties

Permuting the nodes of H yields a matrix $\mathbf{A}$ with quaternionic elements

$$\mathbf{A}_{ij} = \begin{pmatrix} 0 & A_{ij} \\ A_{ji} & 0 \end{pmatrix}, \quad U(z,\eta) = \begin{pmatrix} \eta & z \\ \bar{z} & \eta \end{pmatrix}. \tag{2.23}$$

Now the cavity method can thus be applied to the quaternionic notion of the resolvent

$$\mathbf{G}_\mathbf{A}(U) = (\mathbf{A} - U \otimes I_n)^{-1}. \tag{2.24}$$

Furthermore, writing its diagonal elements as

$$\mathbf{G}_{\mathbf{A}}(U)_{ii} =: \begin{pmatrix} a_i(z,\eta) & b_i(z,\eta) \\ b_i'(z,\eta) & c_i(z,\eta) \end{pmatrix} \tag{2.25}$$

allows one to express the quaternionic resolvent in terms of the usual version: $\mathbf{G}_{\mathbf{A}}(U(z,\eta)) = G_{\mathbf{A}-U(z,0)\otimes I_n}(\eta)$. If $\eta \in i\mathbb{R}_+$, $a_i, c_i \in \mathbb{R}_+$ and $b_i' = \overline{b_i}$ [67],

$$\pi\rho_A(z) = -\lim_{t\to 0} \langle \partial_z b_i(z,it) \rangle_i \tag{2.26}$$

and the IPRs of the left and right eigenvectors of A are

$$\mathrm{IPR}^{(l)}_{q,A}(z) = N^{1-q} \lim_{t\to 0} \frac{\langle (\operatorname{Im} a_i(z,it))^q \rangle_i}{\langle \operatorname{Im} a_i(z,it) \rangle_i^q}, \quad \mathrm{IPR}^{(r)}_{q,A}(z) = N^{1-q} \lim_{t\to 0} \frac{\langle (\operatorname{Im} c_i(z,it))^q \rangle_i}{\langle \operatorname{Im} c_i(z,it) \rangle_i^q} \tag{2.27}$$

The Cavity method and resolvent recursion closure

Define A as an adjacency matrix of a weighted tree. Then for $j,k \in \partial_i$, where ∂_i are the neighbouring nodes of i. $A^{(i)}$ is the adjacency matrix of a forest of isolated trees with roots $j \in \partial_i$. If $j \neq k$, then j,k belong to distinct trees in $A^{(i)}$

$$\mathbf{G}_{\mathbf{A}}(U)_{ii} = \left(\mathbf{A}_{ii} - U - \sum_{j,k\neq i} \mathbf{A}_{ij} \mathbf{G}^{(i)}_{jk} \mathbf{A}_{ki} \right)^{-1}. \tag{2.28}$$

However, $G_A(z)_{jj} \neq G_{A^{(i)}}(z)_{jj}$ in general. Removing another node from $A^{(i)}$ brings the recursion to a second step [67, 90]:

$$\mathbf{G}^{(i)}_{jj} = \left(\mathbf{A}_{jj} - U - \sum_{k\in\partial_j\setminus i} \mathbf{A}_{jk} \mathbf{G}^{(i,j)}_{kk} \mathbf{A}_{kj} \right)^{-1} \tag{2.29}$$

where $j \in \partial_i$. This allows termination of the recursion as $\mathbf{G}^{(i,j)}_{kk} = \mathbf{G}^{(j)}_{kk}$. Moreover, $(\mathbf{G}^{(i)}_{jj})_{j\in\partial_i}$ and $(\mathbf{G}^{(j')}_{kk})_{k\in\partial_{j'}\setminus i}$ are equivalent in distribution for all $j' \in \partial_i$ and

$$\langle f(\mathbf{G}^{(i)}_{jj}) \rangle_{j\in\partial_i} = \langle f(\mathbf{G}^{(j)}_{kk}) \rangle_{k\in\partial_j\setminus i} \tag{2.30}$$

for any function f. Thus computing the diagonal elements $\mathbf{G}_{\mathbf{A}}(U)_{ii}$ of the quaternionic resolvent can be obatained by solving for the recursive distributional equation:

$$\begin{pmatrix} a_j^{(i)} & b_j^{(i)} \\ \bar{b}_j^{(i)} & c_j^{(i)} \end{pmatrix} = -\left(\begin{pmatrix} \eta & z - A_{jj} \\ \bar{z} - \overline{A_{jj}} & \eta \end{pmatrix} + \sum_{k\in\partial_j\setminus i} \begin{pmatrix} |A_{jk}|^2 c_k^{(j)} & A_{jk}A_{kj}\bar{b}_k^{(j)} \\ \overline{A_{kj}A_{jk}} b_k^{(j)} & |A_{kj}|^2 a_k^{(j)} \end{pmatrix} \right)^{-1}. \tag{2.31}$$

2.C Extended data

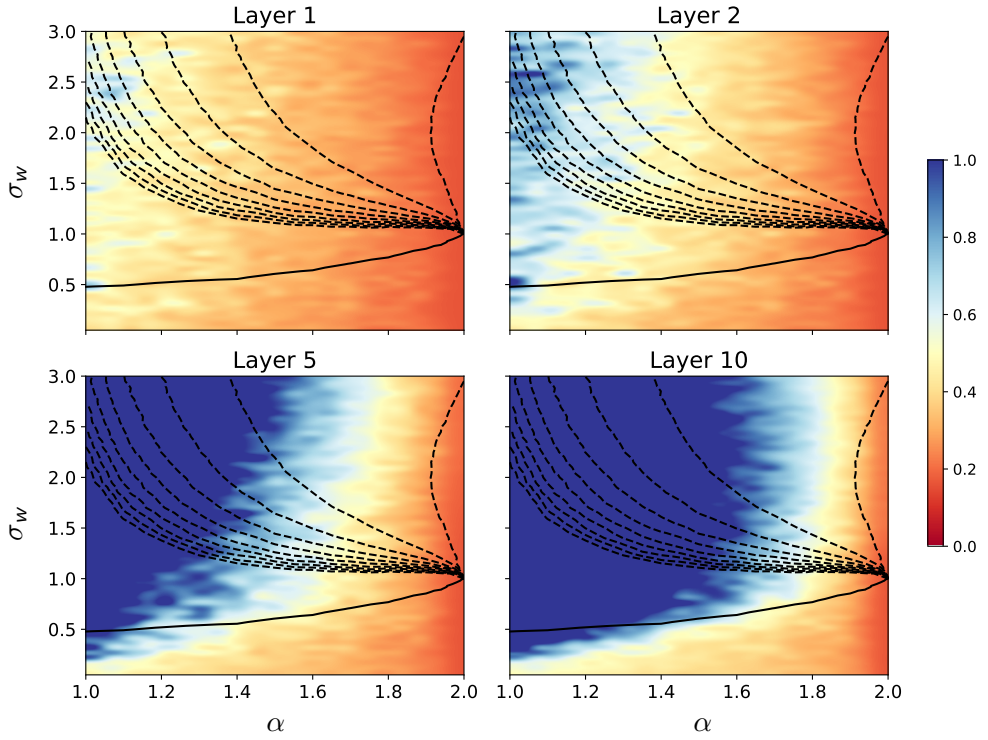

Figure 2.C.1: (Extended Data Fig. 2.3(g–i)) The extended critical regime balances contraction and expansion. (a–d) The coefficient of variation of pairwise distances as in Eq. 2.8 are reproduced for layer 1, 2, 5 and 10 respectively.

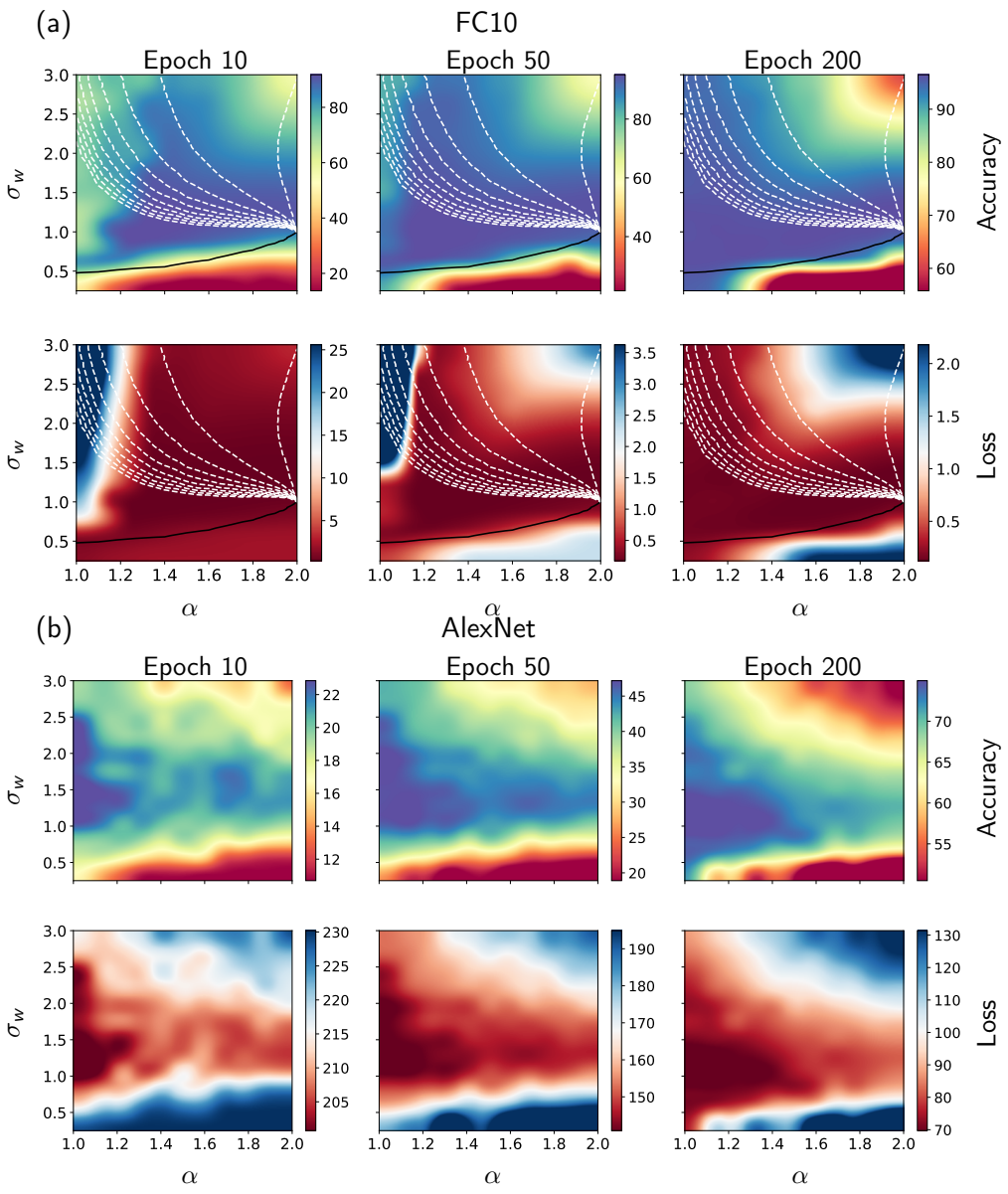

Figure 2.C.2: (Extended Data Fig. 2.4) Accuracy and loss phase transition of DNNs at different epochs. (a) Top-1 test accuracy (top row) and loss (bottom row) of FC10 at the end of epoch 10, 50 & 200 respectively. The realizations of network initialization are the same as in Fig. 2.4. (b) Same as in (a) but the train accuracy and loss of AlexNet are evaluated instead for the colormap.

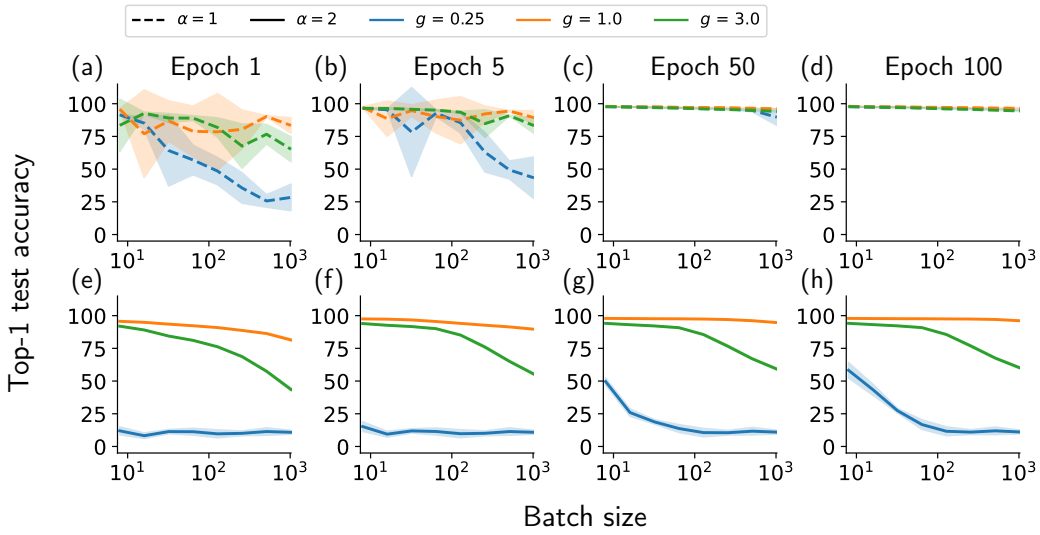

Figure 2.C.3: (Extended Data Fig. 2.4) The effect of batch size on different initializations. (a–d) The Top-1 testing accuracy of FC10 at epoch 1, 5, 50 and 100 respectively; the same training setting as in Fig. 2.4 is used. All networks are initialized at $\alpha = 1$ (dashed lines) and $\sigma_w = 0.25, 1, 3$ (blue, orange, green). Batch sizes are realized at values of $8, 16, 32, \cdots, 1024$. The accuracy for each setting of (α, σ_w) and batch size is averaged over 5 instances of training; the shaded area corresponds to the standard deviation. (e–h) The same as in (a–d) respectively except that $\alpha = 2$ (solid lines).

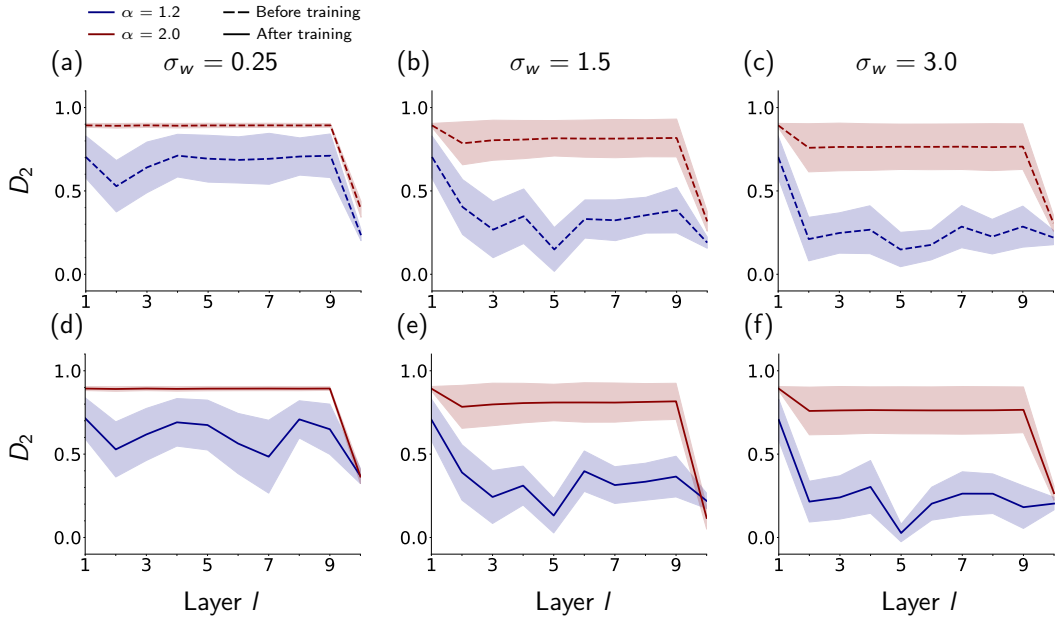

Figure 2.C.4: (Extended Data Fig. 2.5) Preactivation depth vs correlation D_2 of layerwise Jacobian eigenvectors. (a–c) The average correlation dimension D_2 of the preactivation Jacobian plotted against layer l before training. D_2 is averaged across all eigenvectors of that specific layer. The red and blue lines correspond to $\alpha = 1.2$ and $\alpha = 2$; $\sigma_2 = 0.25, 1, 3$ for (a–c) respectively. (d–f) Same as in (a–c) but after training FC10 for 650 epochs.

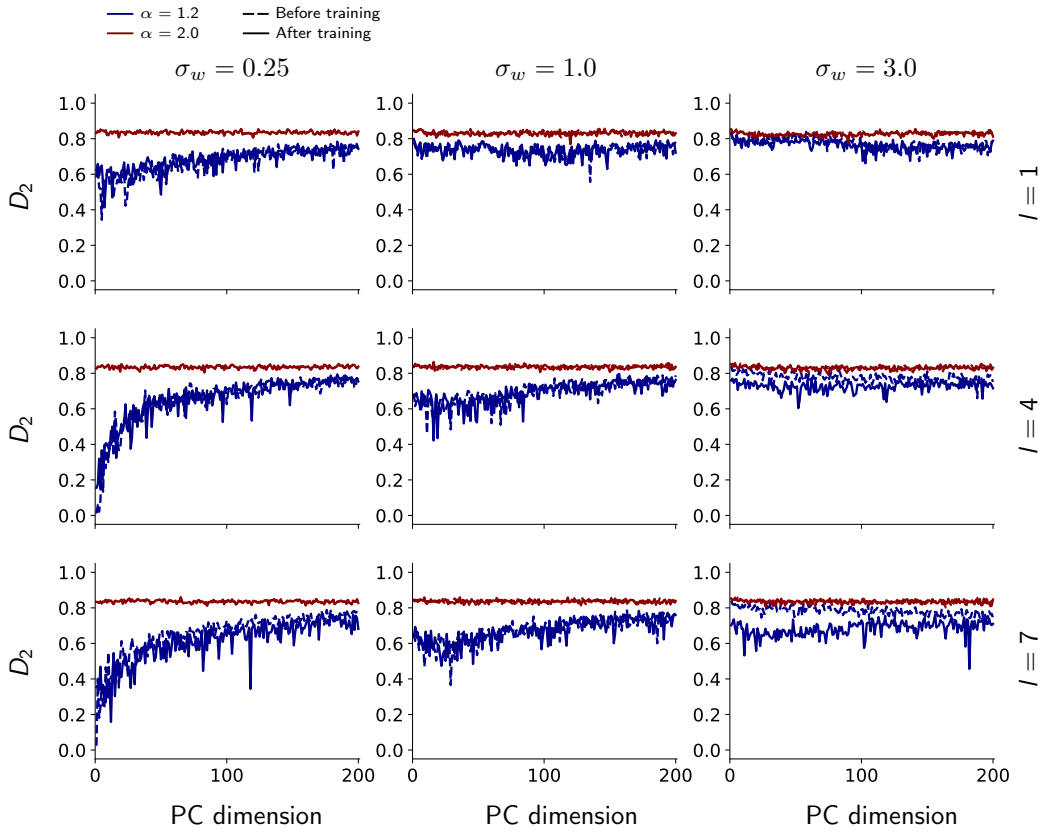

Figure 2.C.5: (Extended Data Fig. 2.6) Correlation dimension D_2 for full-batch principal directions of FC10 activations. D_2 of the top 200 principal directions of $\mathbf{x}^l$ for $l = 1, 4, 7$. The dashed and solid lines depict FC10 before training and after being trained for 650 epochs respectively.

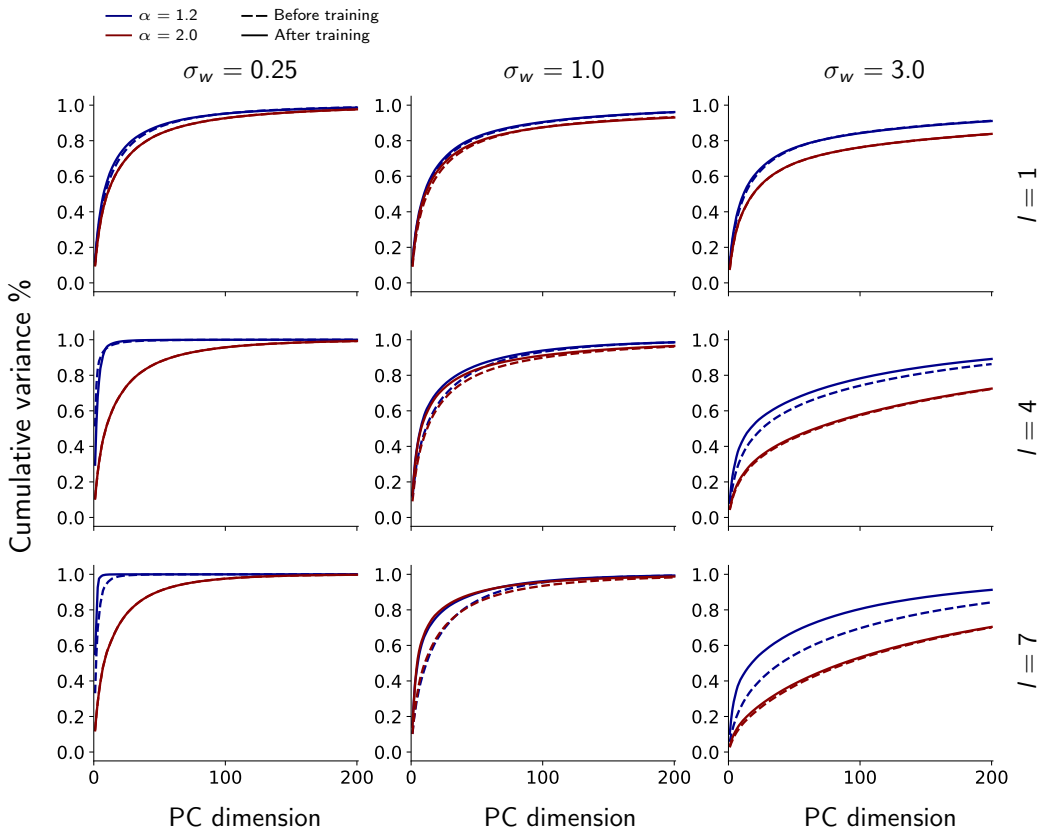

Figure 2.C.6: (Extended Data Fig. 2.6) Cumulative variance of FC10 activation principal components. The cumulative variance of the top 200 principal components of $\mathbf{h}^1$ for $l = 1, 4, 7$. The dashed and solid lines depict FC10 before training and after being trained for 650 epochs respectively.

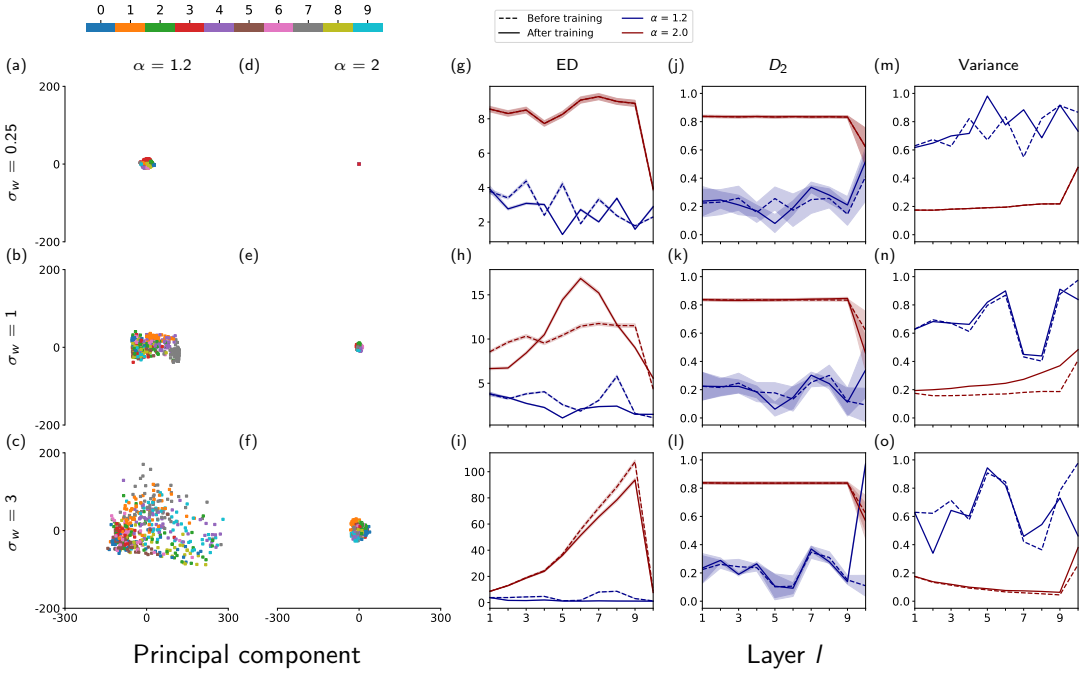

Figure 2.C.7: (Extended Data Fig. 2.6) Localization properties and effective dimensions of DNN preactivations. (a–c) The top 2 PCs of the second final preactivation layer $\mathbf{h}^l$ ($l = 9$) of a trained FC10 at epoch 650, the networks are initialized at $\alpha = 1.2$ with $\sigma_w = 0.25, 1$ and 3 respectively. Different colors represent the associated image classes of MNIST dataset. (d–f) The same as in (a–c) but for $\alpha = 2$. (g–i) The ED for each preactivation layer $\mathbf{h}^l$ is computed based on Eq. 2.12 for $\sigma_w = 0.25$, 1 and 3 respectively. The lines indicate the mean values and the shaded area corresponds to the standard deviation, both averaged over the batches of input images. The blue and red curves represent the network initialized at $\alpha = 1.2$ and $\alpha = 2.0$ respectively. Dashed curves are computed based on FC10 before training; solid curves after the networks have been trained for 650 epochs. (j–l) The average correlation dimension D_2 of the top 5 principal directions is plotted over depth; the neural representations are obtained from the full-batch MNIST train dataset. (m–o) Percentage of the cumulative variance explained by the top 5 PCs plotted against layer l. Neural representations are obtained as in (j–l).

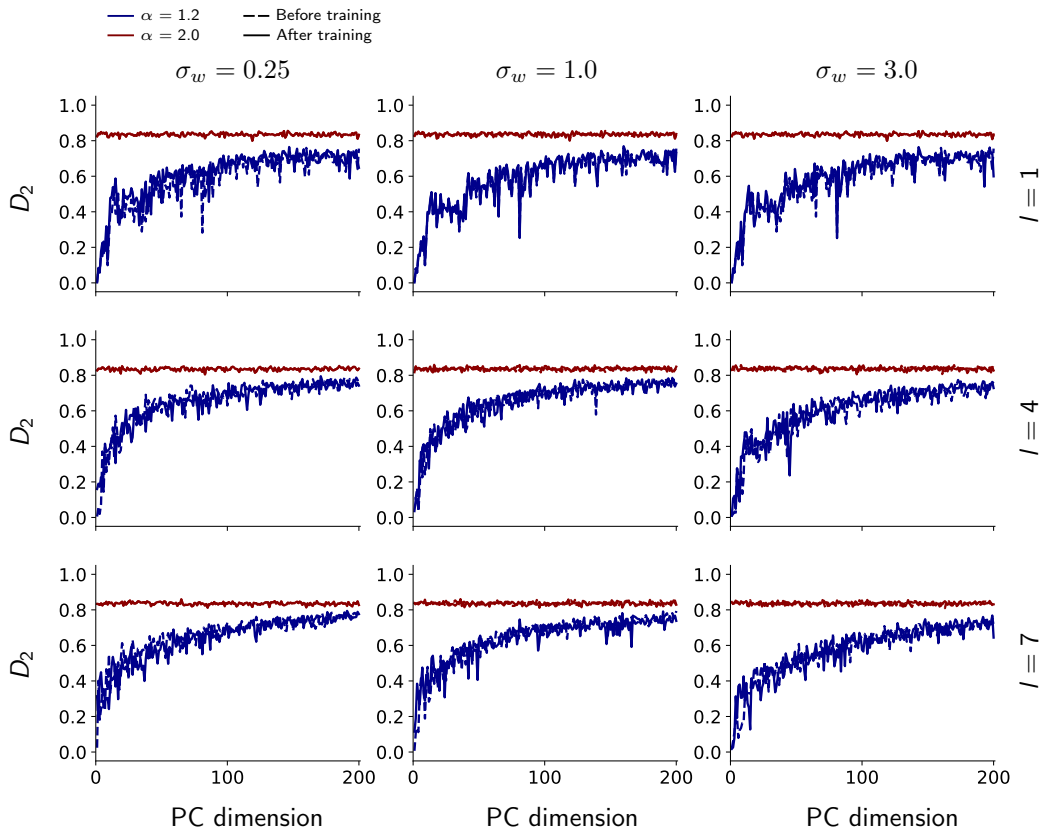

Figure 2.C.8: (Extended Data Fig. 2.6) Correlation dimension D_2 for full-batch principal directions of FC10 preactivations. D_2 of the top 200 principal directions of $\mathbf{h}^1$ for $l = 1, 4, 7$. The dashed and solid lines depict FC10 before training and after being trained for 650 epochs respectively.

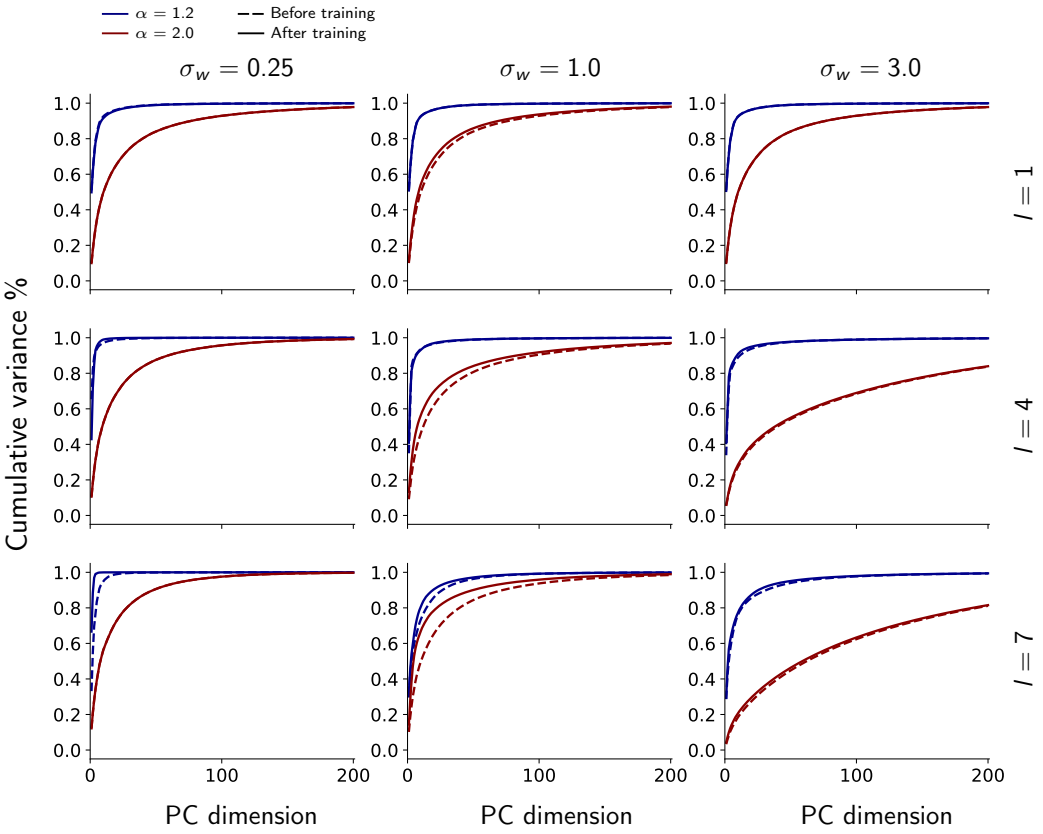

Figure 2.C.9: (Extended Data Fig. 2.6) Cumulative variance of FC10 preactivation principal components. The cumulative variance of the top 200 principal components of $\mathbf{h}^1$ for $l = 1, 4, 7$. The dashed and solid lines depict FC10 before training and after being trained for 650 epochs respectively.

Chapter 3

A Fractional Diffusion implementation of self-attention in transformers

Abstract

The success of transformers in sequential modeling has inspired the search for architectures that are more interpretable and align with biological principles. A recent work on *Fractional Neural Sampling* (FNS) has used stochastic differential equations (SDEs) driven by Lévy processes to characterized neural circuits and sampling in visual-spatial attention for cognitive functions. Motivated by this, we integrate the corresponding fractional diffusion component into the attention mechanism of transformers and refer to it as *fractional self-attention* (FSA). Under appropriate constructions, FSA can be viewed as a discretization of the fractional Laplacian with order α, which is precisely the infinitesimal generator of a symmetric Lévy-α stable process. The parameter α controls the jump size distribution of this underlying stochastic process implicitly generated by self-attention and the non-Gaussian case ($\alpha < 2$) encourages long-range spatial interactions in the embedding space. Though utilizing tools from geometric harmonics and stochastic analysis, we construct a random walk counterpart of fractional diffusion dynamics on a manifold. The model includes a kernel bandwidth hyperparameter ε which naturally introduces the notion of a continuous time scale, we also investigate how α affects the model performance under various settings. Under small time scales, tools from diffusion maps can additionally be leveraged to obtain dimensionality reduction that respects the underlying geometry and

distribution of the initial embeddings and hidden states.

3.1 Introduction

Transformers have achieved great success in processing sequential data [39] including natural language processing (NLP) [91, 92] and general time series [93, 94]. Beyond NLP, their applications extend to other domains such as computer vision [95] and graph learning tasks [96]. Self-attention, the core component of Transformer models, provides endless possibilities for re-innovation. It operates within a feedforward framework, allowing simultaneous processing of an entire n-token sequence [97] without relying on traditional recurrent neural network structures [35]. Consequently, the computation in self-attention scales in quadratic complexity with respect to n, i.e. $\mathcal{O}(n^2)$ in big-O notation. This has spurred considerable recent research focused on reducing computational complexity to handle increasingly long sequences, reaching substantial progress [98, 99, 100, 27]. Moreover, the specific mechanisms within self-attention that drive its effectiveness remain largely unclarified. Along this direction, a few other studies have treated sequence embeddings that contain semantic and positional information as spatial variables [101, 6], formulating the self-attention mapping as a continuous-time dynamical system under an infinitesimal time and infinite sequence ($n \to \infty$) limit. In particular, the bi-stochastic self-attention has limiting second-order dynamics described by the heat equation under specific conditions, which underlies Brownian motion [6]. Despite these advancements, the neurobiological relevance of newly developed or improved Transformer models is often overlooked. For instance, Brownian motion has been observed to be inefficient for sampling and unrealistic for representing brain dynamics. The newly established fractional neural sampling (FNS) theory instead employs stochastic differential equations (SDEs) driven by Lévy motion to model population activity patterns emerging from spiking neural circuits [31]. Under the FNS framework, spatiotemporal dynamics richer than Brownian motion emerge, presenting a more realistic and efficient foundation for (visual) attention sampling [102]. Similar fractional dynamics have also been observed in neural circuits [26] which have significant implications for cognitive functions such as visual-spatial attention [102, 30].

In this study, we construct a novel mechanism termed *Fractional Self-Attention* (FSA) that integrates the fractional diffusion component of the FNS theory into self-attention by adopting a dynamical system formulation. We define models utilizing this mechanism as *Fracformers*, establishing a new class of architectures that leverage principles underlying neuroscience to advance artificial intelligence [103]. A distinction between FSA and the original architecture involves replacing the dot-product between the queries and keys with a (geodesic) distance-based

metric. For the integration of fractional diffusion, a power-law kernel must be used in place of the original exponential kernel. Further conditions on the query and key projection matrices such as weight-tying and orthogonality are required only for relevant theoretical derivations However, these constraints remain fully implementable during the training process without limiting practical applicability.

3.2 Results

Motivated by the stochastic dynamics underlying the theory of *Fractional Neural Sampling* (FNS) established as a realistic construct for dynamics emerging in neural circuits, we establish a new Fractional Self-Attention (FSA). Theoretically, we prove that the second-order dynamics of the associated mapping converge to an equation involving the fractional Laplacian, $(-\Delta)^{\alpha/2}$ [33]. This non-local, pseudo-differential operator with fractional order $\alpha \in (1, 2]$ connects FSA to stochastic dynamics on a manifold. In particular, the fractional Laplacian is precisely the infinitesimal generator of a Symmetric α-stable ($S\alpha S$) Lévy process [104] which possesses heavy-tailed jump sizes spatially. Such large jumps for $\alpha < 2$ have been observed in visual-spatial attention sampling in the brain [30]. In the context of Transformer attention, they encourage important long-range interactions between tokens that are geometrically distant from each other in the embedding space. When $\alpha = 2$, dynamics of the Brownian motion are recovered. Our construction of FSA also connects to geometric harmonics. In particular, a canonical eigendecomposition can be performed on the attention weights through the lens of diffusion maps [47, 105], improving the interpretability of the attention score. Additionally, it enables a dimensionality reduction of the attention score that respects the underlying geometry of the embedding space as well as the stochastic dynamics corresponding to the diffusion process.

3.3 Self-attention map as a dynamical system

The Transformer model employs self-attention mechanisms [97] in a feed-forward manner as the key structure for processing information [39]. In this section, we dissect the SA mechanism as a dynamical system. We establish its first-order dynamics as a discretized form of an ordinary differential equation (ODE) under appropriate assumptions, offering a continuous space-time perspective. Extending this analysis enables us to examine its second-order dynamics, uncovering deeper probabilistic foundations within the system's behavior.

We first summarize the self-attention mechanism. The self-attention block in the $(l+1)^{\text{th}}$ layer receives as input a sequence of n tokens embedded in a manifold such as the Euclidean space, i.e., $(x_1^l, x_2^l, \cdots, x_n^l)$ where $x_i^l \in \mathbb{R}^d$. The self-attention

block projects these embedding vectors to queries, keys and values via $q = W_Q x$, $k = W_K x$ and $v = W_V x$, respectively, where $W_Q, W_K \in \mathbb{R}^{d_k \times d}$ and $W_V \in \mathbb{R}^{d_v \times d}$ are learnable projection matrices. For simplicity, we assume $d_k = d_v = d$ unless otherwise stated. The self-attention block then computes as output the new embedding vector for each token using information across the entire input sequence [106, 107, 6, 108]:

$$x_i^{l+1} = x_i^l + \sum_{j=1}^{n} K_{ij} W_V x_j^l, \tag{3.1}$$

where $K = \text{softmax}(C) := N_R(\exp[C])$. Here, the exponential function is applied element-wise to the similarity matrix $C \in \mathbb{R}^{n \times n}$. The entries of the similarity matrix are calculated as the dot products of the queries and keys, $C_{ij} := q_i^\top k_j = x_i^\top W_Q^\top W_K x_j$. Furthermore, $N_R(\cdot)$ represents an operation that row-normalizes a matrix, resulting in a right-stochastic matrix that represents the Markovian transition probability from one token, v_i, to another, v_j, in a single step: $K_{ij} \equiv \mathbb{P}(V(1) = v_j | V(0) = v_i)$. Here V is the value matrix with row i as $v_i := W_V x_i$. Additionally, we use $N_C(\cdot)$ to denote the column-normalization counterpart.

With the success of deep models such as GPT-3, which features up to 96 layers [3], understanding the theoretical limiting behavior of self-attention in equation 3.1 becomes increasingly important. We now interpret the self-attention mechanism as a dynamical system by taking the following limits in order: 1. Infinite particle limit, $n \to \infty$. 2. Infinitesimal step size, $\varepsilon \to 0$. Through the residual form of the self-attention in equation 3.1, we can establish it as a specific realization of the form

$$x_i^{l+1} = x_i^l + T(x_i^l) \tag{3.2}$$

which also corresponds to an Euler discretization with step-size 1 of the ODEs:

$$\dot{x_i}(t) = T(x_i(t)) \text{ for all } i. \tag{3.3}$$

Here, token embeddings x_i's are treated as spatial variables, while the layer l in equations 3.1 and 3.2 as a time variable t. To bridge the connection between discrete and continuous dynamics, the (push-forward) mapping T is constructed as

$$T_{\mu,\varepsilon}(x) = \int_{\mathbb{R}^d} k_\varepsilon(x, x') W_V x' \mathrm{d}\mu(x'), \tag{3.4}$$

where $k_\varepsilon(x, x') := \frac{\exp(x^\top Q^\top K x')/\varepsilon}{\int_{\mathbb{R}^d} \exp(x^\top Q^\top K x'/\varepsilon) \mathrm{d}\mu(x')}$. Here, the underlying measure μ of the particles is time-dependent and μ_0 is the initial distribution at time $t = 0$ (or equivalently at layer $l = 0$). The mapping T depends on the density $\mu \in \mathcal{P}(\mathbb{R}^d)$, from which the embeddings x' are sampled and can be identified with a density function ρ of the particle (token) positions, i.e. $\mathrm{d}\mu(x) = \rho(x)\mathrm{d}x$. The bandwidth

parameter ε which represents an infinitesimal step size serves as an important component for proving that the ODEs (equation 3.3) can be written as a continuity equation describing the second-order dynamics [109]:

$$\begin{cases} \partial_t \mu + \operatorname{div}(\mu T_\mu) = 0 & (t, x) \in \mathbb{R}^+ \times \mathbb{R}^d \\ \mu|_{t=0} = \mu_0 & x \in \mathbb{R}^d, \end{cases} \tag{3.5}$$

where $T_\mu := T_\mu|_{\varepsilon=1}$ (and similarly $k := k|_{\varepsilon=1}$). The continuity equation provides a deeper understanding of the evolution of the probability measure underlying the particles. Upon setting $\mu_t := \frac{1}{n}\delta_{x_i(t)}$ due to the permutation-equivariance of self-attention, and $\varepsilon = 1$ in the equation 3.4, the original self-attention (equation 3.1) can be recovered from equation 3.2 [106, 107, 110].

A recent work [6] considered a bi-stochastic form of self-attention, where an iterative scheme of column (N_C) and row normalization (N_R) is applied to the attention weights K. Under this construction and certain conditions on the projection matrices (i.e., $W_K^\top W_Q = W_Q^\top W_K = -W_V$), equation 3.5 reduces to the spatially local heat equation [6]:

$$\partial_t \rho - \Delta \rho = 0. \tag{3.6}$$

Note that $\rho := \rho(t, x) \in \mathbb{R}^+ \times \mathbb{R}^d$ is the infinitesimal generator of Brownian motion with the initial condition $\rho(0, x) = \rho_0(x)$. Based on the above formulation, the connection between self-attention mapping and the continuity equation as in equation 3.5 is established.

In our work, inspired by the stochastic dynamics of Lévy processes underlying fractional neural sampling (FNS) theory for brain dynamics, we extend the functional advantages of a non-local fractional diffusion equation to a novel self-attention mechanism which further generalizes equation 3.6. By carefully modifying k_ε in equation 3.4, we re-design the self-attention mapping to incorporate the spatially non-local fractional heat equation on a manifold (not necessarily Euclidean), yielding a wider class of diffusion processes. Such non-local dynamics have been found in a wide range of neurobiological systems [26, 30] and neural circuits specifically based on the theory of FNS [31]. Drawing inspiration from these studies, we realize *Fractional Self-Attention* (FSA) as a discretized counterpart of these continuous-time systems, enabling tokens to interact in a fractional-diffusive-like manner.

3.4 Fractional diffusion

In this section, we first introduce the fractional heat equation (equation 3.7), where, combined with equation 3.6, allows us to draw connections to two diffusion

process types that could occur on a manifold, highlighting a dichotomy in their corresponding spatial profiles. In particular, we showcase spatially local and non-local heat kernels induced by the Heat Kernel Dichotomy [46], and elaborate on their relation to Lévy α-stable processes, governed by the fractional order α in equation 3.7. For $\alpha < 2$, the heat kernel is non-local and is associated with the α-stable Lévy process. When $\alpha = 2$, a scaled Brownian motion $\sqrt{2}B_t$ is recovered. These stochastic dynamics are not inherent to self-attention itself, but arise from the choice of kernels applied to the token embeddings, influencing the attention weights K in equation 3.1 and thus their spatial interactions.

3.4.1 Local and non-local heat kernels

We begin by considering a spatially non-local generalization of equation 3.6 known as the fractional heat equation on a closed and compact manifold $\mathcal{M}$.

$$\partial_t \rho + (-\Delta_{\mathcal{M}})^{\alpha/2} \rho = 0. \tag{3.7}$$

where fractional Laplacian $(-\Delta_{\mathcal{M}})^{\alpha/2}$ is a non-local, pseudo-differential operator and $\alpha \in [1, 2]$ is its corresponding fractional order. There are several equivalent definitions of the fractional Laplacian, which we will defer to the references cited herein for further details [111, 33]. For this work, we focus on the case $\mathcal{M} = \mathbb{R}^d$ (written as $\Delta := \Delta_{\mathbb{R}^d}$), and the operator is defined by a singular integral operator as the following

$$(-\Delta)^{\alpha/2} u(x) = c_{d,\alpha} \lim_{\varepsilon \downarrow 0} \int_{\mathbb{R}^d \setminus B(x,\varepsilon)} \frac{u(x) - u(y)}{|x - y|^{d+\alpha}} \mathrm{d}y, \tag{3.8}$$

where $c_{d,\alpha} := \frac{\alpha 2^{\alpha-1} \Gamma\left(\frac{d+\alpha}{2}\right)}{\pi^{d/2} \Gamma(1-\alpha/2)}$ is a normalization constant and Γ is the Gamma function; $B(x, \varepsilon)$ is the open ball of radius ε centered at x. When $\alpha = 2$, the operator reduces to the standard Laplacian as defined in equation 3.6. To illustrate the effects of fractional diffusion, we demonstrate in Fig. 1(a) the fractional Laplacian applied to the (centered) Gaussian function $f(x) := f(x; 0, 1)$ (we denote $f(x; \mu, \sigma)$ as the probability density function of $\mathcal{N}(\mu, \sigma)$) for various values of α, comparing the transformed functions $(-\Delta)^{\alpha/2} f$. Unlike the standard Laplacian, the fractional counterpart applied to the function at x depends on the entire domain, rather than just the local neighborhood around the point as indicated in equation 3.8.

The solution to equation 3.7 is related to an important class of heat kernels known as the α-scale invariant heat kernel k which depends on the geodesic distance $d_g(\cdot, \cdot)$ on the manifold (for $\mathcal{M} = \mathbb{R}^d$, $d_g(x, y) = |x - y|$). In particular, it can be bounded asymptotically:

$$\frac{c_1}{t^{d/\alpha}} \Phi\left(C_1 \frac{|x-y|}{t^{1/\alpha}}\right) \leq k_t(x, y) \leq \frac{c_2}{t^{d/\alpha}} \Phi\left(C_2 \frac{|x-y|}{t^{1/\alpha}}\right), \tag{3.9}$$

where c_1, c_2, C_1, C_2 are all positive constants and $\Phi : [0, \infty) \to [0, \infty)$ is monotone decreasing. Under technical conditions, such class of heat kernels is either *local* or *non-local*. This seminal result is known as the Heat Kernel Dichotomy [46], stating that given the intrinsic dimension of the manifold $d_{\mathcal{M}}$, for $\alpha \leq d_{\mathcal{M}} + 1$, either

1. k is local, i.e. $\alpha \geq 2$ and equation 3.9 holds with

$$\Phi(z) = \exp\left(-z^{\frac{\alpha}{\alpha-1}}\right); \tag{3.10}$$

or

2. k is non-local, i.e. $\alpha \in (0, 2)$ and equation 3.9 holds with

$$\Phi(z) = (1+z)^{-(d_{\mathcal{M}}+\alpha)}. \tag{3.11}$$

Here the parameter α is also referred to as the walk dimension of the heat kernel which aligns with the fractional order in equation 3.7. Additionally, the intrinsic dimension $d_{\mathcal{M}}$ depends on the underlying manifold, e.g. $d_{\mathbb{R}^d} = d$ and $d_{\mathbb{S}^d} = d - 1$ for the d-sphere manifold $\mathbb{S}^d := \left\{x \in \mathbb{R}^{d+1} : |x| = 1\right\}$.

For the classical case $\alpha = 2$[1], the solution to equation 3.7 involves the well-known Gaussian heat kernel in $\mathbb{R}^d$:

$$k_t(x, y) := \frac{1}{(4\pi t)^{d/2}} e^{-\frac{|x-y|^2}{4t}}. \tag{3.12}$$

In non-local case where $\alpha \in [1, 2)$, the corresponding kernel satisfies equation 3.9 with Φ as in equation 3.11. For example, the heat kernel on $\mathbb{R}^d$ when $\alpha = 1$ is given by the Poisson kernel:

$$k_t(x, y) = \frac{\tilde{c}_d}{t^d}\left(1 + \frac{|x-y|}{t^2}\right)^{-\frac{d+1}{2}}, \tag{3.13}$$

where $\tilde{c}_d = \Gamma\left(\frac{d+1}{2}\right)/\pi^{(d+1)/2}$, remains consistent with the Heat Kernel Dichotomy.

The non-local kernel (equation 3.11) explicitly depends on the intrinsic dimension d for $\mathcal{M} = \mathbb{R}^d$, which, as shown later in Section 3.5, aligns with either the embedding or hidden layer dimension. In contrast, the local kernel (equation 3.10) lacks such dependence. Consequently, higher-dimensional spaces lead to a faster decay, making increasingly smaller distinctions between values $\alpha < 2$ as shown in Fig. 1(b). Due to the monotonicity of Φ, non-local kernels always intersect with the local kernel at a single point which progressively becomes larger with respect to the dimensionality d (Fig. 1(c)). For a more in-depth discussion on the fractional Laplacian and its connection to the α-scale invariant heat kernel (equation 3.9), refer to Appendix 3.A.2. Additionally, we provide a detailed analysis of the specific case where $\mathcal{M} = \mathbb{S}^{d-1}$.

[1] An extra constant term is needed for equation 3.6, i.e. $\partial_t u = \frac{1}{2}\Delta_{\mathcal{M}} u$.

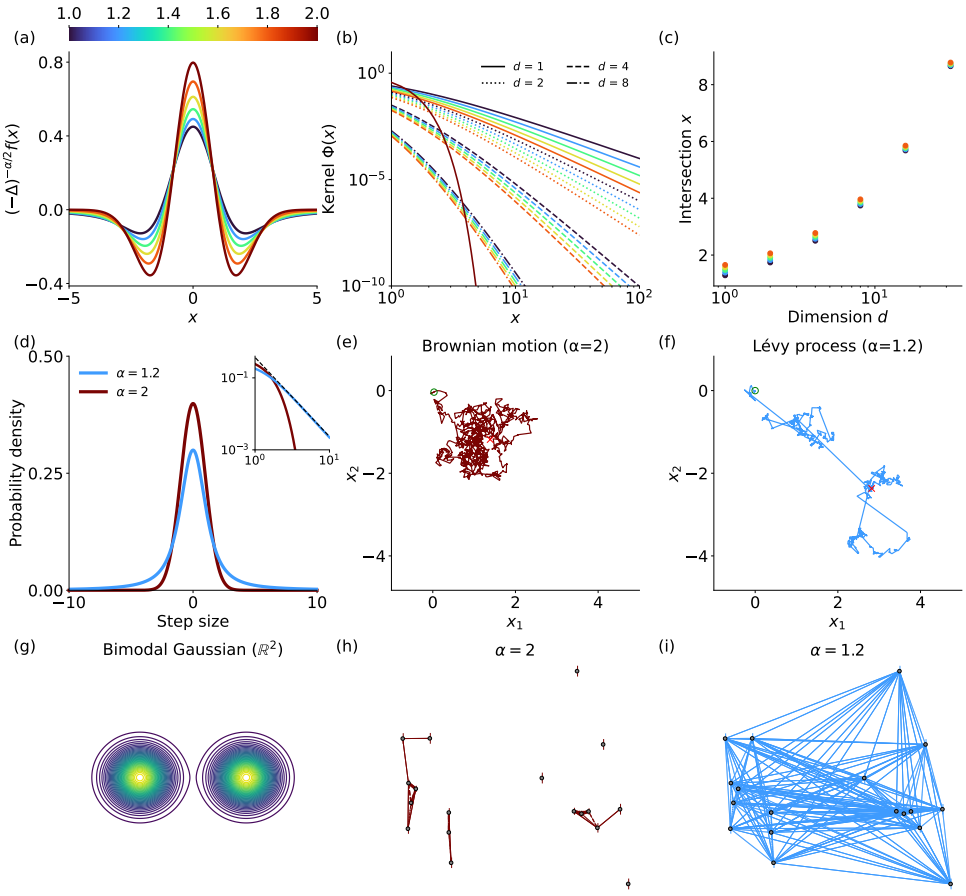

Figure 1: Fractional Laplacian, heat kernels and Lévy processes. (a) Fractional Laplacian applied to the standard Gaussian probability density function $f(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ for $\alpha = 1, 1.2, 1.4, \cdots, 2$ with colors represented in the colorbar. (b) The non-local kernel (equation 3.11) for $\alpha = 1, 1.2, 1.4, \cdots, 1.8$ and $d = 1, 2, 4, 8, 16$, and the local kernel (equation 3.10) on a log-log plot. (c) The intersection of the non-local kernel with the local kernel for $\alpha = 1, 1.2, 1.4, \cdots, 1.8$ and $d = 1, 2, 4, 8, 16, 32$. (d) The probability density function of a Gaussian (red) and Lévy α-stable distribution (blue). Inset zooms in on the right tail on a log-log plot with the black dashed lines indicating the tail asymptotics as in equation 3.14. (e) A sample path of Brownian motion on $\mathbb{R}^2$, up to terminal time $T = 10$ simulated with 1000 steps. The green cross and red circle markers are the starting ($(x_1, x_2) = (0, 0)$) and ending points of the process respectively. (f) Same as in (a) but for the sample path of a $S\alpha S$ Lévy process with stability parameter $\alpha = 1.2$. (g) The contour plot of a bi-modal Gaussian distribution centered at $(-3, 0)$ and $(3, 0)$ with the covariance matrix $I/2$. (h) Assuming equivalence between the queries and keys (Assumption 2). The thickness of the dashed lines reflects the strength of attention weights between connected queries and keys, scaled by a constant, based on FSA (equation 3.16) for $\alpha = 2$. (i) Same as in (h) but for $\alpha = 1.2$.

3.4.2 Lévy process

An important feature of the fractional heat equation (equation 3.7) is its role in describing the evolution of probability densities associated with a significant class of stochastic processes—Lévy processes. We now establish this connection by demonstrating how the α-scale invariant kernels (equation 3.9) serve as the corresponding transition densities across space and time. First, we highlight the difference in statistical properties between the Gaussian ($\alpha = 2$) and (non-Gaussian) heavy-tailed cases ($\alpha \neq 2$). Based on the Heat Kernel Dichotomy discussed earlier, a well-behaved heat kernel (see Appendix 3.A.2) can be either spatially local or non-local. It is well-known that the Laplacian is the infinitesimal generator of Brownian motion, i.e. its time-dependent probability density function is a solution to equation 3.7 where $\alpha = 2$. Analogously, the fractional Laplacian is the infinitesimal generator for a $S\alpha S$ Lévy process [104], which we will simply refer to as a Lévy process herein unless otherwise stated.

Given a random variable X drawn from a stable distribution (see Appendix 3.A.3), we denote it by $X \sim S_\alpha(\beta, \sigma, \mu)$ or $X \sim S_\alpha(\sigma)$ if it is symmetric and centered with $\beta = \mu = 0$ [32]. For infinitesimal time steps Δt, the transition probability for a Lévy process ($\alpha < 2$) in $\mathbb{R}$ follows an α-scale invariant kernel (equation 3.9). In particular, the corresponding increments $\Delta x := L_{t+\Delta t} - L_t \sim S_\alpha((\Delta t)^{1/\alpha})$ exhibit an asymptotic power-law:

$$k_{\Delta t}(x, x + \Delta x) \sim c_\alpha \Delta t |\Delta x|^{-1-\alpha}, \tag{3.14}$$

where $c_\alpha := \Gamma(1 + \alpha) \sin(\pi\alpha/2)/\pi$ is a normalization constant. The slower asymptotic decay of Lévy increments allow extreme jump sizes to occur at a higher probability (Fig. 1(d)). Therefore, unlike Brownian motion (Fig. 1(e)), the sample paths of a Lévy process (Fig. 1(f)) contain sudden jumps of varying sizes, demonstrating more irregularities in its motion. For more details on Lévy processes, please see Appendix 3.A.2.

So far, we have introduced an extended class of stochastic processes that exhibit richer dynamics beyond Brownian motion. As demonstrated by the FNS theory, the heavy-tailed regime is more biologically realistic and efficient for neural circuits [31] and spatial-visual sampling [30] compared to Brownian motion. In the next section, we integrate these statistical mechanisms into Transformers by strategically modifying the self-attention map (equation 3.1), leveraging the key concepts discussed above.

3.5 Fracformer

We now integrate these fractional diffusion dynamics in our construction of the fractional self-attention (FSA). The idea is based on the fractional heat kernel

driving the interactions between the queries and keys on the underlying manifold. Since each token embedding represents a high-dimensional state $x_i \in \mathbb{R}^d$ or, more generally, $x_i \in \mathcal{M}$ with intrinsic dimension $d_{\mathcal{M}}$, we assume that each token embedding x_i is sampled from an unknown probability measure $\mu \in \mathcal{P}(\mathcal{M})$. As demonstrated in Sec. 3.3, the residual form of the self-attention mechanism represents a transition of the token embeddings, leading to a local operator under certain assumptions [6]. Rather than using the dot product, which does not preserve the geometrical structure of the underlying distribution of token embeddings, we use a distance-dependent metric $\tilde{C}_{ij}$ to quantify the similarity between tokens. This in turn allows us to realize a fractional diffusion according to the setup in Section 3.3 where either the local or non-local kernels (equations 3.10, 3.11 resp.) is applied to the geodesic distance d_g between the token embeddings. With additional conditions, $\tilde{C}_{ij}$ exhibits more desirable properties such as Lipschitz continuity compared to the dot product C used to obtain K in equation 3.1 [27, 43]. A difference worth pointing out is that a larger value of C_{ij} now indicates a weaker similarity, whereas this is the opposite when using the dot product. Via an isometric embedding of the tokens (or hidden states), we introduce spatially non-local diffusion dynamics into self-attention and theoretically show that the transition of the values V involves the fractional Laplacian as discussed in Section 3.4.

3.5.1 Geometry of embeddings

In our implementation of fractional diffusion, we preserve various geometrical properties of the token embeddings. To accomplish this, we must generalize the definition of the query projection as $q_i := \iota(x_i)$, where $\iota : \mathcal{M} \to \mathbb{R}^d$. While the original self-attention mechanism applies a linear query projection through $\iota(x_i) = W_Q x_i$, and similarly for the key projection, this does not preserve geometrical properties such as the distance between embeddings. Thus, an isometric embedding is desirable in this situation. For an unknown $\mathcal{M}$, the geodesic distance d_g can only be estimated with the presence of many data points (see Appendix 3.A.2). However, embedding x_i's onto the Euclidean space $\mathbb{R}^d$ (or d-sphere $\mathbb{S}^{d-1}$ through a simple L^2-normalization of the hidden states) can inherently eliminate the need for this estimation (see Methods). An isometric embedding of $\mathcal{M} = \mathbb{R}^d$ can be exactly realized for $\iota(x)$ under the following assumptions. **Assumption 1**: Orthogonality of the projection matrix ($W_{Q,K} \in O(d)$) and single attention-head $H = 1$. For general deep learning purposes, the orthogonality of a trainable weight matrix can be enforced even during training by projecting the natural gradients with respect to the loss from the Stiefel manifold onto the Euclidean space [112]. This condition has also been incorporated into other architectural designs for Transformers to maintain the token embedding manifold structure and thereby preserve the injectivity of the model input-output mapping [113]. **Assumption**

2: Weight-tying preserving the symmetry between the queries and keys ($Q = K$). This simple condition has motivated the development of new architectures [27]. Although enforcing this condition could potentially reduce model expressivity. On one hand, this issue can be mitigated by increasing the number of attention block layers [43]; on the other, the condition is optional for the general implementation of FSA. The main importance of weight-tying between W_Q and W_K allows the fractional diffusion limit of FSA to be derived in the limit. In a series of recent studies [108], various types of assumptions on the key and query are imposed to theoretically understand the mechanisms of self-attention such as $Q^\top K \succ 0$ (positive-definiteness) and $Q^\top K = I$, etc.

When both assumptions are satisfied, we effectively have $W_Q = W_K = I$ and the Euclidean distance is preserved, i.e. $|q_i - k_j| = |x_i - x_j|$. Additionally, they are essential for introducing a rotation-invariant heat kernel (see Appendix 3.A.2). To this end, we now construct the fractional self-attention block. Building on the previous section, we define an alternative construction of a stochastic matrix that is eventually applied to the values. For $1 \leq i, j \leq n$, we define the attention score $K^{(0)}$ based on the construction in equation 3.15. When the Assumption 1 is guaranteed, we can also reduce the parameter of the Fracformer through the following proposition.

Proposition 1. *For single-head attention and $W_{Q,K} \in O(d)$, for $\mathcal{M} = \mathbb{R}^2$ or $\mathbb{S}^{d-1}$. This is equivalent to having a projection matrices $W_Q = I$ and $W_K = W_Q^\top W_K$.*

We present the simple proof of this proposition in Appendix 3.A.4. The geodesic distance between arbitrary x_i, x_j is generally not preserved, nonetheless, we have achieved a meaningful and justified parameter reduction for FSA under the case of Assumption 1.

3.5.2 Fractional self-attention

Suppose that we have n token embeddings $(x_1, x_2, \cdots, x_n)$ sampled from a manifold $\mathcal{M}$ according to some probability measure $\mathrm{d}\mu_0(x) := \rho_0(x)\mathrm{dvol}_x$. Such settings have been explored in algorithms like (fractional) diffusion maps [47, 48, 105] for dimensionality reduction, with which our approach shares parallelisms. While we do not aim to project these tokens onto a lower-dimensional space, we propose constructing fractional self-attention, which asymptotically converges to operators linked to Lévy processes under Assumption 1 & 2.

1. Define the entries of the diffusion matrix K_{ij} based on all pairs of points x_i, x_j in the dataset

$$\tilde{C}_{ij} = \Phi\left(\varepsilon^{-1}\frac{|W_Q x_i - W_K x_j|}{\kappa}\right) \tag{3.15}$$

for Φ as equation 3.10 or 3.11 depending on α. To address the dependence of the non-local kernel on d, we introduce an alternative d-independent non-local kernel $\tilde{\Phi}$ (equation 3.23) for $\alpha < 2$, which will be used in the above for large embedding dimensions (see Methods). Unless otherwise stated, we set $\kappa = \sqrt{d/H}$ as a distant-scaling constant. We also impose conditions on $W_{Q,K}$ as discussed in Section 3.5.1.

2. The FSA block output is the transition of the values by the row-normalization of equation 3.15

$$\mathrm{FAttn}(X) := KV^\top, \;\; K := N_R(\tilde{C}). \tag{3.16}$$

The kernel bandwidth ε of the (fractional) heat kernel k_ε also acts as a small time increment Δt. The scaling constant $\sqrt{d/H}$ appears in the original dot-product self-attention [39]. Additionally, for $\alpha = 2$ our definition overlaps with that in [43]. For simplicity of the derivation in Theorem 1, the constant is set to $\kappa = 1$. Before training, we always set $\varepsilon = 1$, nonetheless, the inclusion of ε remains important for the derivation of Theorem 1 later on. Additionally, the spatial-continuous analog can be recovered through the integral representations in equations 3.37–3.41, Appendix 3.A.6. The hyperparameter α requires specification before training.

Diffusion maps are renowned for their effectiveness in non-linear dimensionality reduction [47, 48]. The typical setup is that the underlying manifold as well as the sampling density is unknown apriori, hence the geodesic distance must be estimated, for instance through equation 3.30 in Appendix 3.A.2. However, introducing a non-local heat kernel becomes feasible without this extra step by explicitly representing the hidden states (embedding outputs) from a known manifold such as $\mathbb{R}^d$ or $\mathbb{S}^{d-1}$.

Let us define the operator

$$T_{\mu_0,\varepsilon}[f](x) = \int_{\mathcal{M}} k_\varepsilon(x,y) f(y) \rho_0(y) \mathrm{dvol}_y \tag{3.17}$$

and the corresponding infinitesimal generator $\mathcal{H}[f] := \lim_{\varepsilon\to\infty} \frac{T_{\mu_0,\varepsilon}[f]-f}{\varepsilon}$. Provided the local coordinates, its volume form is $\mathrm{dvol}_x := \sqrt{|g|}\mathrm{d}x^1 \wedge \cdots \wedge \mathrm{d}x^n$.

3.5.3 Operator limits of FSA

Our formulation of FSA provides a principled framework for analyzing attention scores and weights through their spectral properties. To this end, we first investigate the second-order dynamics derived from the infinitesimal generator in the following theorem. We then highlight the advantages of the non-local case ($\alpha \neq 2$) in comparison to the local case ($\alpha = 2$) within a low-dimensional setting. Finally, we explore the fundamental differences between the Laplace-Beltrami operator and its fractional counterpart, shedding light on their distinct mathematical properties.

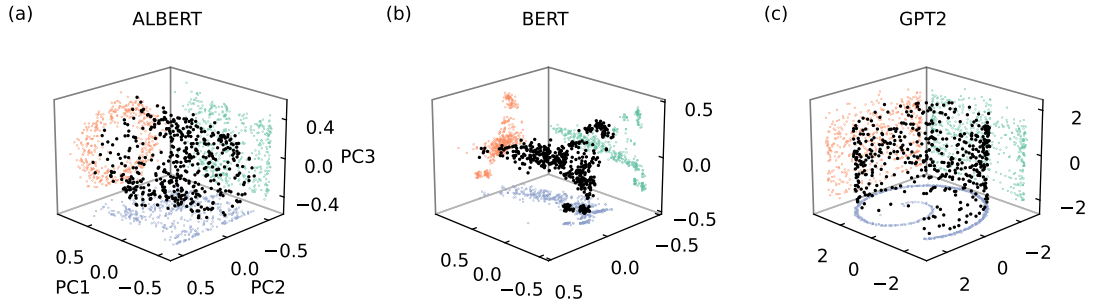

Figure 2: PCA of token embeddings. (a–c) Top 3 principal components of embeddings of a random input sentence (generated by GPT2) represented by scattered black dots for pretrained ALBERT, BERT and GPT2 respectively. The other colored points are the projections onto two principal components, i.e. blue for PC1 vs PC2, green for PC1 and PC3 and orange for PC2 and PC3.

Theorem 1 (Fracformer's PDE)**.** *Let $\mu \in \mathbb{P}(\mathcal{M})$ be a probability measure supported on $\mathcal{M} = \mathbb{R}^d$ or $\mathbb{S}^{d-1}$ with density $\rho_0 \in \mathcal{C}^3(\mathcal{M})$. Under Assumptions 1 and 2, as the bandwidth parameter $\varepsilon \to 0$, the Fracformer operator has convergence*

$$\mathcal{H}[f] = \frac{(-\Delta_{\mathcal{M}})^{\alpha/2}(f\rho_0) - f(-\Delta_{\mathcal{M}})^{\alpha/2}\rho_0}{\rho_0} \tag{3.18}$$

in L^2.

For a uniform initial density, i.e. $\rho_0 \equiv 1$, $\mathcal{H}[f] = (-\Delta_{\mathcal{M}})^{\alpha/2} f$, allowing us to recover the fractional diffusion as in equation 3.7.

We outline the proof of Theorem 1 in Appendix 3.A.7, which is related to the diffusion map algorithm leading up to eigen-decomposition and the spatially continuous counterpart in Section 3.5.2 and Appendix 3.A.6, respectively. In the mathematical formulation of our fractional self-attention (FSA), it is important to account for the underlying geometry and distribution of the token embeddings, which are generally complex and non-uniform, i.e. $\mathrm{d}\mu(x) \neq \mathrm{d}x$. To demonstrate these non-trivial geometrical structures, we examine the embeddings of an identical input sequence in three different, popular, pre-trained Transformer models: ALBERT [114], BERT [115] and GPT2 [116]. In Fig. 2(a,c), ALBERT and GPT2 reveal circular and spiral patterns, respectively. These geometrical structures have been linked to the encoding of positional information within a sequence [117]. In contrast, the clustering of embeddings in BERT has been associated with the representation of semantic information [115] (Fig. 2(b)). Thus, the underlying distribution μ_0 plays an important role.

Now we demonstrate how non-local kernels allow for longer-range interactions under non-uniform densities. As shown in Fig. 2, the embedding density is typically

non-uniform. In a low-dimensional setting, let us construct a distribution bimodal 2D-multivariate Gaussian distribution $f_2(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. We construct $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ that represent two distant clusters in the embedding space in Fig. 1(g),

$$\rho_0(x) = \frac{1}{2} f_2(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + \frac{1}{2} f_2(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}).$$

After applying equation 3.15 in contrast to $\alpha = 2$ (Fig. 1(h)), we can see that the case $\alpha < 2$ encourages more frequent interactions between tokens, especially between the two modes $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ (Fig. 1(i)).

Our self-attention structure based on the stochastic dynamics of Lévy processes can efficiently preserve these structures. Replicating these processes based on the description of self-attention as equation 3.1 requires Assumptions 1 & 2 which can be fulfilled during training.

3.6 Experiments

In this section, we showcase the performance of the Fracformer architecture across various tasks and modalities. The stability parameter α is specified before training. In Section 3.5, we have shown that fractional diffusion occurs exactly under both Assumptions 1 & 2 which are only satisfied for a single attention head (see Methods). For simplicity, we will abbreviate the original Transformer as *DP* due to the dot-product operation being deployed in SA.

3.6.1 Text classification

We consider the IMDb movie review dataset [118] for binary text classification which involves predicting whether a movie review expresses a positive or negative sentiment. Fracformer includes the stability parameter α which controls the jump-intensity of the underlying Lévy process.

Here we present results for models with and without orthogonal projection matrices for the queries and keys. For $W_{Q,K} \notin O(d)$, Fracformer with $\alpha = 1.2$ achieves the highest median evaluation accuracy by the end of training for 20 epochs across 5 trials (Fig. 3(a) blue curve). Although Transformer attains an accuracy of $\sim 82.7\%$ earlier in the learning process, its performance quickly plateaus after epoch 7, whereas Fracformer continues to improve. When $W_{Q,K} \in O(d)$, almost identical effects are observed but the test accuracy discrepancy between Fracformer ($\alpha = 1.2$) and the Transformer further expands. Exact opposite trends occur for the test loss in Fig. 3(c–d). In Table 1, we perform a non-exhaustive parameter sweep over $\alpha = 1, 1.2, \cdots, 1.8$ and find that for both cases of $W_{Q,K} \in O(d)$ and $W_{Q,K} \notin O(d)$, $\alpha = 1$ yields the best performance. For further implementation details, please refer to Methods.

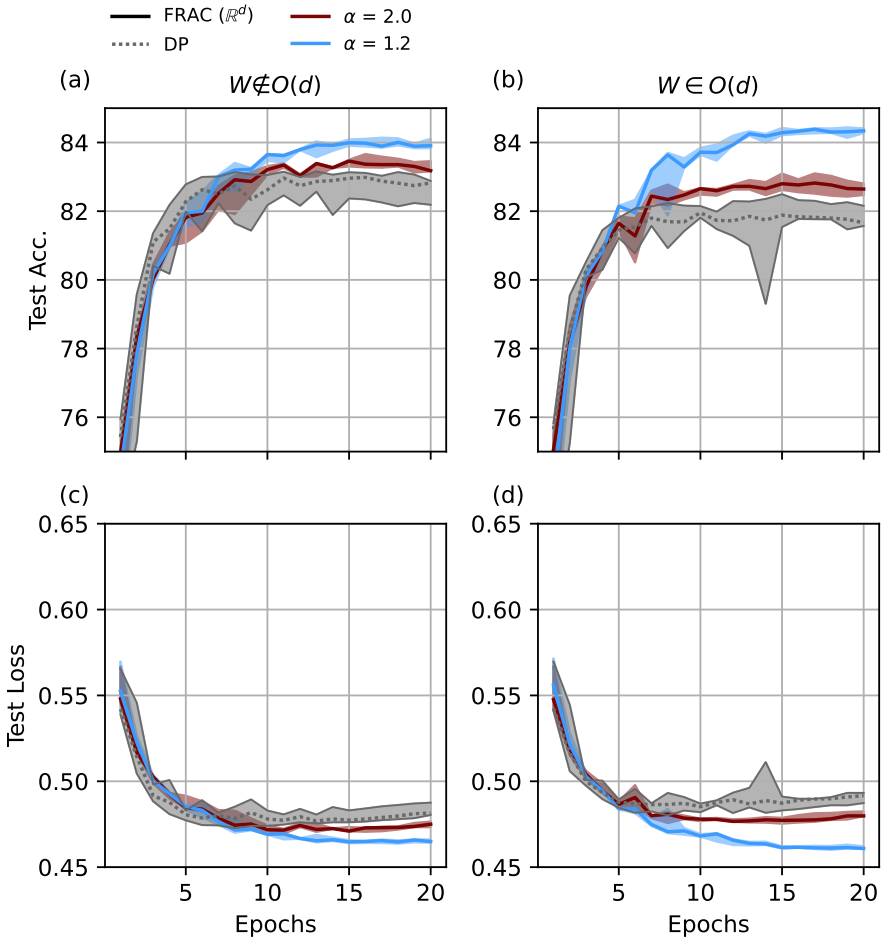

Figure 3: Learning curves for text classification. (a – b) The median test accuracy percentage for Fracformer (FRAC) and Transformer (DP) over 5 trials respectively for $W_{Q,K} \notin O(d)$ and $W_{Q,K} \in O(d)$. The lower and upper parts of the color shade represent the worst and best over the 5 trials at each epoch. The blue and red curves represent the cases of the hyperparameter setting $\alpha = 1.2$ and $\alpha = 2$ for Fracformer. (c – d) Same as in (a – b) respectively but for the test loss.

Projection matrix	Best Acc. ($\alpha < 2$)	α value	Acc. ($\alpha = 2$)	Acc. (DP)
$W_{Q,K} \notin O(d)$	$84.06 \pm 0.26\%$	1	$83.28 \pm 0.21\%$	$82.60 \pm 0.36\%$
$W_{Q,K} \in O(d)$	$84.40 \pm 0.19\%$	1	$82.65 \pm 0.28\%$	$81.76 \pm 0.23\%$

Table 1: Model performance on IMDb dataset with $Q \neq K$. Column 1. Setup of the query and key projection matrices W_Q, W_K. Column 2. Best mean test accuracy across 5 iterations of trianing for the parameter search across $\alpha = 1, 1.2, 1.4, 1.6, 1.8$. Column 3. The α value corresponding to the accuracy in Column 2. Column 4. The mean test accuracy for $\alpha = 2$. Column 5. The mean test accuracy for the Transformer (DP) in Fig. 3. The best accuracy for each row is highlighted in bold.

3.6.2 Image processing

The notion of a sequence can be introduced to spatial structures such as images, by feeding ordered and selected patches of fixed size in images to the attention mechanism. Thus Vision Transformers (ViT) [40] have emerged as a promising alternative for computer vision tasks. We train Fracformers and compare them to the Transformer model class on CIFAR10 [78].

As in Section 3.6.1, effects of relaxing Assumption 1 are also further examined. Throughout training, Fracformer ($\alpha = 1.2$) dominates the performance of its Gaussian counterpart ($\alpha = 2$) and the Transformer model under the condition $W_{Q,K} \notin O(d)$ (Fig. 4(a)). When orthogonality is enforced ($W_{Q,K} \in O(d)$), Fracformer experiences a slight performance degradation which nonetheless outperforms Transformer which has its median accuracy almost identical to the previous case (Fig. 4(b)). Opposite trends are observed for the test loss in Fig. 4(c–d).

3.6.3 Pretrained analysis

Our findings indicate that Fracformer outperforms Transformer in both text and image classification tasks. Additionally, its performance varies across different hyperparameter configurations, particularly with respect to the choice of α. In the following analysis, we further investigate the impact of α, demonstrating how the spatially fractional regime ($\alpha < 2$) contributes to improved results.

Fracformer hyperparameter ablation

In Fig. 3, a notable performance gap between the non-Gaussian and Gaussian case has been observed. Here, we investigate the effects of α on the performance of 1-layer Fracformer. As the embedding layer evolves during training, we instead use a set of pre-trained GloVe (Global Vectors for Word Representation) embeddings

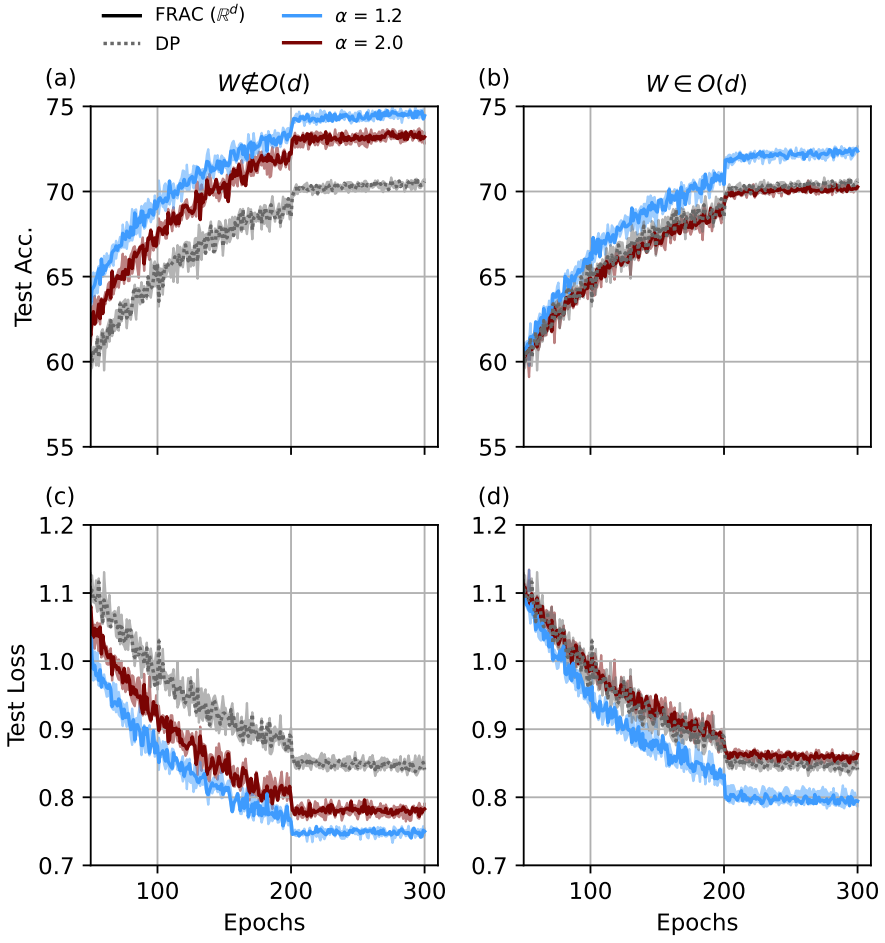

Figure 4: CIFAR10 classification. (a – b) The median test accuracy percentage for 2-layer and 6-attention-head Fracformer and Transformer over 5 trials resp. for $W_Q, W_K \notin O(d)$ and $W_Q, W_K \in O(d)$. The lower and upper parts of the color shade represent the 25% and 75% percentile over the 5 trials at each epoch. The blue and red curves represent the cases of the hyperparameter setting $\alpha = 1.2$ and $\alpha = 2$ for Fracformer. (c – d) Same as in (a – b) resp. but for the test loss. (c–d) Same as in (a - b) resp. but for the test loss.

Dataset	$\mathcal{M}$	WE fixed	Best Acc. ($\alpha < 2$)	α value	Acc. ($\alpha = 2$)
MNIST	$\mathbb{S}^d$	No	$93.87 \pm 0.35\%$	1.6	$93.66 \pm 0.41\%$
IMDB	$\mathbb{R}^d$	Yes	$62.89 \pm 1.0\%$	1.4	$62.74 \pm 1.4\%$

Table 2: Model performance on IMDb dataset with $Q \neq K$. Column 1.The corresponding task for the Fracformer model. Column 2. The manifold $\mathcal{M}$ setup for the model. Column 3. Whether pretrained word/token embeddings (WE) are used during the training process. Column 4. Best mean test accuracy across 5 iterations of training for the parameter search across $\alpha = 1, 1.2, 1.4, 1.6, 1.8$. Column 5. The α value corresponds to the accuracy in Column 4. Column 6. The mean test accuracy for $\alpha = 2$.

[119], this will preserve the token embeddings and allow for a direct contrast later on in our pretrained analysis. We repeat a similar training process as before on the MNIST dataset for vision tasks and IMDb for text classification and summarize the results in Table 2.

FSA as diffusion maps

The trainable parameters of Transformers are often dominated by linear/dense layers instead of the actual self-attention layers. To understand the the impact of α which controls the notion of non-locality of FSA, we again conduct our analysis for a single-layer and attention-head Fracformer. As the embedding layer also evolves during training, we dissect our experiments into two parts: 1. Text classification: we fix the embedding layer to the set of pretrained GloVe embeddings that allows us to control the hidden dimension. Thus for the same sequence, this would retain the same pre-FSA hidden state representation and allow for a direct comparison. 2. Image classification: we keep the embedding layer as a learnable component and examine the effects of the corresponding inductive bias introduced by α.

Through the lens of diffusion interactions on manifolds, we examine the properties of FSA for pretrained Fracformers with ($W_Q = W_K = I$) as in Section 3.5.3, similar to the demonstration in Fig. 3.A.1. We can obtain an eigen-decomposition of the attention weights (equation 3.16)

$$K_{ij}^{\varepsilon} := \sum_{m \geq 1} \lambda_m^{\varepsilon} \psi_m(x_i) \phi_m(x_j) \tag{3.19}$$

where ϕ_m and ψ_m are the left and right eigenvectors of K; λ_m are the corresponding eigenvalues (see Appendix 3.A.8 for more details). The diffusion distance between two points is defined by

$$D_t^2(x_i, x_j) = \sum_y \left(p(t, y \mid x_i) - p(t, y \mid x_j)\right)^2 w(y) \tag{3.20}$$

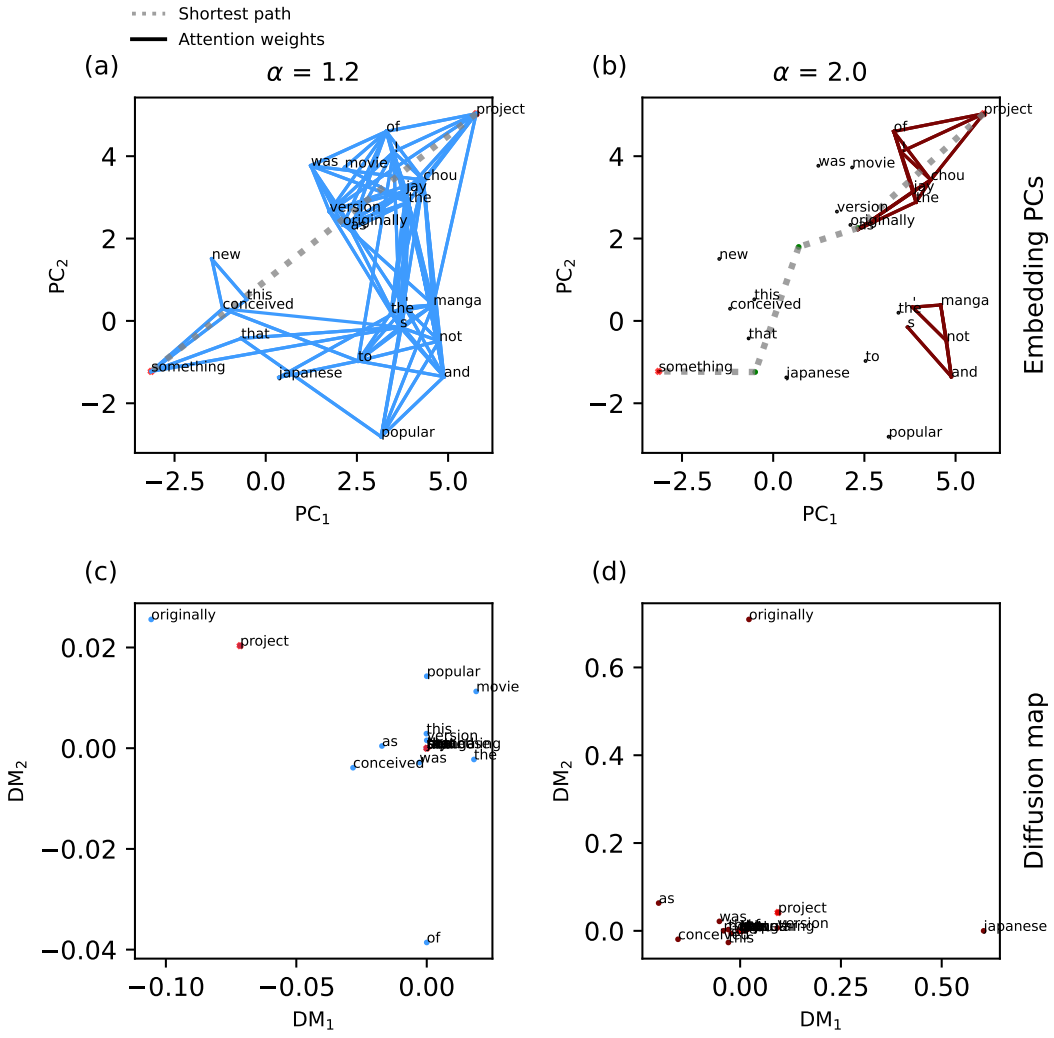

Figure 5: Fractional self-attention and diffusion maps. (a – b) The top 2 principal components of the Fracformer embeddings tokens of a random selected sequence from the IMDb dataset with with length of 512. For $\alpha = 1.2, 2$ respectively, we plot the attention weights between all presented tokens which are strictly greater than 5×10^{-17}. The solid line thickness is scaled linearly with respect to $5/(1 - 5\log(K_{ij}))$, indicating the strength of the corresponding attention weight between the tokens that encode both semantic and positional information. The furthest tokens apart in the embedding space are presented as red crosses and the dotted gray lines trace out the most likely path between them; the tokens on this path are highlighted in green. The likelihood for $\alpha = 1.2$ and $\alpha = 2$ are (c – d) The diffusion map embeddings for the same sequence as in (a,b) with the top 2 components of equation 3.21 for $\alpha = 1.2, 2$ respectively.

with the weight function $w(y) = 1/\phi_0(y)$ accounting for the (empirical) local density of points [47, 48]. The *diffusion map* is the mapping between the original space and the first m eigenvectors

$$\Psi_t(x_i) = \left(\lambda_1^t \psi_1(x_i), \lambda_2^t \psi_2(x_i), \ldots, \lambda_m^t \psi_k(x_i)\right). \tag{3.21}$$

The following seminal result establishes the diffusion distance and the Euclidean distance in the diffusion map space with all $n-1$ eigenvectors

$$D_t^2(x_i, x_j) = |\Psi_t(x_i) - \Psi_t(x_j)|^2 \tag{3.22}$$

Therefore we can obtain a canonical dimensionality reduction of the FSA through equation 3.21, and the diffusion distance is retained to a certain degree depending on the number of eigenvectors chosen and the eigenspectrum.

In Fig. 5(a–b), we randomly select a sequence of maximal length (n = 512) and project the top two principal components of distant tokens onto $\mathbb{R}^2$ (see Methods). We then visualize attention weights between tokens that exceed a certain threshold, with line thickness representing interaction strength. We observe that the heavy-tailed case $\alpha = 1.2$ exhibits more frequent interactions in the embedding space compared to $\alpha = 2$. Additionally, we perform the diffusion map algorithm to the sequence embedding and project the same tokens in (a–b) to the top 2 components in Fig. 5(c–d). These tokens remain in proximity to each other for $\alpha = 1.2$, demonstrating the ability of non-local kernels to capture long-range interactions.

3.7 Discussion

In this study, we adopt a novel approach by incorporating the dynamics and statistical properties inspired by neurobiological systems [30, 31]. Specifically, we integrate a broader class of fractional diffusion processes which appears in the theory of Fractional Neural Sampling (FNS) [31] into the self-attention mechanism, termed fractional self-attention (FSA). We demonstrate that the corresponding architecture Fracformer reveals performance gains from heavy-tailed dynamics ($\alpha < 2$), driven by long-range spatial interactions, analogous to Lévy processes. Through utilizing a time-scale parameter (kernel bandwidth), methods from diffusion maps help facilitate meaningful low-dimensional token representations, beneficial for model distillation and interpretability [120]. Additionally, under simple conditions on the query/key projection matrices, we mathematically derive the corresponding partial differential equation that describes the second-order dynamics of FSA, which involves a non-local fractional Laplacian operator.

The non-locality of the fractional Laplacian should not be conflated with that in non-local neural networks [121]. Transformers exhibit non-locality through

self-attention, which allows tokens to interact globally, particularly during the encoding process. The bi-stochastic self-attention map also contributes to a non-local structure, but, as detailed in Section 3.3, its second-order dynamics corresponds to the standard Laplacian [6]. Therefore, the concept of locality differs for networks and operators. In our approach, we extend self-attention dynamics to non-local operators, which serve as generators for broader classes of stochastic processes known as Lévy processes. The heavy-tailed dynamics of Lévy processes demonstrate desirable properties in self-attention and contribute to long-range interactions in the embedding space.

In this work, we integrate biologically plausible mechanisms based on the probabilistic nature of Lévy processes into artificial neural networks, showcasing bio-inspired AI within the framework of Transformers [103]. We highlight the significance of probabilistic sampling as a key element in both neuroscience and machine learning. By connecting continuous space-time stochastic processes to their discrete counterparts, we incorporate analogous dynamics into self-attention networks. As a direction for future work, the drift term and momentum component from the FNS theory can be additionally incorporated as part of the mechanism which can potentially further enhance sampling efficiency.

3.8 Methods

3.8.1 Model implementations

Orthogonal and semi-orthogonal projection. A token's query/key projection involves a linear transformation $\iota : \mathbb{R}^d \to \mathbb{R}^{d_k}$. When the number of attention heads $H > 1$, strict orthogonality is no longer preserved due to the division of the embedding dimension by H. Instead we have $\iota : \mathbb{R}^d \to \mathbb{R}^{d/H}$ where H is chosen so that $d \mod H \equiv 0$. Enforcing an orthogonal matrix W_Q can be implemented in PyTorch, the projection for each attention head would be $W_Q^{(i)}x \in \mathbb{R}^{d/H}$ where $W_Q^{(i)}$ is the submatrix $[W_Q]_{(i-1)H:iH,:}$ for $i = 1, \cdots, H-1$. All the rows of W_Q are orthogonal, and this does not make $W_Q^{(i)} \in \mathbb{R}^{d/H \times d}$ semi-orthogonal from the left, i.e. $W_Q^{(i)}x$ no longer preserves the Euclidean distance for $x \in \mathbb{R}^d$.

Kernel function. As discussed in Section 3.4.1, the non-local kernel (equation 3.11) explicitly depends on the dimensionality for $\mathcal{M} = \mathbb{R}^d$. For many existing Transformer architectures, the choice of the embedding dimension is typically large [114, 115, 116], e.g. $d = 512, 768, \cdots$. This high dimensionality causes the intersection point with the local kernel to increase with respect to d, leading to a diminishing distinction between the non-local and local kernels (Fig. 1(b–c)). To

address this, we adopt an alternative formulation for the non-local case using

$$\tilde{\Phi}(z) = (1+z)^{1+\alpha}, \quad \alpha \in [1,2). \tag{3.23}$$

Scaling geodesic distances on the spherical manifold. Given $\mathcal{M} = \mathbb{S}^{d-1}$ for $\alpha < 2$, we have $\Phi = (1 + d_g(x_i, x_j))^{-d_{\mathcal{M}}-\beta}$, which has poor asymptotic properties for large d_k (assuming that $W_Q \in \mathbb{R}^{d_k \times d_k}$). By inserting a scale parameter $r = \frac{\pi^{1/(1-d_{\mathcal{M}})}-1}{\pi}$ into

$$\Phi(d_g(x_i, x_j)) = (1 + r d_g(x_i, x_j))^{-d_{\mathcal{M}}-\beta} \tag{3.24}$$

which corresponds to the radius of the $(d_k - 1)$-sphere that the x_i's are embedded on, this particular issue can be mitigated effectively. The intrinsic dimension of the manifold $d_{\mathcal{M}} = d_k + 1$ in this case.

Floating point errors. In the symmetrical case of $Q = K$ and $\mathcal{M} = \mathbb{S}^{d_k-1}$ (normalized embeddings), arithmetic errors are particularly induced by the diagonal of the dot-product matrix where the norm of each embedding could be evaluated to be over 1, thus we mask the dot product with 1 along this diagonal.

Sequence paddings. Due to shorter sequences being padded, masks are typically applied to keys but not queries. Hence for sequences with length shorter than the pre-specified maximum, symmetry of the pairwise distance is not attained, nonetheless this does not impact performance. As only limiting sequence lengths are of interest theoretically, symmetry of pairwise token distance is always assumed under $Q = K$.

3.8.2 Experimental details

Text classification. In Fig. 3, all models trained are of 6 layers and 8 attention heads; we use an initial learning rate of 10^{-4} and decay it by a factor of 10 at epoch 15. The Fracformers trained are trained for $\alpha = 1, 1.2, \cdots, 2$. For $\alpha < 2$, we use the d-independent non-local kernel as in equation 3.23 due to the large embedding dimension $d = 256$ deployed.

Visual processing. In Fig. 4, all models trained are of 2 layers and 6 attention heads; we use an initial learning rate of 10^{-4} and decay it by a factor of 10 at epoch 200. The Fracformers trained are set at $\alpha = 1.2, 2$ and the scaling-constant in equation 3.15 is set to $\kappa = 1$. Additionally, the original non-local kernel function (equation 3.11) is applied.

Single-layer Fracformers. Table 2 presents results for Fracformers configured with a single layer and a single attention head, using an embedding dimension of $d = 16$. The projection matrices $W_Q = W_K = I$ are set to the identity matrix in order to satisfy Assumptions 1 & 2 in the main text. The pretrained semantic embeddings are initialized with token and position embeddings from a set of pretrained GloVe embeddings [119] (dimension 50), which are projected onto the top d principal components using principal component analysis (PCA). The training setup follows the Text Classification methods above, with modifications: training extends to 30 epochs, and the learning rate is set to 5e-4 with a decay factor of 10 applied at epoch 23.

Pretrained analysis. To illustrate long-range interactions between distant tokens in the embedding space, we first arbitrarily select a maximum-length sequence (512) and compute the center of the token embeddings as the mean coordinates of the entire sequence. Next, we remove tokens located within a radius of 6.1 from this center and project the remaining ones onto $\mathbb{R}^2$ using PCA (Fig. 5(a–b)). The diffusion map embeddings in the second column are obtained through the algorithm as discussed in Section 3.6.3 (Fig. 5(c–d)), a diagonalization method is also detailed in Appendix 3.A.8.

Supplementary Materials

3.A Theoretical results

3.A.1 Fractional Laplacian on manifolds

Given a square integrable function $u \in L^2(\mathcal{M})$, the fractional Laplacian is defined spectrally through

$$(-\Delta_{\mathcal{M}})^{\alpha/2} u(x) := \sum_{i=1}^{\infty} \lambda_i^{\alpha/2} \langle u, \phi_i \rangle_{L^2(\mathcal{M})} \phi_i(x) \tag{3.25}$$

for x in the fractional Laplacian domain. Given the compactness of the manifold, the set $\{\phi_i\}_{i=1}^{\infty}$ composes the orthonormal eigenfunctions through $-\Delta_{\mathcal{M}}\phi_i = \lambda_i \phi_i$ with countably many eigenvalues that satisfy $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots$ with $\lim_{i\to\infty} \lambda_i = \infty$.

For general manifolds $\mathcal{M}$, the fractional Laplacian typically does not present a form as in equation 3.8. Additionally, the operator applied to $f(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ as in Fig. 1 takes form [122]:

$$(-\Delta)^{\alpha/2} f(x) = 2^{\alpha} \frac{\Gamma\left(\frac{n+\alpha}{2}\right)}{\Gamma\left(\frac{n}{2}\right)} {}_1F_1\left(\frac{n+\alpha}{2}, \frac{n}{2}, -\frac{|x|^2}{2}\right). \tag{3.26}$$

where ${}_1F_1$ is the hypergeometric function.

3.A.2 Heat kernels

A function $k : [0, \infty) \times \mathcal{M} \times \mathcal{M} \to [0, \infty)$ is a heat kernel if for almost every $x, y \in \mathcal{M}$ and $\forall s, t \geq 0$

1. Positivity: $k_t(x, y) \geq 0$.
2. Total mass inequality: $\int_{y\in\mathcal{M}} k_t(x, y) \mathrm{dvol}_y \leq 1$.
3. Symmetry: $k_t(x, y) = k_t(y, x)$.

4. Semi-group: $k_{s+t}(x,y) = \int_{z\in\mathcal{M}} k_s(x,z)k_t(z,y)\mathrm{dvol}_z$.

5. Approximation of identity:

$$\lim_{t\to 0^+} \left\| \int_{y\in\mathcal{M}} k_t(x,y)f(y)\mathrm{dvol}_y - f(x) \right\|_{L^2(\mathcal{M},g)} = 0,\ f\in L^2(\mathcal{M},g) \tag{3.27}$$

Every heat kernel gives rise to an associated semigroup

$$\mathcal{K}_t f(x) = \int_{y\in\mathcal{M}} k_t(x,y)f(y)\mathrm{dvol}_y. \tag{3.28}$$

A semigroup also gives rise to a quadratic form

$$\xi(f) = \lim_{t\to 0^+} \left\langle \frac{f-\mathcal{K}_t f}{t}, f \right\rangle_{L^2(\mathcal{M},g)}. \tag{3.29}$$

Let us define $d_g : \mathcal{M}\times\mathcal{M}\to\mathbb{R}^+$ as the geodesic metric given g, which is the determinant of the matrix representation of the metric tensor on $\mathcal{M}$. In general, heat kernels can take various forms, but they must satisfy the conditions outlined above. Our focus will be on stochastically complete heat kernels, meaning they are normalized, or equivalently, the corresponding semigroups satisfy $\mathcal{K}_t 1 = 1\ \forall t > 0$ (equation 3.28).

A quadratic form is *regular* if there exists a set $\mathcal{C}$ of continuous functions with compact support ($C_c(\mathcal{M})$) that are also in the domain $\mathcal{D}(\xi)$ of ξ, i.e. $\mathcal{C}\subset C_c(\mathcal{M})\cap\mathcal{D}(\xi)$ such that $\mathcal{C}$ is dense both in $C_c(\mathcal{M})$ and $\mathcal{D}(\xi)$ under appropriate norms [105]. Assuming the following two conditions:

1. All balls defined with metric d_g in $\mathcal{M}$ are relatively compact;

2. The heat kernel $k_t(x,y)$ is a stochastically complete and α-scale invariant heat kernel such that the associated quadratic form is regular;

the *Heat Kernel Dichotomy* holds [46].

Let $r_0 = r_0(\mathcal{M})$ be the minimum radius of the curvature of $\mathcal{M}$. Then the minimum branch separation $s_0 = s_0(\mathcal{M})$ is defined as the largest positive number such that $|\iota(x)-\iota(y)| < s_0$, then $d_g(x,y)\le \pi r_0$ for $\forall x,y\in\mathcal{M}$. For an unknown manifold that satisfies the minimum branch separation, $d_g(x,y)$ can only be estimated from the graph distance

$$d_G(x,y) := \min_P \left(|\iota(x_0)-\iota(x_1)| + \cdots + |\iota(x_{p-1})-\iota(x_p)|\right) \tag{3.30}$$

over all paths P connecting x_0 to x_p where ι is an isometric embedding. This can be realized via a computationally demanding Dijkstra's algorithm [105].

3.A.3 Lévy process

A Lévy α-stable distribution (often termed *stable distribution*) [70], is defined by a characteristic function involving a tuple $(\alpha, \beta, \sigma, \mu)$ containing the stable, skewness, scale and location parameters respectively,

$$\varphi(u; \alpha, \beta, \sigma, \mu) = e^{-|\sigma u|^{\alpha}(1 - i\beta \text{sgn}(u)\varphi(u;\alpha)) + iu\mu} \tag{3.31}$$

where $\text{sgn}(u)$ is the sign of u and

$$\varphi(u; \alpha) = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right) & \alpha \neq 1 \\ -\frac{2}{\pi}\log|u| & \alpha = 1 \end{cases}.$$

For $\alpha < 2$, the random variable associated with the stable distribution has infinite m^{th}-moment for $m \geq \alpha$.

An α-stable Lévy process L_t is a stochastic process with

1. $L_0 = 0$.
2. L_t has independent increments.
3. $(L_t - L_s) \sim S_{\alpha}((t-s)^{1/\alpha})$ for all $0 \leq s \leq t$.
4. L_t is stochastically continuous.
5. L_t is RCLL (right continuous with left limits) almost surely.

If $\beta = \mu = 0$, then the process is $S\alpha S$.

3.A.4 Orthogonal projection matrices

In the following, we present the proof for Proposition 1:

Proof. In $\mathbb{R}^d$, the distance between each query-key pair under Assumption 1 follows

$$\begin{aligned} |q_i - k_j| &= (|q_i|^2 - 2q_i^{\top}k_j + |k_j|^2)^{1/2}, \\ &= (|W_Q x_i|^2 - 2x_i^{\top}W_Q^{\top}W_K x_j + |W_K x_j|^2)^{1/2}, \\ &= (|x_i|^2 - 2x_i^{\top}\tilde{W}_K x_j + |x_j|^2)^{1/2}, \\ &= |x_i - \tilde{W}_K x_j|, \end{aligned}$$

where $\tilde{W}_K := W_Q^{\top}W_K \in O(d)$, allowing us to set without loss of generality $Q = X$ and $K := \tilde{W}_K X$. For $\mathcal{M} = \mathbb{S}^{d-1}$, the geodesic is simply the great circle $d_g(q_i, k_j) = \arccos(q_i^{\top}k_j)$. Following a similar line of reasoning, this result naturally extends to the spherical manifold.

□

3.A.5 FSA on spherical manifold

To differentiate between local and non-local operators, one needs to remove the sampling density or force a uniform grid [105]. Let us consider an example in the low-dimensional setting on $\mathbb{S}^1$ where we create a grid of 500 evenly spaced points. By Weyl's law, the eigenvalues of the Laplace-Beltrami operator $\lambda_j \propto j^2$ for large j as shown by the dashed lines (red curves, Fig. 3.A.1(a)). Similarly, we have the power spectra close to $\alpha/2$ for fractional Laplacians (blue curves, Fig. 3.A.1(a)). Under a uniform grid of the manifold, the Laplacian and fractional Laplacian can be differentiated through the properties of their eigenspectrum.

The von-Mises Fisher distribution supported on $\mathbb{S}^d$ is given by

$$f_d(\mathbf{x}; \boldsymbol{\mu}, \omega) = C_d(\omega) \exp\left(\omega \boldsymbol{\mu}^\top \mathbf{x}\right); \quad C_d(\omega) = \frac{\omega^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\omega)} \tag{3.32}$$

the normalization constant $C_d(\omega)$ contains the modified Bessel function of the first kind at order v: $I_v(\omega)$. The parameters $\boldsymbol{\mu} \in \mathbb{S}^d$ and $\omega \geq 0$ determine the mean and concentration of the distribution respectively.

Similar to the Euclidean space cse, we display longer-range interactions from the non-kernel under non-uniform densities. Consider the von Mises-Fisher distribution (equation 3.32) parametrized by $(\boldsymbol{\mu}, \omega)$ which is the analog of the Gaussian distribution on $\mathbb{S}^d$; $\boldsymbol{\mu}$ and $1/\omega$ represent the location and dispersion respectively. As shown in Fig. 2, the embedding density is typically non-uniform. In a low-dimensional setting, let us construct a distribution bimodal von-Mises distribution that represents two distant clusters in the embedding space in Fig. 3.A.1,

$$\rho_0(x) = \frac{1}{2} f_d(\mathbf{x}; \boldsymbol{\mu}_1, \omega) + \frac{1}{2} f_d(\mathbf{x}; \boldsymbol{\mu}_2, \omega).$$

After applying equation 3.15, we can see that the case $\alpha < 2$ encourages more frequent interactions between tokens, especially between the two modes $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ (Fig. 3.A.1(c)).

3.A.6 Continuous limits of the kernel construction

Here we outline a more general construction of the attention weights K in equation 3.16 encountered in diffusion maps [47]. Construct the diagonal matrix $D_{ii} = \sum_j \tilde{C}_{ij}$ and the new kernel $K^{(a)}$

$$\tilde{C}^{(a)} = D^{-a} \tilde{C} D^{-a}. \tag{3.33}$$

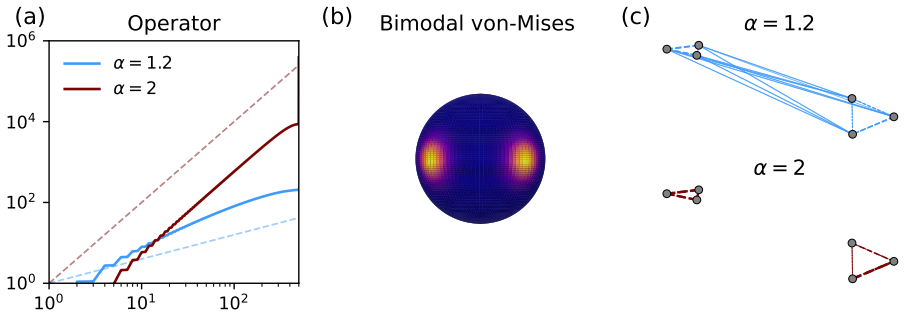

Figure 3.A.1: Token interactions and spectral properties of FSA. (a) Eigenvalues of the fractional Laplacian for $\alpha = 1.2$ (red) and $\alpha = 2$ (blue) computed based on the samples from a (deterministic) uniform grid on $\mathbb{S}^2$. The bandwidth parameter $\varepsilon = 1 \times 10^{-4}$. (b) Probability density function of the bimodal von-Mises Fisher distribution on $\mathbb{S}^2$ centered at $\boldsymbol{\mu}_1 = (0, -1, 0)$, $\boldsymbol{\mu}_2 = (-1, 0, 0)$; $\omega = 50$. (c) A sample size of 6 queries is drawn from the distribution in (b), and mapped onto $\mathbb{R}^2$ through the stereographic projection. Here we assume equivalence between the queries and keys (Assumption 2). The thickness of the dashed lines reflects the strength of attention weights between connected queries and keys, scaled by a constant, based on FSA (equation 3.15) for $\alpha = 1.2$ (top) and $\alpha = 2$ (bottom).

After applying the weighted graph Laplacian normalization based on the parameter a to form the new kernel (equation 3.33), the alternative kernel is simply its row-normalization:

$$D^{(a)} = \sum_j \tilde{C}^{(a)}_{ij}, \tag{3.34}$$

$$K^{(a)} = \left(D^{(a)}\right)^{-1} \tilde{C}^{(a)} \equiv N_R(\tilde{C}^{(a)}). \tag{3.35}$$

The case of equation 3.16 is equivalent to $K^{(0)}$ where we suppress the superscript for notation simplicity. In the limit of an infinite sequence $n \to \infty$, the summation of equation 3.15 can be written in the form of Monte Carlo integration:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} K^{(a)}_{ij} = \int_{\mathcal{M}} k^{(a)}_{\varepsilon}(x, y) q(y) \mathrm{dvol}_y \propto q(x) + \mathcal{O}(\varepsilon) \tag{3.36}$$

where $\mathcal{O}(\cdot)$ is the big-O notation.

We list the continuous counterpart of the kernel construction in Section 3.5.2 and the current section through integral representations

1. Set a rotation-invariant kernel for all pairs of points $x, y \in \mathcal{M}$

$$\tilde{c}_{\varepsilon}(x, y) = \Phi\left(\frac{1}{\sqrt{\varepsilon}} d_g(x, y)\right) \tag{3.37}$$

2. Let

$$q_\varepsilon(x) := \int k_\varepsilon(x,y)q(y)\mathrm{d}y \tag{3.38}$$

and the new kernel

$$\tilde{c}_\varepsilon^{(a)}(x,y) := \frac{\tilde{c}_\varepsilon(x,y)}{q_\varepsilon(x)^a q_\varepsilon(y)^a}. \tag{3.39}$$

Finally, apply the weighted graph Laplacian normalization to this kernel by setting

$$d_\varepsilon^{(a)}(x) := \int_X k_\varepsilon^{(a)}(x,y)q(y)\mathrm{d}y \tag{3.40}$$

and by defining the anisotropic transition kernel

$$k_\varepsilon^{(a)}(x,y) := \frac{\tilde{c}_\varepsilon^{(a)}(x,y)}{d_\varepsilon^{(a)}(x)}. \tag{3.41}$$

The choice of normalization index a affects the operator limit of $k_\varepsilon^{(a)}$ as $\varepsilon \to 0$ (equation 3.33–3.35). For $\alpha = 2$, in the case of $a = 1$[2], the model removes the effects of ρ and thus becomes agnostic of the initial sampling distribution of the tokens/hidden states ρ. In our construction, the underlying manifold of the hidden states is known, making information from the sampling density ρ crucial. Therefore, we set $a = 0$ for FSA in equation 3.16.

3.A.7 Convergence of PDEs

Lemma 1. *The infinitesimal generator from the normalization scheme in the main text yields*

$$\mathcal{H}f = -c\frac{(-\Delta)^{\alpha/2}(f\rho) - f(-\Delta)^{\alpha/2}\rho}{\rho}. \tag{3.42}$$

Proof. First, we emphasize that our setting is slightly different from that of diffusion maps, in our case the manifold $\mathcal{M}$ is known.

Step 1: Similar to the standard case, we start from the diffusion equation [47, 123]

$$u_t = -c(-\Delta)^{\alpha/2}u,\ u(x,0) = f(x) \tag{3.43}$$

for $x \in \mathcal{M}$. The solution at time $t = \varepsilon$ is

$$u(x,\varepsilon) = \frac{1}{Z_{\alpha,d}}\int_{\mathcal{M}} k_\varepsilon(x,x')f(x')\mathrm{dvol}_{x'} \tag{3.44}$$

[2]Slight modifications are required as listed in Appendix 3.A.8

where $Z_{\alpha,d}$ is the normalization constant and k_ε depends on α. For small ε, we can approximate $u(x,\varepsilon)$ through a Taylor expansion

$$u(x,\varepsilon) = u(x,0) + \frac{\partial}{\partial t}u(x,0)\varepsilon + \mathcal{O}(\varepsilon^2) = f(x) - \varepsilon c(-\Delta)^{\alpha/2} + \mathcal{O}(\varepsilon^2). \tag{3.45}$$

Step 2: Fix a normalization scheme:

$$\mathrm{FAttn}[f](x) = \frac{\frac{1}{n}\sum_j \Phi\left(\frac{d_g(x,x_j)}{\varepsilon^{1/\alpha}}\right) f(x_j)}{\frac{1}{n}\sum_j \Phi\left(\frac{d_g(x,x_j)}{\varepsilon^{1/\alpha}}\right)}. \tag{3.46}$$

Step 3: After taking the limit $n \to \infty$, obtain the expression for the Taylor expansion w.r.t ε:

$$\begin{aligned}
&\lim_{n\to\infty} \frac{\frac{1}{n}\sum_j \Phi\left(\frac{d_g(x,x_j)}{\varepsilon^{1/\alpha}}\right) f(x_j)}{\frac{1}{n}\sum_j \Phi\left(\frac{d_g(x,x_j)}{\varepsilon^{1/\alpha}}\right)} = \frac{\int_{\mathcal{M}} k_\varepsilon(x,x') f(x')\,\rho(x')\mathrm{dvol}_{x'}}{\int_{\mathcal{M}} k_\varepsilon(x,x')\rho(x')\mathrm{dvol}_{x'}} \\
=&\frac{(f\rho)(x) - \varepsilon c(-\Delta)^{\alpha/2}(f\rho)(x) + \mathcal{O}(\varepsilon^2)}{\rho(x) - \varepsilon c(-\Delta)^{\alpha/2}\rho(x) + \mathcal{O}(\varepsilon^2)} \\
=&\left[(f\rho)(x) - \varepsilon c(-\Delta)^{\alpha/2}(f\rho)(x) + \mathcal{O}\left(\varepsilon^2\right)\right]\frac{1}{\rho(x)}\left(1 + \varepsilon\frac{(-\Delta)^{\alpha/2}\rho(x)}{\rho(x)} + \mathcal{O}\left(\varepsilon^2\right)\right) \\
=&f(x) - \frac{\varepsilon c(-\Delta)^{\alpha/2}(f\rho)(x)}{\rho(x)} + f(x)\varepsilon c\frac{(-\Delta)^{\alpha/2}\rho(x)}{\rho(x)} \\
=&f(x) - \varepsilon\frac{c}{\rho(x)}\left((-\Delta)^{\alpha/2}(f\rho)(x) - f(x)(-\Delta)^{\alpha/2}(\rho)(x)\right) + \mathcal{O}(\varepsilon^2)
\end{aligned}$$

The third equality was obtained through the expansion of the function $1/x$:

$$\frac{1}{\rho - \varepsilon c(-\Delta)^{\alpha/2}\rho(x) + \mathcal{O}(\varepsilon^2)} = \frac{1}{\rho}\left(1 + \varepsilon\frac{(-\Delta)^{\alpha/2}\rho}{\rho} + \mathcal{O}\left(\varepsilon^2\right)\right).$$

□

In the case of $\mathcal{M} := \mathbb{R}^d$, equation 3.42 further reduces to

$$\begin{aligned}
\mathcal{H}f &:= -c\frac{(-\Delta)^{\alpha/2}(f\rho) - f(-\Delta)^{\alpha/2}\rho}{\rho} \\
&= -c\frac{f(-\Delta)^{\alpha/2}\rho + \rho(-\Delta)^{\alpha/2}f - I_{\alpha/2}(f,\rho) - f(-\Delta)^{\alpha/2}\rho}{\rho} \\
&= -c\frac{\rho(-\Delta)^{\alpha/2}f - I_{\alpha/2}(f,\rho)}{\rho} \\
&= -c(-\Delta)^{\alpha/2}f + c\frac{I_{\alpha/2}(f,\rho)}{\rho}
\end{aligned} \tag{3.47}$$

where $I_{\alpha/2}(f,\rho) := c_{d,\alpha/2}\int_{\mathbb{R}^d} \frac{(f(x)-f(y))(\rho(x)-\rho(y))}{|x-y|^{d+\alpha}}\mathrm{d}y$.

3.A.8 Operator characteristics

Here we state the fractional diffusion map algorithm accredited to [105] for general manifolds. Let us denote the normalized Gaussian kernel as $G(z) := (2\pi\epsilon)^{-d/2} \exp\left(-\frac{z^2}{2\epsilon}\right)$. In order to remove the effects of the sampling density q, we ought to replace the estimation of Step 2 in Section 3.5.2 with

$$\hat{q}_i = \frac{1}{n} \sum_{j=1}^{n} G(d_g(x_i, x_j)) \tag{3.48}$$

and replace the entries of the diagonal matrix D used in equation 3.33 with $\hat{q}_i$ above. Then we proceed to obtain $K^{(1)}$ as usual, the goal is to obtain the eigenvalues and eigenfunctions of the heat kernel in equation 3.35. In low-dimensional cases like in Fig. 3.A.1 where $\mathcal{M} = \mathbb{S}^1$, we simply set ρ to be the uniform grid and $a = 0$.

The diagonalization is conducted on a row-normalized matrix based on equation 3.33–3.35 (with $a = 0$ as in equation 3.16). First note that equation 3.16 is adjoint to the symmetric matrix

$$\hat{K} = D^{-1/2} K D^{-1/2}, \tag{3.49}$$

thus K share the same set of eigenvalues as $\hat{K}$ which are all real (the matrix $(\hat{K} + \hat{K}^\top)/2$ can be made for the case of numerical stability):

$$1 = \eta_0 \geq \eta_1 \geq \cdots \geq \eta_n. \tag{3.50}$$

Additionally, the eigenvectors of $\hat{K}$ form an orthonormal basis $\{\boldsymbol{v}_j\}$. The left and right eigenvectors of P, denoted ϕ_j and ψ_j are related to $\{\boldsymbol{v}_j\}$ through

$$\phi_j = D^{1/2} \boldsymbol{v}_j, \quad \psi_j = D^{-1/2} \boldsymbol{v}_j \tag{3.51}$$

and they are also bi-orthonormal, i.e. $\langle \phi_i, \psi_j \rangle = \delta_{i,j}$. Set $t = \varepsilon^{\alpha/2}$, then the fractional Laplacian eigenvalues $\lambda_i = -t^{-1} \log \eta_i$ and the eigenfunctions ψ satisfy $K\psi_i = e^{t\lambda_i} \psi_i$ for the heat kernel as in equation 3.35.

Chapter 4

Anomalous diffusion dynamics of learning in deep neural networks

Abstract

Learning in deep neural networks (DNNs) is implemented through minimizing a highly non-convex loss function, typically by a stochastic gradient descent (SGD) method. This learning process can effectively find good wide minima without being trapped in poor local ones. We present a novel account of how such effective deep learning emerges through the interactions of the SGD and the geometrical structure of the loss landscape. We divide our findings into two parts, providing an empirical examination on learning dynamics and theoretical discussion for a new insights for modeling this process. In the first part, we find that the SGD exhibits rich, complex dynamics when navigating through the loss landscape; initially, the SGD exhibits anomalous superdiffusion, which attenuates gradually and changes to subdiffusion at long times when approaching a solution. Such learning dynamics happen ubiquitously in different DNNs types such as ResNet and VGG-like networks and are insensitive to batch size and learning rate. The anomalous superdiffusion process during the initial learning phase indicates that the motion of SGD along the loss landscape possesses intermittent, big jumps; this non-equilibrium property enables the SGD to escape from sharp local minima. For the second part which is ongoing work, we propose a framework that models the evolution of weights as a matrix-valued stochastic process during SGD, unlike traditional formulations that flatten all parameters, disregarding the inherent matrix structure of the connectivities. This approach enables the analysis of the eigenvalues and eigenvectors of the connectivity matrix, offering deeper insights into how features are extracted and learned throughout the training process.

4.1 Introduction

Deep neural networks (DNNs) have achieved great success in many application areas such as speech processing, visual object recognition, drug discoveries and material identification [63, 124]. A crucial ingredient for the powerful DNNs is a relatively simple stochastic optimization procedure, in particular, an algorithm based on stochastic gradient descent (SGD). Despite its importance, the fundamental problem of why SGD is effective in finding a solution with good generalization properties [53, 54] in a high-dimensional nonconvex loss function landscape remains unclear. Here, we adapt methods from non-equilibrium, complex systems to investigate the anomalous diffusion of SGD and explain how these dynamics arise from interactions of SGD and the loss landscape.

Related Work. There is a long line of literature that focuses on the learning dynamics of deep neural networks trained with SGD. One line of research has indicated that the structure of the loss landscape is essential for understanding deep learning dynamics. For instance, loss functions have been studied using random matrix theory by drawing comparisons between DNNs and spin glass models [125] along with algebraic geometry methods [55]. Through these methods, it has been shown that the number of local minima decreases with depth asymptotically and becomes more clustered in the parameter space. Additionally, visualization-based methods have been investigated; examination on the principal components of SGD trajectory has indicated that the inclusion of skip connections expedites training by smoothing out the loss landscape [126]. Similarly, dimension reduction of the loss landscape by interpolating between different initialization weights suggests that SGD encounters saddle points and selects certain descent directions to settle on final solutions [127]. These results rely on empirical observations and dimension reduction techniques that overlook the information from lower-ranked principal components. It has also been revealed that saddle points are the main obstacles for the training process and a second order optimization method which escapes saddle points was proposed [128], nonetheless first order methods are still preferred due to efficiency. In another line of research, SGD has been examined via models of stochastic Langevin dynamics with an assumption that gradient noise is Gaussian [129, 130]; in these studies, SGD has been assumed to be driven by Brownian motion. However, it has been increasingly demonstrated that the SGD walker exhibits highly anisotropic, Non-Gaussian dynamics in DNNs [131, 132, 133]. Hence, some researchers have tackled the problem by modeling the SGD as a process exhibiting heavy-tailed behaviors. Particularly, a heavy-tailed behavior in the stochastic gradient noise has been reported [34] and also confirmed in follow-up studies [134]. They further proposed modeling the SGD as a single stochastic differential equation driven by a Lévy alpha-stable process, invoking an existing metastability theory [135] to justify why these dynamics would prefer wide minima

[34]. However, the distribution of gradient noise have been found to be Gaussian in the early phases, and changes throughout the training process [136]. Modelling the gradient noise as a single type of noise is thus inappropriate.

Moreover, in relation to the interactions of SGD and the loss landscape, it has been demonstrated that the anisotropic SGD noise strength inversely depends on the landscape flatness in each principal component direction [85]. In another recent study, [137] introduced a new approach for investigating the generalization properties of dee p neural networks by utilizing results from heavy-tailed random matrix theory. They empirically present a linear relationship between the power law norm and generalization accuracy for pretrained networks due to the heavy-tailed behavior of the singular spectrum of the weight matrices, which is an indication that the weight matrices themselves exhibit heavy tails as well [138]. There have also been advances in model-based segmentation methods [139]. Despite the progress made by these studies, a deep understanding of how the interaction of SGD with the structure of the loss function gives rise to complex learning dynamics, and whether and how such dynamics enable SGD to find wide minima is still lacking.

Main Contributions. In this study, we first reveal the non-equilibrium, anomalous diffusive nature of SGD in deep learning; these include superdiffusion during the initial learning phase, which changes to subdiffusion at long times when approaching a solution at the final stage. We further reveal that, during training, the SGD walker moves from rougher (fractal-like) regions to flatter regions of the loss landscape. These properties are quite general across different neural network types and optimization hyperparameter settings. We use the relation between the mean squared loss of the landscape and the end-to-end distance along the SGD trajectory [140] to quantify the roughness of the loss landscape, which reveals how fractal-like it is. The fractal-like regions of the loss landscape possess varying slopes (Fig. 1) and the corresponding loss gradients along the optimization trajectory display large fluctuations with heavy-tailed distributions, thus resulting in superdiffusion which consists of small movements that are intermittently interrupted by big jumps; this property enables the SGD walker to effectively explore the loss landscape. Finally, we develop a phenomenological model to reveal the mechanistic relationship between the fractal-like loss landscape and anomalous diffusive learning process. We find that such anomalous diffusion dynamics result from the fractal-like structure of loss landscape and further design an experiment to draw comparisons between SGD dynamics on low-dimension landscapes with different fractal dimensions (roughness) which illustrates the computational advantage of superdiffusion. A subdiffusive process, on the other hand, consolidates the residence of the SGD in the flatter areas with good solutions. By uniquely interrelating the structure

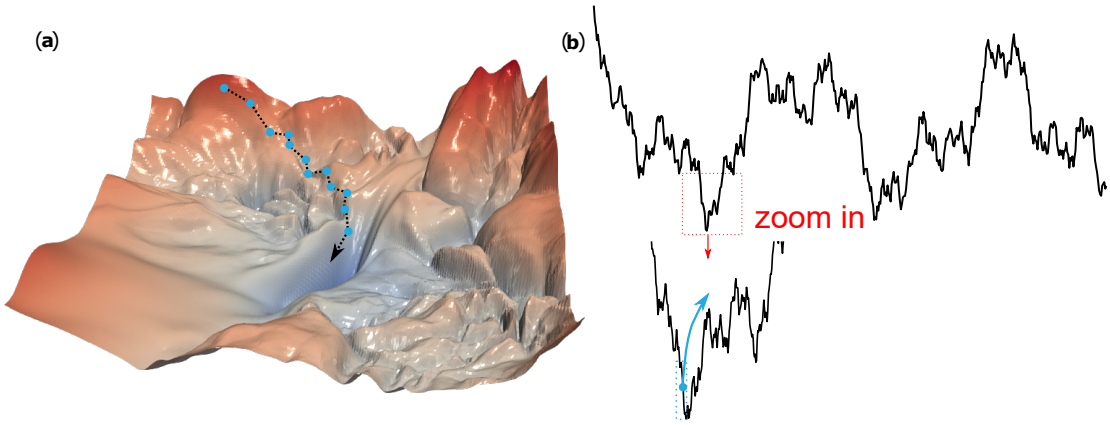

Figure 1: Schematic diagram of non-convex loss landscape with a fractal-like structure. (a) The log-value of loss landscape projected to 2D shows complex structures. The training process based on SGD can be regarded as a SGD optimizer moving on the loss landscape. (b) The fine structure of non-convex loss landscape with a fractal structure shows self-affine and hierarchical properties.

of the loss landscape and diffusion nature of SGD, our results thus offer a novel perspective to understand how deep learning networks work.

4.2 DNNs setup and characterizations of anomalous diffusion

DNN setup. We consider three classes of neural networks: 1) ResNet-14/20/56/110 [17], where each type is labeled with the total number of layers it has. 2) "VGG-like" networks that do not contain shortcut/skip connections. We produce these networks simply by removing the shortcut connections from ResNets, termed ResNet-14/20/56/110-noshort. 3) Vision Transformer (ViT) [141], where an input image is split to a patches size of 2 and is mapped to 128 dimensions. 5-head attention is used. The depth is 5, and the dropout rate is 0.1. All models are trained by vanilla SGD on multiple datasets including MNIST and CIFAR-10; two types of loss functions (i.e., cross-entropy, and Kullback Leibler divergence losses) are applied. The training processes each run for 500 epochs. All networks are initialized in the standard procedures of the PyTorch library (version 1.3.0). The overhead cost for training and analyzing each model is approximately 20 hours on a single V100 GPU. Source code is available at https://github.com/ifgovh/Anomalous-diffusion-dynamics-of-SGD.git.

4.2.1 Understanding SGD via anomalous diffusion

Given that the full-batch gradient descent is computationally expensive, in the SGD algorithm, the weight parameters $\mathbf{\Theta} = (\Theta_1, \Theta_2, \cdots, \Theta_d)$ are estimated by minimizing the minibatch loss function $\nabla\tilde{L}(\mathbf{\Theta}) : \mathbb{R}^d \to \mathbb{R}$, according to

$$\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t - \eta\nabla\tilde{L}_t\left(\mathbf{\Theta}_t\right), \tag{4.1}$$

where $\mathbf{\Theta}_t$ denotes the d-dimension weight vector $(\Theta_1, \Theta_2, \cdots, \Theta_d)$ at time t, and η is the learning rate. In the scenario of a feedfoward network, the minibatch gradient is evaluated via backpropagation with complexity $O(n\prod_{i=1}^{L} N_{i-1}N_i)$ where n is the number of SGD iterations in each epoch and N_i for $i = 0, 1, \cdots, L$ is the dimension of the layer of a network with $L+1$ layers including the input and output. The partial absence of the dataset generates gradient noise $U_t \triangleq \nabla\tilde{L}_t\left(\mathbf{\Theta}_t\right) - \nabla L_t\left(\mathbf{\Theta}_t\right)$ and the updating rule can be rewritten as:

$$\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t - \eta\nabla L_t\left(\mathbf{w}_t\right) + U_t, \tag{4.2}$$

Hence the SGD training process is also a random diffusive process, where $\mathbf{w}$ can be geometrically interpreted as coordinates of the SGD optimizer in the loss landscape L (Fig. 1). Recently, similar methods have also been developed for large-scale graph conventional networks (GCNs), allowing them to be trained in a minibatch fashion [142].

Anomalous diffusion. The loss function of DNN exhibits complex structures, as demonstrated by the projected 2D loss landscape of ResNet-56-noshort [126] in Fig. 1, analogous to complex energy landscape in physical systems [143, 144, 145, 146]. In these physical systems, energy landscapes possess fractal-like structures and anomalous diffusion motions of particles stem directly from such kinds of structures. In this study, we apply similar methods used in these systems to quantify the diffusion process of the SGD optimizer. Particularly, the time-averaged mean squared displacement (MSD) [147, 148] is used to characterize the dynamics of SGD moving through the loss landscape, which is defined by

$$\Delta r^2(t_w, \tau) = \frac{1}{T}\sum_{t=t_w}^{t_w+T}\sum_{i=1}^{d}\left(\Theta_i\left(t+\tau\right) - \Theta_i\left(t\right)\right)^2, \tag{4.3}$$

where τ is the lag time, T is the length of the time interval $[t_w, t_w + T]$, and $\Theta_i(t)$ is the value of the i^{th}-coordinate weight at time t. t_w is the time lapse after the start of the training process (i.e., waiting time). The time variables t, t_w, and T are in units of iteration, and the unit time step corresponds to a single update step of the weights.

Diffusion exponent. We characterize the diffusion dynamics based on time-averaged MSD. Although ensemble-averaged MSD is the most appropriate in theory

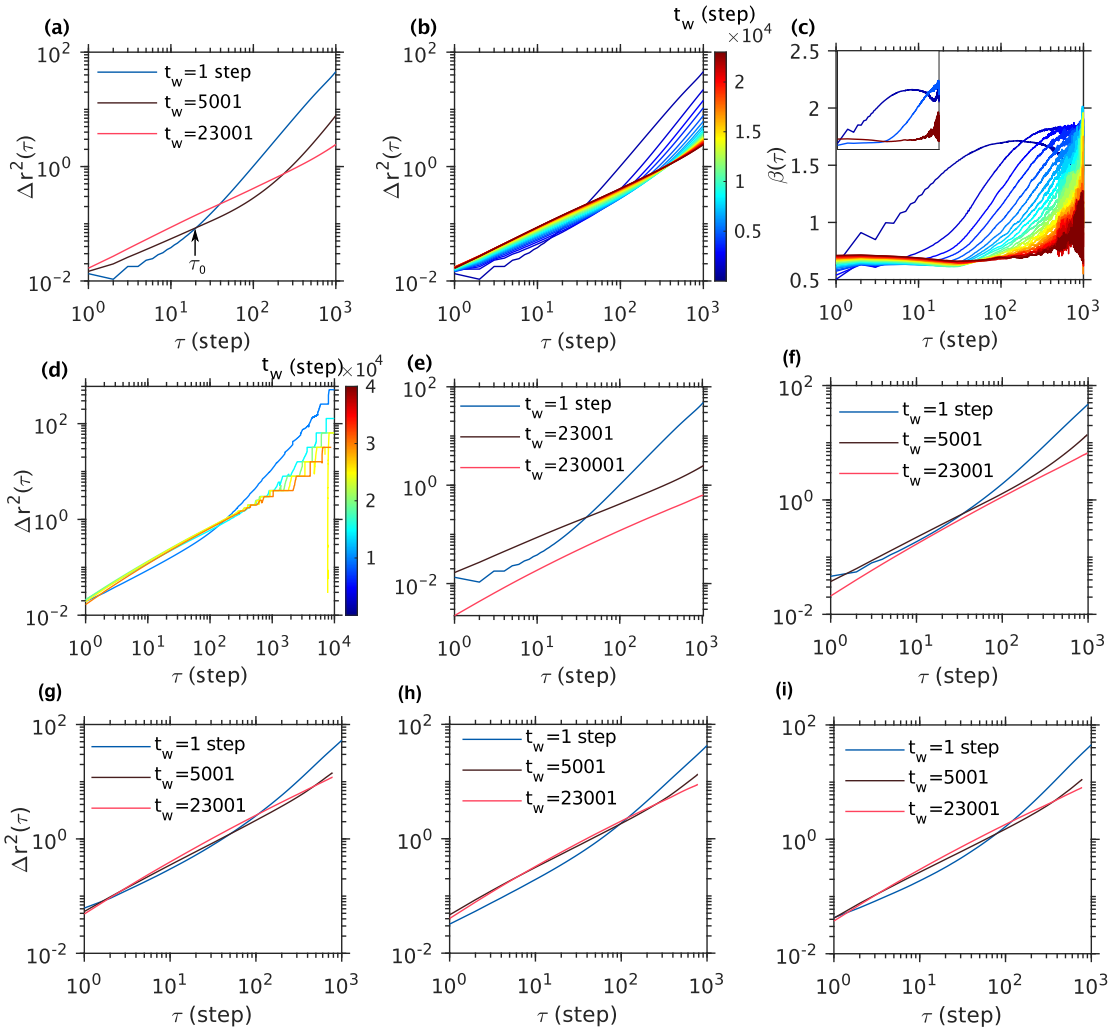

Figure 2: SGD dynamics are anomalous diffusion. (a) MSD $\Delta r^2(\tau)$ of SGD as a function of lag time τ in interval $[t_w, t_w + T]$, $T = 1000$ for ResNet-14 (batch size of 1024, 500 epochs). (b) Same as in (a) but the starting time of the interval t_w gradually increases from 1 to 24000 steps, covering the whole training process. (c) Logarithmic derivative $\beta(\tau)$ of the MSD shown in (b). Inset: Logarithmic derivative $\beta(\tau)$ of the MSD shown in (a). The color scheme is the same as in (b). (d) Same as in (b) but for $T = 10000$. (e) Same as in (a) but for training ResNet-14 5000 epoch. (f-i) Same as in (a) but for ResNet-110 (batch size: 512), ResNet-56 (batch size: 128), ResNet-20-noshort (batch size: 128), and ResNet-20 (batch size: 128), respectively.

for limiting dynamics [149], averaging ensembled trajectories in a high-dimensional system is extremely expensive and hence impossible in practice. Additionally, the ensembled-average MSD does not properly differentiate the type of diffusion dynamics over different time intervals. Therefore, time-averaged MSD is a common tool used to quantify anomalous diffusive processes [147, 148]. The no-averaged MSD (Eq. 4.4) has been used to demonstrate how much the configuration of DNNs and spin glass models at time $t_w + \tau$ decorrelates from the one at time t_w [132].

$$\Delta(t_w, t_w + \tau) = \frac{1}{d} \sum_{i=1}^{d} \left(\Theta_i\left(t + \tau\right) - \Theta_i\left(t\right)\right)^2 . \tag{4.4}$$

We have computed the no-averaged MSD and discovered that no-averaged MSD curves are too noisy to characterize the diffusion process.

Brownian motion is identified by $\Delta r^2(\tau) \propto \tau^{\alpha}$, for large T with the diffusion exponent $\alpha = 1$; its MSD is a linear function of lag time τ. When $\alpha \neq 1$, the corresponding diffusion process has a nonlinear relationship with respect to τ and is defined as anomalous diffusion [150]. If $1 < \alpha < 2$, the process is superdiffusive; superdiffusion consists of small movements that are intermittently interrupted by long-range jumps. This process has been widely observed in complex physical and biological systems [149]; the mixture of small movements and big jumps in this process is essential for optimally transporting energy in turbulent fluid [151], and for animals to optimally search for spatially distributed food [152]. If $0 < \alpha < 1$, the optimizer is subdiffusive, indicating that it moves slower on average than a normal diffusion process.

4.3 Results

4.3.1 Anomalous diffusive learning process

Characterization of anomalous diffusion. We first illustrate that anomalous diffusion processes characterize SGD. As the findings of different DNN settings are similar, we thus demonstrate all results in ResNet-14 with a batch size of 1024, trained on CIFAR-10 with the cross-entropy loss function and a learning rate of 0.1, unless otherwise stated.

The MSD of each interval $[t_w, t_w + T]$ is calculated according to Eq. 4.3 ($T = 1000$) for a given t_w. To demonstrate how the diffusion dynamics of SGD optimizer change during the training process, we change t_w systematically. As shown in Fig. 2(a), there are distinct regimes of the SGD characterized by the diffusion exponent α. For the first regime $t_w < t_0$, $t_0 = 21000$ (blue curve, Fig. 2(a)), MSD curves have two segments on the scale $\tau \in [1, T]$ with the smooth transitions around τ_0 (τ_0 is

labeled in Fig. 2(a)). When the lag time $\tau > \tau_0$, the MSD curves can be fitted to $\Delta r^2(\tau) \propto \tau^\alpha$ with $\alpha > 1$, indicating that the SGD optimizer exhibits superdiffusive dynamics. However, when $\tau < \tau_0$, $\alpha < 1$, i.e. the SGD dynamics are subdiffusive (Fig. 2(a)). The diffusion exponent α is calculated via the least-squared fitting method. We determine τ_0 by fitting $\Delta r^2(t_w, \tau) \propto \tau^\alpha$ for $\tau \in [\tau', T]$, and denote $\tau' \in [0, T)$ with the smallest value such that the root-mean-square deviation RMSE < 0.03 as τ_0.

During the initial phase of the training process, the interval of the superdiffusion is much longer than that of the subdiffusion. Nevertheless, as t_w increases, the superdiffusion gradually attenuates, as demonstrated by the decrease of the diffusion exponent α and the increase of τ_0 (brown curve in Fig. 2(a); all curves are shown in Fig. 2(b)). In the second regime $t_w > t_0$, the diffusion exponent $\alpha = 0.78$, i.e., the subdiffusion process eventually becomes dominant, as shown by the red curve in Fig. 2(a). To identify the change from the first regime to the second one, we estimate t_0 by fitting the MSD curves ($[t_w, t_w + T]$). We shift t_w from 1 to 23001, and $[t_0, t_0 + T]$ is the first curve with the goodness of fit reaching RMSE < 0.03 (0.03 is the standard deviation of all RMSE values). These phenomena can be summarized as below:

$$\Delta r^2(t_w, \tau) \propto \begin{cases} \tau^{\alpha_1} & \text{if } t_w < t_0, \tau < \tau_0 \\ \tau^{\alpha_2} & \text{if } t_w < t_0, \tau \geq \tau_0 \\ \tau^{\alpha_3} & \text{if } t_w \geq t_0 \end{cases} \tag{4.5}$$

where $\alpha_1, \alpha_3 \in (0, 1)$ and $\alpha_2 > 1$. Such processes can be further quantitatively characterized by the logarithmic derivative of the MSD, β [153, 154],

$$\beta(t_w, \tau) = \frac{\ln\, \Delta r^2(t_w, \tau + \Delta\tau) - \ln\, \Delta r^2(t_w, \tau)}{\ln(\tau + \Delta\tau) - \ln\, \tau}, \tag{4.6}$$

where ($\Delta\tau = 20$, Fig. 2(c)); in the first regime, β quickly increases from a value smaller than 1 to a value larger than 1, but in the second regime, β is larger than 1 only when $\tau > 532$.

The time-inhomogeneous anomalous diffusion dynamics are not sensitive to the interval T. For other values such as $T = 5000$, $T = 10000$, the SGD process exhibits similar dynamics with a change from a superdiffusion-dominated regime to a subdiffusion one. Figure 2(d) shows an example of $T = 10000$; supperdiffusion quickly emerges at τ_0 (blue curve) and gradually transitions back to a subdiffusion process (red curve). Note that we show the results in 500 epochs (24000 steps) because the dynamics after 500 epochs remain the same. The same model is also trained for 5000 epochs, the corresponding MSD curves for $t_w > 24000$ still demonstrate subdiffusion (Fig. 2(e)).

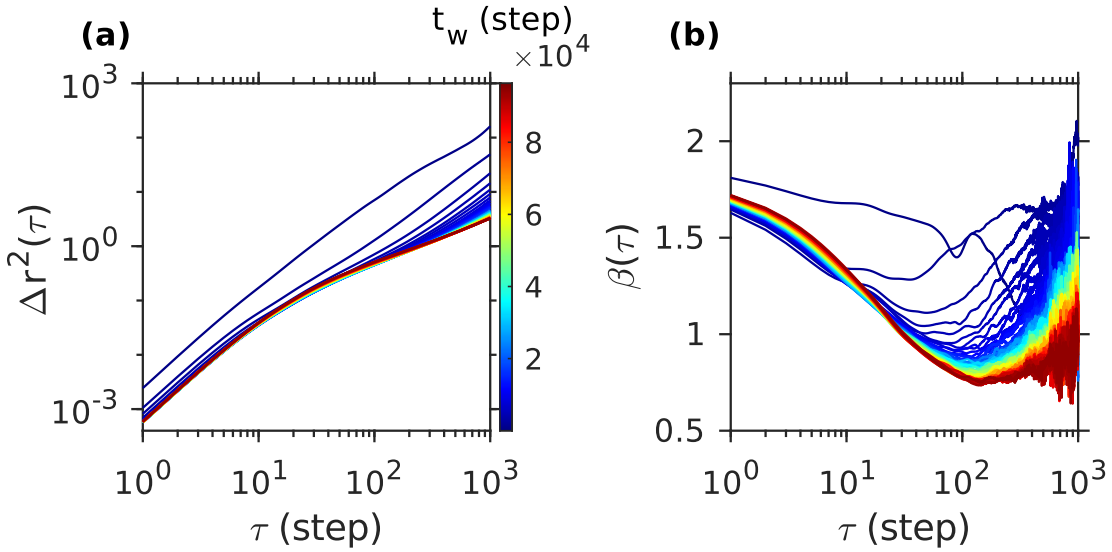

Figure 3: Training dynamics of ViT with Adam are anomalous diffusion. (a) MSD $\Delta r^2(\tau)$ of SGD as a function of lag time τ in interval $[t_w, t_w + T]$, $T = 1000$ for ViT (batch size of 256, learning rate of 0.0001, 500 epochs). The starting time of the interval t_w gradually increases, covering the whole training process. (b) Logarithmic derivative $\beta(\tau)$ of the MSD shown in (a). The color scheme is the same as in (b).

In addition, the time-inhomogeneous anomalous dynamics are not unique to DNN models. Figures 2(f-i) illustrate results for several models and they also demonstrate similar dynamics, including ResNet-110 (batch size: 512) as well as the remaining ResNet-56, ResNet-20-noshort and ResNet-20 all with batch size 128. These models are trained with a learning rate of 0.1. We also train a Vision Transformer (ViT) with an adaptive moment estimation (Adam) algorithm and find that the training dynamics transfer from superdiffusion to subdiffusion (Fig. 3). These results thus indicate that SGD generally possess an initial superdiffusion-dominated phase, which gradually evolves to subdiffusion.

4.3.2 Effect of DNN Architectures and Training Hyperparameters

Effect of DNN architecture. Anomalous diffusion provides a way to identify contributions of network structures and different hyperparameters to the training process. To demonstrate this, we train two types of DNNs, i.e., ResNet and VGG-like networks. VGG-like networks are produced simply by removing shortcut connections from ResNets. As shown in Figs. 4(a-b) and (d-b), shortcut connections

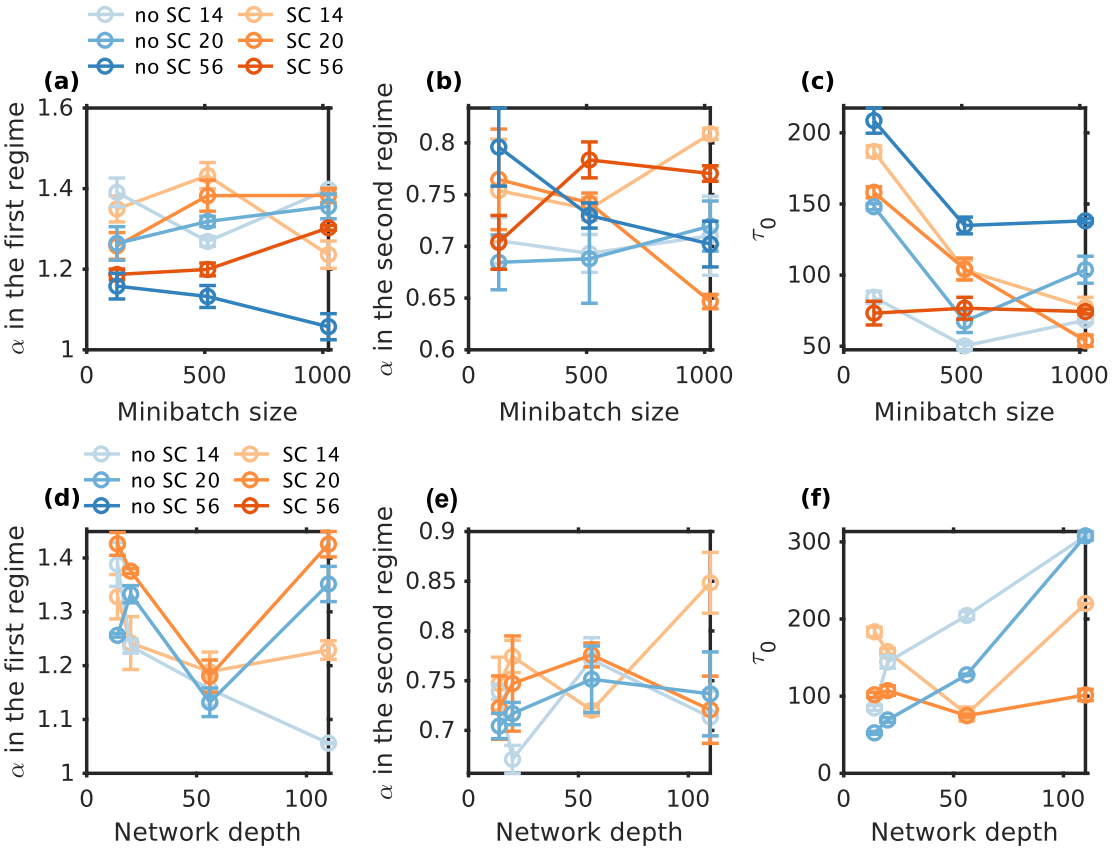

Figure 4: The effects of the depth, batch size, and shortcut connections of DNNs on the anomalous diffusion of SGD. The orange and blue colors represent the network with/without shortcut connections (SC) respectively. The digits in legends of the first reow (a-c) represent network depth; for example, no SC 14 denotes ResNet-14-noshort. Those in the second row (d-f) represent minibatch size; for example, no SC 128 denotes ResNet training using the minibatch size of 128. (a) The diffusion exponents α on larger lag times ($\tau > \tau_0$) when $t_w = 1$ as a function of minibatch size. (b) Similar to (a) but for α in the second regime (when $t_w = t_0$). (c) The crossover τ_0 as a function of minibatch size. Here, τ_0 is the lag time when the MSD curve transitions from subdiffusion to superdiffusion when $t_w = 1$. (d-f) Similar to (a-c) but for varying network depth. The error bars represent the standard deviation calculated over 10 trials.

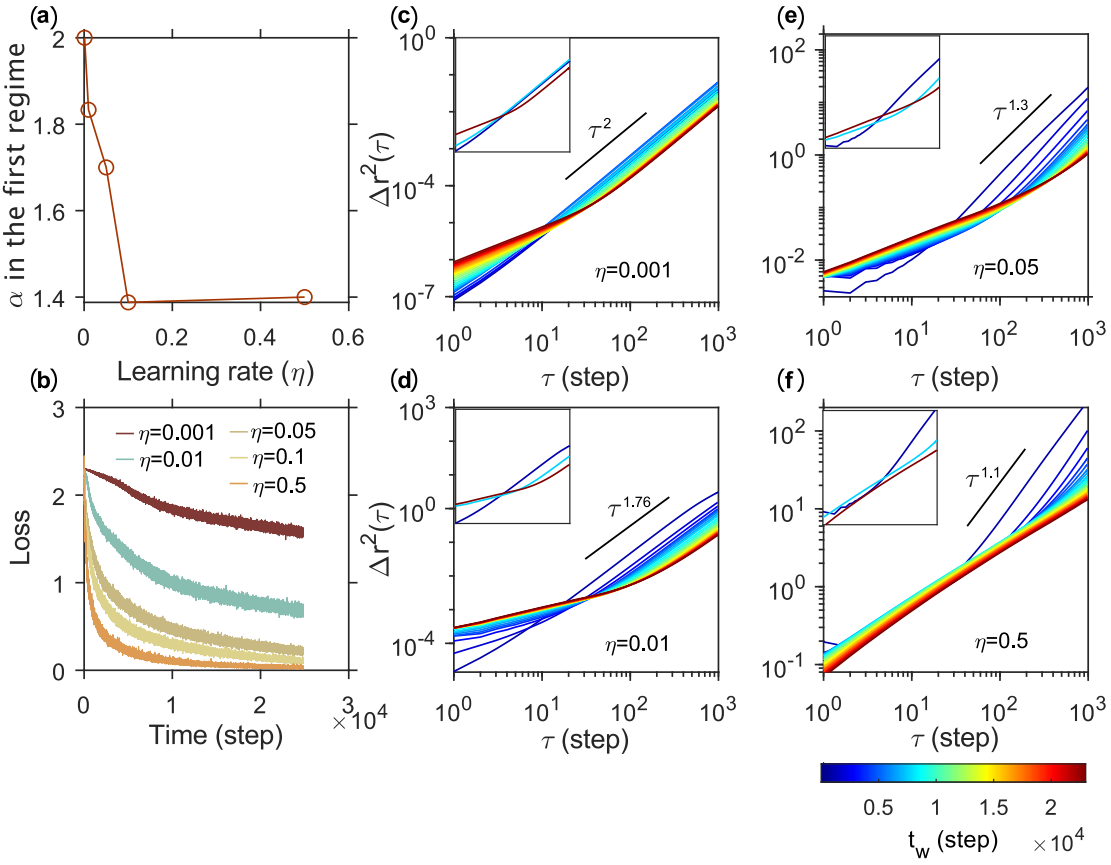

Figure 5: The effect of learning rate on the anomalous diffusion of SGD. All results are from ResNet-14 with a batch size of 1024 on CIFAR10 dataset. (a) The diffusion exponent α on larger lag times ($\tau > \tau_0$) when $t_w = 1$ as a function of learning rate η. (b) The loss value as a function of time. (c-f) The MSD of SGD for the learning rates of 0.001, 0.01, 0.05, and 0.5, respectively. Jet colormap represents the starting time point (t_w) as in Fig. 2(b). One curve represents MSD in 1000 steps. Inset: The MSD of SGD when $t_w = 1, 7001, 23001$. All black lines are eye guides.

do not change the diffusion exponent α significantly in both regimes, with α being greater than 1. However, they do affect the scale range of superdiffusion characterized by τ_0 when $t_w = 1$. τ_0 indicates the crossover of MSD in the first 1000 steps. Specifically, with shortcut connections, τ_0 is smaller than those without shortcut connections, indicating that the scale of superdiffusion is elongated (Fig. 4(c) and Fig. 4(f)). The superdiffusion dynamics enable the SGD walker to explore larger areas of loss landscape in a fixed time than normal diffusion and subdiffusion processes. From this perceptive, DNNs with shortcut connections can facilitate training, which is consistent with previous studies [17, 126].

Effect of minibatch. The anomalous diffusion dynamics are not very sensitive to minibatch sizes. As shown in Fig. 4(a), the diffusion exponent α in the first regime does not significantly vary ($\pm 15\%$) with respect to the change of minibatch size from 128 to 1024 in all networks, although τ_0 decreases in ResNet-14/20 with the increase of the minibatch size (Fig. 4(c)).

Effect of network depth. Furthermore, as shown in Fig. 4(d), α changes nonlinearly with respect to the network depth. However, the network depth reduces the scale range of superdiffusion, characterized by τ_0 (Fig. 4(f)); this result suggests that the deeper DNNs are more difficult to be trained [155] due to the shorter scale of superdiffusion.

Effect of learning rate. We next study the effect of the learning rate η on the anomalous diffusion processes. By varying learning rates from 0.001 to 0.5, we find that it only influences the emerging sequence of the superdiffusion (Fig. 5(a)). As shown in Fig. 5(b), DNN training with small learning rates is much slower than that with large learning rates. Thus, a small or large learning rate can only slow down or speed up the training procedure, but does not change the fundamental occurrence of anomalous diffusion. When the learning rate is small ($\eta < 0.05$), during 500 epochs, the training processes remain in the first regime; there is no pure subdiffusion in Fig. 5(c) and Fig. 5(d). For example, when $\eta = 0.001$, the MSD curves for $t_w < 5000$ have only one segment instead of two and a diffusion exponent of $\alpha = 2$. This is because small learning rates delay the training process. However, DNNs trained with larger learning rates grant superdiffusion in the first regime and pure subdiffusion dynamics in the second regime, as shown in Fig. 5(e) and Fig. 5(f).

These ablation analyses by varying the depth, batch size, learning rate and shortcut connections indicate that the anomalous diffusive processes of SGD are universal in various hyperparameter settings.

4.3.3 Heavy-tailed gradients

Lévy α-stable distribution. To gain further insights into the physical origin of the anomalous diffusion, we next demonstrate the statistical property of minibatch

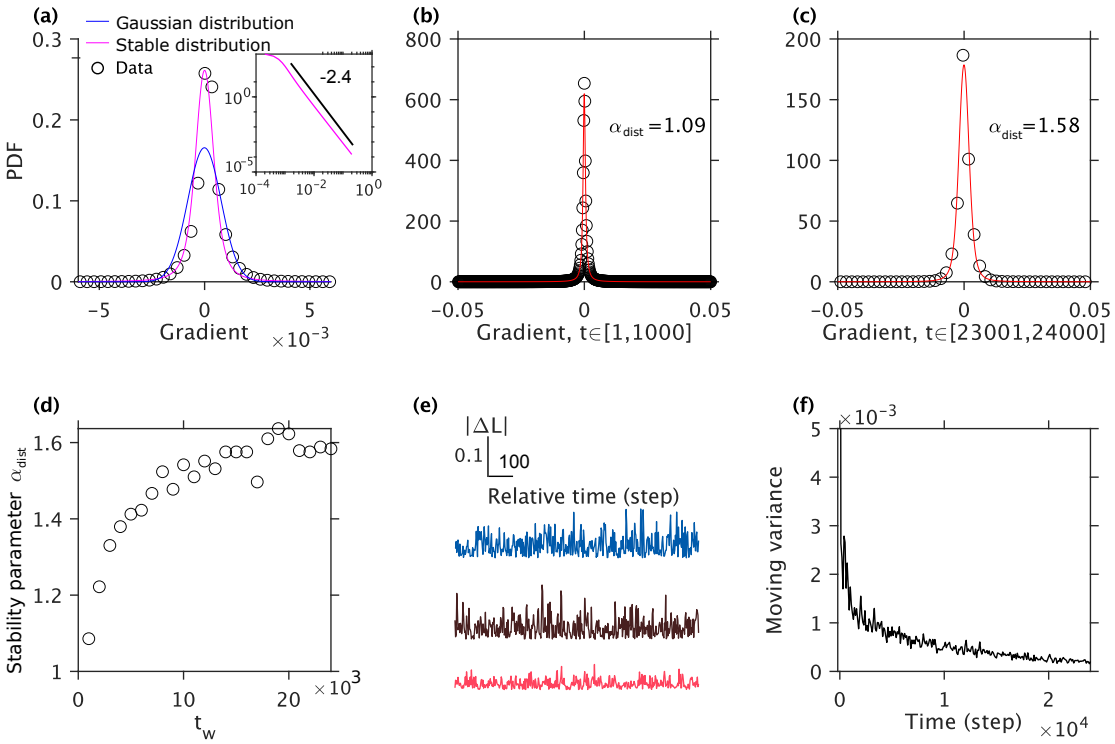

Figure 6: The gradient is heavy-tailed. The distribution of minibatch gradients $\nabla\tilde{L}(\boldsymbol{\Theta})$ in the first regime can be fitted as a symmetric Lévy α-stable distribution (red curve). Inset: the log-log plot of the positive part of the distribution indicates that it has a power-law tail. (b) When $t \in [1, 1000]$, $\nabla\tilde{L}(\boldsymbol{\Theta})$ can be fitted in symmetric Lévy α-stable distribution with the stability parameter $\alpha_{\text{dist}} = 1.09$. (c) Same as in (b) but for $t \in [23001, 24000]$. (d) The fitted stability parameter in the interval of $[t_w, t_w + T]$ as a function of t_w. (e) Fluctuations of absolute value of change-of-loss ΔL decrease as t_w increases. Coloarmap is the same as in Fig. 2(a). (f) The moving variance of loss as a function of time.

gradients $\nabla\tilde{L}$. We first introduce the definition of the Lévy α-stable distribution. Given a Lévy stable random variable X, it is defined by the characteristic function [70]

$$\varphi(u;\alpha_{\text{dist}},\beta,\gamma,\delta)=\exp\left(iu\delta-|\gamma u|^{\alpha_{\text{dist}}}(1-i\beta\text{sgn}(u)\Phi)\right) \tag{4.7}$$

where $\text{sgn}(\cdot)$ is the sign function and

$$\Phi=\begin{cases}\tan\left(\frac{\pi\alpha_{\text{dist}}}{2}\right) & \alpha_{\text{dist}}\neq 1\\ -\frac{2}{\pi}\log|t| & \alpha_{\text{dist}}=1\end{cases},$$

α_{dist} is the stability parameter with the range $0<\alpha_{\text{dist}}<2$. The probability density function (PDF) decays with a power-law tail $|x|^{-\alpha_{\text{dist}}-1}$ which is slower compared to Gaussian distributions; thus the distribution is heavy-tailed [156]. When $\alpha_{\text{dist}}=2$, the distribution is Gaussian. β is the skewness parameter. In particular, for a symmetric Lévy α-stable ($\mathcal{S}\alpha\mathcal{S}$) random variable X, i.e. $X\sim\mathcal{S}\alpha\mathcal{S}$, the skewness parameter $\beta=0$ which indicates the PDF is symmetric around 0. γ is the scale parameter and δ is the shift/location parameter.

Heavy-tailed gradients. We next estimate the minibatch gradient $\nabla\tilde{L}$ in Eq. 4.1 (vanilla SGD) with respect to each w_i at each time point. In the first regime ($t_w<t_0$, $t_0=21000$), it can be fitted to a symmetric Lévy α-stable distribution by the maximum likelihood method; stability parameter $\alpha_{\text{dist}}=1.46528$ $[1.46509,1.46546]$ (the brackets denote the 95% confidence interval).

The power-law tail of the distribution of $\nabla\tilde{L}$ shown in Fig. 6(a) inset further validates the heavy-tailed distribution. We further compare log-likelihood ratios between the fitted Lévy α-stable distribution and Gaussian distribution, and find that the log-likelihood ratios (1.52×10^9) are sufficiently positive, indicating that the distributions follow a Lévy α-stable distribution ($p<10^{-15}$, Vuong test).

To illustrate the evolution of gradient distribution, we also fit the distributions of gradients in intervals $[t_w,t_w+T]$ to Lévy α-stable distribution; the log-likelihood ratios compared with Gaussian distribution are sufficiently positive. Figure 6(b) demonstrates the distribution in the first interval, $t_w=1$; the stability parameter $\alpha_{\text{dist}}=1.09$ $[1.05,1.12]$. Figure 6(c) demonstrates the distribution in the final interval, $t_w=23001$ and correspondingly $\alpha_{\text{dist}}=1.58$ $[1.54,1.63]$. The stability parameter α_{dist} increases as training evolves (Fig. 6(d)), indicating a reduction in the heavy-tailedness of gradient distribution. The fluctuations of gradient values increase as the the tails of the distribution becomes heavier; as the changes of gradient are directly related to the MSD (Eq. 4.1 and Eq. 4.2), the result regarding the changes of gradient distributions is consistent with the time-inhomogeneous anomalous diffusion dynamics where superdiffusion attenuates gradually to subdiffusion. It is also interesting to note that in the physics literature, it has been found that the increase of the heavy-tailedness of step sizes of random walkers

results in super-diffusive motions [149]. Such superdiffusive processes with intermittent long-range jumps help the optimizer escape local minima, facilitating basin hopping during the initial exploratory phase of training. This point is further illustrated below, based on a phenomenological model of SGD. Entering and leaving local minima give rise to the fluctuations of loss values (L). To demonstrate the changes of these fluctuations, we first calculate the absolute value of change-of-loss $|\Delta L| = L(\mathbf{w}(t+1) - \mathbf{w}(t))$. As shown in Fig. 6(e), as t_w increases, $|\Delta L|$ decreases. Such behavior is further quantified by the decreasing moving variance of the loss L against time (Fig. 6(f)); the moving variance is calculated over a sliding window of 100 steps across neighboring L.

4.4 Discussion

We have revealed the anomalous diffusion nature of deep learning dynamics which arises from the interactions of the SGD walker with the geometry structure of the loss landscape. We have found that the SGD optimizer moves from rougher (fractal-like) regions to flatter regions of the loss landscape. During this process, its dynamics evolve from superdiffusion to subdiffusion. The superdiffusive dynamics enable the SGD optimizer to rapidly explore the loss landscape without being trapped in narrow minima; the subdiffusion, on the other hand, consolidates the residence of the SGD in the flatter areas with good solutions. Additionally, we have also outlined a framework that entails a more general description of the training process in Appendix 4.A. We emphasize that in order to obtain analytical results, matrix-valued stochastic differential equations (SDEs) must undergo further progression, especially for modeling correlated tensor-valued processes.

Supplementary Materials

4.A Dysonian dynamics of SGD

As stochasticity arises naturally when a network is being trained, an increasing amount of work has applied random matrix theory (RMT) in describing the singularvalues of the weight matrices [13], or the eigenvalues of the loss Hessian [157]. There have also been tremendous efforts of using (discrete) stochastic processes [56], or stochastic differential equations (SDEs) [57, 58] to describe first-order optimization methods of deep neural networks (DNNs), most of which consider the vectorized or flattened weights, ignoring the matrix structure. Thus, we integrate tools from both stochastic analysis and RMT to propose a more complete description of deep learning, where we outline a framework for modeling the weights as a matrix-value diffusion process, in particular a random non-Hermitian matrix driven by the Ginibre ensemble [60]; this in turn elicits the eigenvalue dynamics during training which is generally disregarded in the context of stochastic modelling. It is well known that the initialization of a random network is crucial to training, i.e. bad initialization could consequentially retard training.

The first application of random matrices was introduced by Wigner to model the nuclei of heavy atoms [158]. Since then, random matrices and matrix-valued diffusion processes have gained interests in physics, biology, finance, etc [159]. Empirically, the training process contains highly complex dynamics, especially at the beginning and the minibatch noise generated is often not Gaussian [34, 14]. Hence, we restrict ourselves to small learning rates, large batch sizes and Gaussian initialization of weights prior to training which empirically allows one to incorporate the Brownian motion. Under these conditions, one can understand how the eigenvalues and eigenvectors of the weight matrices evolve during training, that further elucidates the transformation of the input across all layers during learning. Here, we outline a set of equations for describing the matrix processes of connectivity weights in a multilayer perceptron during training. Given the underdevelopment of matrix-valued stochastic differential equations (SDEs), especially in the non-Hermitian setting, we outline our formulation and discuss the challenges faced when tackling this problem.

For our notation, we will denote by $\mathcal{M}_{m,n}(\mathbb{R})$ as the set of all $m \times n$ matrices with entries in $\mathbb{R}$; if $m = n$, we will write $\mathcal{M}_n(\mathbb{R})$ instead. The group of all invertible matrices of $\mathcal{M}_p$ (general linear group) will be represented as $\mathrm{GL}(p)$. Similarly for a matrix-valued Brownian motion $\mathbf{B} = (B)_{i,j}$ in $\mathcal{M}_{n,p}$, i.e. its entries $B_{i,j}$ are independent one-dimensional Brownian motions for $1 \leq i \leq n$ and $1 \leq j \leq p$, we write $\mathbf{B} \sim \mathcal{BM}_{n,p}$ and $\mathbf{B} \sim \mathcal{BM}_n$ if $n = p$. For $\mathbf{A} \in \mathcal{M}_{m,n}(\mathbb{R})$ with columns $\mathbf{a}_i \in \mathbb{R}^m$, $i = 1, \cdots, n$, we define a function $vec : \mathcal{M}_{m,n} \to \mathbb{R}^{mn}$ via

$$vec(\mathbf{A}) = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_n \end{pmatrix} . \tag{4.8}$$

Throughout this section, bold letters represent matrices or vectors, while scalars are represented in standard form unless otherwise stated.

4.A.1 Formulation and future directions

Recently, a resurgence of studies on mulilayer perceptrons (MLPs) has demonstrated that with image patching or gating, MLPs can compete with convolution neural networks (CNNs) and vision Transformers (ViT) in visual tasks [160, 161]. Despite the lack of convolutions or attention as well as the difficulty of training historically, MLPs still have desirable qualities such as its simplicity and richness in input-output functional expressitivity [162, 163]. These results further spark interests in research beyond various DNN architectures, and fundamental apparatuses of neural information processing.

Let us again consider a wide MLP of depth L, so that both the pre- and post-activation at each layer l have N_l neurons described by a neural activity vector $\mathbf{x}^l$ along with a weight matrix $\mathbf{W}^l \in \mathcal{M}_{N_l,N_{l-1}}(\mathbb{R})$. The propagation dynamics arising from the input $\mathbf{x}^0 \in \mathbb{R}^N$ is given by

$$\mathbf{x}^l = \phi(\mathbf{h}^l) \ , \qquad \mathbf{h}^l = \mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l \ , \tag{4.9}$$

where $\mathbf{h}^l$ is the pre-activation input at layer l, $\mathbf{b}^l$ is a bias vector, and ϕ is an element-wise nonlinear activation function. The training process of a supervised learning task encompasses minimizing the discrepancy between the predictions of the network and the data label of a training dataset $\mathcal{D} := \{(\mathbf{x}^i, y^i)\}_{i=1}^D$ where D is the total number of training samples, the inputs $\mathbf{x}^i$ and labels y^i are samples from the probability spaces $(\mathcal{X}, \mathbb{P}_\mathcal{X})$ and $(\mathcal{Y}, \mathbb{P}_\mathcal{Y})$ respectively. To do so efficiently, a first-order algorithm stochastic gradient descent (SGD) or any of its variants can be applied for this optimization problem. Let us write $\mathrm{NN}_{\boldsymbol{\Theta}} : \mathcal{X} \to \mathcal{Y}$ and the network prediction as $\hat{y}^i := \mathrm{NN}_{\boldsymbol{\Theta}}(\mathbf{x}^i)$. For a pre-specified loss function f, the overall (empirical) loss $L(\boldsymbol{\Theta}) := \frac{1}{D} \sum_{i=1}^D f_{\boldsymbol{\Theta}}(\hat{y}^i, y^i)$ is minimized via approximating

the overall loss L with minibatches $\tilde{L}_k := \frac{1}{|B_k|}\sum_{i\in B_k} f_{\boldsymbol{\Theta}}(\hat{y}^i, y^i)$ and realize Eq. 4.1 where $\boldsymbol{\Theta} \in \mathbb{R}^d$ corresponds to the flattened vector containing all the weights (and biases).

SGD is a highly complex process where the training and testing accuracy encounter rapid changes at early stages, and gradually stabilizes towards the end [1]. One can write down a general Itô SDE for describing the matrix diffusion process as in Eq. (4.9) for $l = 1, \cdots, L$:

$$\begin{cases} \mathrm{d}\mathbf{W}_t^l = -\nabla L(\mathbf{W}_t^l)\mathrm{d}t + \sigma^l(\mathbf{W}_t^l)\mathrm{d}\mathbf{B}_t, \\ \mathbf{W}_{T_1}^l = \tilde{\mathbf{w}}_0^l. \end{cases} \tag{4.10}$$

where $\sigma^l : \mathcal{M}_{N_l,N_{l-1}}(\mathbb{R}) \to \mathcal{M}_{N_l,p}$ is a measurable function and $\mathbf{B} \sim \mathcal{BM}_{p,N_{l-1}}$ is the matrix of independent Brownian motion processes. Note that we have a set of coupled equation where there are also pairwise correlations between $\mathbf{W}^l$'s particularly introduced from backpropagation. For simplicity, let us consider the case $p = N_{l-1} = N_l$, enforcing a square matrix structure for the above so the associated eigen-decomposition can be performed. A general Itô SDE for describing the matrix diffusion process can be formulated as shown, but the exploration of underlying eigenvalue and eigenvector dynamics remains largely unexplored. Current studies have focused mainly on driftless cases with constant diffusion terms, which only allows for addressing only additive noise [164, 165]. Future directions should explore the impact of multiplicative noise, non-square matrices, general covariances and more complex interactions among layers to better understand training dynamics.

[1] This is assuming that the learning hyperparameters such as the learning rate, batch size, etc are fixed at some point.

Chapter 5

Conclusion

In this thesis, we explored the inherent heterogeneity within various deep learning models, challenging the conventional assumption of Gaussian statistics often used in theory and practice. To address discrepancies between theoretical analyses and commonly observed experimental results, we propose heavy-tailed dynamics as a core distributional framework, where classical methods fall short due to the divergence of second moments. The analytical tools employed, rooted in fields like non-equilibrium statistical physics, condensed matter physics, stochastic analysis, and geometric harmonics, provide precise descriptions of large systems, either in the infinite system limit or under infinitesimally small time scales.

The class of Lévy α-stable distributions plays a key role in our studies, where the case of the stability parameter $\alpha < 2$ is subjected to an asymptotic power-law tail. Moreover, the Gaussian dynamics is recovered when $\alpha \to 2$, enabling reconciliation with the classically established theories. For development of a more general framework for understanding deep neural networks motivated by emprically found heavy-tailed weights [34, 13], our first study begins with the fundamental MLP architecture with power-law weights Chapter 2. Treating a MLP as a time-discretizaton of the rate-based model and along with their shared Jacobian structure establish common ground where the same techniques from non-Hermitican heavy-tailed random matrix theory can be transferred [23]. A similar type of extended criticality can also be observed in both MLPs and convolutional neural networks (CNNs) that closely match the experimental results such as the phase transition of the performance according to our proposed initialization scheme. Multifractality also emerges in spatial sites of the Jacobian eigenvector and persists during training. We continue our investigation by adapting Lévy processes which entail heavy-tailed jump sizes, we implement a spatially fractional diffusion sampling method in the self-attention mapping of Transformers and consequently derive the underlying second-order equation Chapter 3. In Chapter 4, we first demonstrate anomalous diffusion dynamics that emerge in the learning

process of various types of deep networks, further reinforcing the importance of heterogeneous neural networks. In the following Appendix 4.A, we outline a formulation that puts forward a matrix-valued stochastic process description of stochastic gradient descent, which is a part of ongoing work. To the author's knowledge, matrix-valued stochastic processes do not generally omit closed forms, especially for deep learning as a complex system. A grounded approach would be to examine special cases where isotropic (additive) noises are the driving process of the SDE, from here onwards an analytical form can be thus derived for the eigenvalue and eigenvectors dynamics.

It has been recognized that artificial intelligence (AI) systems excel in very specific domains when compared to humans or animals but often lack general flexibility and energy efficiency. For applications in reinforcement learning, small perturbations to the environment or input pixels can leads to major performance degradation [166]. Additionally, the training process of large models such as GPT3 consume overwhelming amounts of energy [167] in comparison to the human brain [168]. Despite this, the development of deep learning architectures is frequently detached from biologically plausible principles. Gaussian networks with independent weights have been the central focus of many theoretical studies, providing invaluable insights to aspects of their properties. Nonetheless, they become progressively difficult to apply in real-world scenarios as the finiteness of the second moment is often violated in experimental observations. Additionally, the associated initialization scheme also relies on a high level of precision according to the edge-of-chaos which scales inversely with respect to layer depth [77], contradicting phenomena of commonly found complex systems operating in a region of stretched criticality [24, 25]. Our framework for quantifying information propagation of deep heterogeneous networks offers a more accurate explanation for the singular value density found in pretrained models from the perspective of heavy-tailed universality [13]. Similarly, major efforts have been directed towards designing a more computationally efficient self-attention mapping in Transformers, where the biological aspects of attention are vastly overlooked. Additionally, the global nature of training models necessitate that weight parameters must be known for computing gradients, diverging from local learning rules found in biological systems [169]. These inadequacies are underpinned by the differences between artificial and biological systems, and therefore push for a research direction that unifies both, which has recently been proposed as Neuroscience-Inspired Artificial Intelligence, or *NeuroAI* in short [170, 103]. On the contrary, deriving a unified theory for the brain needs accurate experimental measurements which are generally difficult to achieve. Observations of artificial networks can be obtained with greater precision as they stand on a set of pre-defined rules. Established theories under imposed assumptions can be relatively easily verified and specific dynamics unveiled from the neurobiological systems can also be implemented as relevant mechanisms.

Hence artificial neural networks serve not only as powerful learning tools but also as a theoretical testbed for neuroscience.

For future directions, an immediate generalization would be to extend our theory from MLPs [59] and RNNs [23] to other architectures such as CNNs. Further incorporating correlations into the analysis of random networks would be of great significance for gaining insights into heavy-tailed singular spectrum observed in pretrained models [13]. Moreover, as the fundamental driving force of applications, stochastic gradient descent (SGD) encompasses the evolution of underlying networks towards feature extraction. Additional mathematical insights are required for progression on the intersection of stochastic analysis and random matrix theory such that a rigorous theory can be directly applied to create a dynamical theory for learning. Free probability, a field that generalizes random variables to non-commutative matrix objects, is undergoing active development but remains limited by assumptions such as free independence [171]. Our ongoing work explores these limitations, beginning with simpler covariance structures for the driving process.

In the modern era of super-exponential data generation, deep learning has become a critical tool for processing vast amounts of information, advancing both user-centered applications and scientific discovery. As AI benefits from neuroscience, the future of deep learning will depend on continued scientific advancements towards the field specifically, fostering innovative methodologies in mathematics and physics. In this thesis, we have illustrated the intersection between AI and science by developing theoretical frameworks for heavy-tailed dynamics in deep learning models and learning processes, offering new perspectives on the shared principles governing both artificial and biological systems.

Bibliography

[1] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105.

[3] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[4] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589.

[5] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[6] Michael E. Sander et al. "Sinkformers: Transformers with Doubly Stochastic Attention". In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, 28–30 Mar 2022, pp. 3515–3530.

[7] H. Sompolinsky, A. Crisanti, and H. J. Sommers. "Chaos in Random Neural Networks". In: *Phys. Rev. Lett.* 61 (3 July 1988), pp. 259–262.

[8] Radford M. Neal, P. Diggle, and S. Fienberg. *Bayesian Learning for Neural Networks*. New York, NY, UNITED STATES: Springer New York, 1996.

[9] Ben Poole et al. "Exponential expressivity in deep neural networks through transient chaos". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona, Spain: Curran Associates Inc., 2016, pp. 3368–3376.

[10] Jaehoon Lee et al. *Deep Neural Networks as Gaussian Processes*. Oct. 2017.

[11] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. "On the Impact of the Activation function on Deep Neural Networks Training". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 2672–2680.

[12] Charles H. Martin and Michael W. Mahoney. "Heavy-Tailed Universality Predicts Trends in Test Accuracies for Very Large Pre-Trained Deep Neural Networks". In: *arXiv e-prints*, arXiv:1901.08278 (Jan. 2019), arXiv:1901.08278.

[13] Charles H. Martin, Tongsu (Serena) Peng, and Michael W. Mahoney. "Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data". In: *Nature Communications* 12.1 (2021), p. 4122.

[14] Guozhang Chen, Cheng Kevin Qu, and Pulin Gong. "Anomalous diffusion dynamics of learning in deep neural networks". In: *Neural Networks* 149 (2022), pp. 18–28.

[15] Umut Simsekli et al. "Fractional Underdamped Langevin Dynamics: Retargeting SGD with Momentum under Heavy-Tailed Gradient Noise". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 8970–8980.

[16] Samuel S. Schoenholz et al. "Deep Information Propagation". In: *International Conference on Learning Representations*. 2017.

[17] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778.

[18] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. June 2018.

[19] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. "The emergence of spectral universality in deep networks". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. Proceedings of Machine Learning Research: PMLR, Sept. 2018, pp. 1924–1932.

[20] Thommen George Karimpanal and Roland Bouffanais. "Self-organizing maps for storage and transfer of knowledge in reinforcement learning". In: *Adaptive Behavior* 27.2 (Dec. 2018), pp. 111–126.

[21] Jean-Philippe Bouchaud and Marc Potters. *Theory of Financial Risk and Derivative Pricing: From Statistical Physics to Risk Management*. Cambridge University Press, Dec. 2003.

[22] Didier Sornette. *Critical phenomena in natural sciences*. en. 2nd ed. Springer Series in Synergetics. Berlin, Germany: Springer, Mar. 2006.

[23] Asem Wardak and Pulin Gong. "Extended Anderson Criticality in Heavy-Tailed Neural Networks". In: *Phys. Rev. Lett.* 129 (4 July 2022), p. 048103.

[24] Paolo Moretti and Miguel A. Muñoz. "Griffiths phases and the stretching of criticality in brain networks". In: *Nature Communications* 4.1 (2013), p. 2521.

[25] Leticia F. Cugliandolo et al. "Multifractal phase in the weighted adjacency matrices of random Erdös-Rényi graphs". In: *arXiv e-prints*, arXiv:2404.06931 (Apr. 2024), arXiv:2404.06931.

[26] Asem Wardak and Pulin Gong. "Fractional diffusion theory of balanced heterogeneous neural networks". In: *Phys. Rev. Research* 3 (1 Jan. 2021), p. 013083.

[27] Yifan Chen et al. "Skyformer: Remodel Self-Attention with Gaussian Kernel and Nystr\"om Method". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 2122–2135.

[28] Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer New York, 1998.

[29] David Applebaum. *Lévy Processes and Stochastic Calculus*. 2nd ed. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2009.

[30] Guozhang Chen and Pulin Gong. "A spatiotemporal mechanism of visual attention: Superdiffusive motion and theta oscillations of neural population activity patterns". In: *Science Advances* 8.16 (2022), eabl4995.

[31] Yang Qi and Pulin Gong. "Fractional neural sampling as a theory of spatiotemporal probabilistic computations in neural circuits". In: *Nature Communications* 13.1 (2022), p. 4572.

[32] Gennady Samorodnitsky and Murad S. Taqqu. *Stable non-Gaussian random processes: stochastic models with infinite variance*. 1994.

[33] Anna Lischke et al. "What is the fractional Laplacian? A comparative review with new results". In: *Journal of Computational Physics* 404 (2020), p. 109009.

[34] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. "A Tail-Index Analysis of Stochastic Gradient Noise in Deep Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR. Sept. 2019, pp. 5827–5837.

[35] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80.

[36] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *arXiv e-prints*, arXiv:1301.3781 (Jan. 2013), arXiv:1301.3781.

[37] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166.

[38] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv e-prints*, arXiv:1406.1078 (June 2014), arXiv:1406.1078.

[39] Ashish Vaswani et al. "Attention is All you Need". In: vol. 30. Curran Associates, Inc., 2017.

[40] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021.

[41] Sebastian Ruder. *On word embeddings - Part 1*. http://ruder.io/word-embeddings-1/. 2016.

[42] Jorge Pérez, Pablo Barceló, and Javier Marinkovic. "Attention is Turing-Complete". In: *Journal of Machine Learning Research* 22.75 (2021), pp. 1–35.

[43] Hyunjik Kim, George Papamakarios, and Andriy Mnih. "The Lipschitz Constant of Self-Attention". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 5562–5571.

[44] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. "Attention is not all you need: pure attention loses rank doubly exponentially with depth". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 2793–2803.

[45] A. Singer. "From graph to manifold Laplacian: The convergence rate". In: *Applied and Computational Harmonic Analysis* 21.1 (2006). Special Issue: Diffusion Maps and Wavelets, pp. 128–134.

[46] Alexander Grigor and Takashi Kumagai. "On the dichotomy in the heat kernel two sided estimates". In: 77 (Jan. 2008).

[47] Ronald R. Coifman and Stéphane Lafon. "Diffusion maps". In: *Applied and Computational Harmonic Analysis* 21.1 (2006). Special Issue: Diffusion Maps and Wavelets, pp. 5–30.

[48] Boaz Nadler et al. "Diffusion Maps - a Probabilistic Interpretation for Spectral Embedding and Clustering Algorithms". In: *Principal Manifolds for Data Visualization and Dimension Reduction*. Ed. by Alexander N. Gorban et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 238–260.

[49] Ning Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural Networks* 12.1 (Jan. 1999), pp. 145–151.

[50] Alex Graves. "Generating Sequences With Recurrent Neural Networks". In: *arXiv e-prints*, arXiv:1308.0850 (Aug. 2013), arXiv:1308.0850.

[51] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA, 2015.

[52] Timothy Dozat. "Incorporating Nesterov Momentum into Adam". In: *Proceedings of the 4th International Conference on Learning Representations*. 2016, pp. 1–4.

[53] Zachary C Lipton. "Stuck in a what? adventures in weight space". In: *The International Conference on Learning Representations (ICLR) workshop*. 2016.

[54] Elad Hoffer, Itay Hubara, and Daniel Soudry. "Train longer, generalize better: closing the generalization gap in large batch training of neural networks". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1731–1741.

[55] Simon Becker, Yao Zhang, and Alpha A Lee. "Geometry of Energy Landscapes and the Optimizability of Deep Neural Networks". In: *Physical Review Letters* 124.10 (2020).

[56] Liam Hodgkinson and Michael Mahoney. "Multiplicative Noise and Heavy Tails in Stochastic Optimization". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. 18–24 Jul 2021, pp. 4262–4274.

[57] Qianxiao Li, Cheng Tai, and Weinan E. "Stochastic Modified Equations and Adaptive Stochastic Gradient Algorithms". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 2101–2110.

[58] Qianxiao Li, Cheng Tai, and Weinan E. "Stochastic Modified Equations and Dynamics of Stochastic Gradient Algorithms I: Mathematical Foundations". In: *Journal of Machine Learning Research* 20.40 (2019), pp. 1–47.

[59] Cheng Kevin Qu, Asem Wardak, and Pulin Gong. "Extended critical regimes of deep neural networks". In: *arXiv e-prints*, arXiv:2203.12967 (Mar. 2022), arXiv:2203.12967.

[60] Jean Ginibre. "Statistical Ensembles of Complex, Quaternion, and Real Matrices". In: *J. Math. Phys.* 6.3 (Mar. 1965), pp. 440–449.

[61] Dong. Yu. *Automatic Speech Recognition A Deep Learning Approach*. eng. 1st ed. 2015. Signals and Communication Technology. London: Springer London, 2015.

[62] Dante R. Chialvo. "Emergent complex neural dynamics". In: *Nature Physics* 6.10 (Oct. 2010), pp. 744–750.

[63] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[64] Terrence J. Sejnowski. "The unreasonable effectiveness of deep learning in artificial intelligence". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30033–30038.

[65] Yasaman Bahri et al. "Statistical Mechanics of Deep Learning". In: *Annual Review of Condensed Matter Physics* 11.1 (2020), pp. 501–528.

[66] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256.

[67] Charles Bordenave, Pietro Caputo, and Djalil Chafaï. "Spectrum of Non-Hermitian Heavy Tailed Random Matrices". In: *Communications in Mathematical Physics* 307.2 (2011), p. 513.

[68] R. Benzi et al. "On the multifractal nature of fully developed turbulence and chaotic systems". In: *Journal of Physics A: Mathematical and General* 17.18 (1984), pp. 3521–3531.

[69] György Buzsáki and Kenji Mizuseki. "The log-dynamic brain: how skewed distributions affect network operations". In: *Nature Reviews Neuroscience* 15.4 (2014), pp. 264–278.

[70] John P. Nolan. "Basic Properties of Univariate Stable Distributions". In: *Univariate Stable Distributions: Models for Heavy Tailed Data*. Cham: Springer International Publishing, 2020, pp. 1–23.

[71] Li Deng. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[72] Jaehoon Lee et al. "Deep Neural Networks as Gaussian Processes". In: *International Conference on Learning Representations*. 2018.

[73] Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. "Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4788–4798.

[74] Tong He et al. "Bag of Tricks for Image Classification with Convolutional Neural Networks". In: *arXiv e-prints*, arXiv:1812.01187 (Dec. 2018), arXiv:1812.01187.

[75] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (1980), pp. 193–202.

[76] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[77] Lechao Xiao et al. "Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10, 000-Layer Vanilla Convolutional Neural Networks". In: *ICML*. 2018.

[78] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research)". In: (2009).

[79] Benoit B. Mandelbrot. "Intermittent turbulence in self-similar cascades: divergence of high moments and dimension of the carrier". In: *Journal of Fluid Mechanics* 62.2 (1974), pp. 331–358.

[80] Tsuneyoshi Nakayama. "Fractal Structures in Condensed Matter Physics". In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A. Meyers. New York, NY: Springer New York, 2009, pp. 3878–3893.

[81] Ferdinand Evers and Alexander D. Mirlin. "Anderson transitions". In: *Rev. Mod. Phys.* 80 (4 Oct. 2008), pp. 1355–1417.

[82] Matthew Farrell et al. "Gradient-based learning drives robust representations in recurrent neural networks by balancing compression and expansion". In: *Nature Machine Intelligence* 4.6 (2022), pp. 564–573.

[83] Peiran Gao et al. "A theory of multineuronal dimensionality, dynamics and measurement". In: *bioRxiv* (2017).

[84] Alessio Ansuini et al. "Intrinsic Dimension of Data Representations in Deep Neural Networks". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[85] Yu Feng and Yuhai Tu. "The inverse variance-flatness relation in stochastic gradient descent is critical for finding flat minima". In: *Proceedings of the National Academy of Sciences* 118.9 (2021), e2015617118.

[86] Thierry Mora and William Bialek. "Are Biological Systems Poised at Criticality?" In: *Journal of Statistical Physics* 144.2 (2011), pp. 268–302.

[87] Kanaka Rajan and L. F. Abbott. "Eigenvalue Spectra of Random Matrices for Neural Networks". In: *Phys. Rev. Lett.* 97 (18 Nov. 2006), p. 188104.

[88] Johnatan Aljadeff, Merav Stern, and Tatyana Sharpee. "Transition to Chaos in Random Networks with Cell-Type-Specific Connectivity". In: *Phys. Rev. Lett.* 114 (8 Feb. 2015), p. 088101.

[89] Charles Bordenave and Djalil Chafaï. "Around the circular law". In: *Probability Surveys* 9.none (2012), pp. 1–89.

[90] Fernando Lucas Metz, Izaak Neri, and Tim Rogers. "Spectral theory of sparse non-Hermitian random matrices". In: *Journal of Physics A: Mathematical and Theoretical* 52.43 (2019), p. 434003.

[91] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv e-prints*, arXiv:1907.11692 (July 2019), arXiv:1907.11692.

[92] Jinhua Zhu et al. "Incorporating BERT into Neural Machine Translation". In: *arXiv e-prints*, arXiv:2002.06823 (Feb. 2020), arXiv:2002.06823.

[93] Shiyang Li et al. "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[94] Tian Zhou et al. "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 27268–27286.

[95] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *arXiv e-prints*, arXiv:2010.11929 (Oct. 2020), arXiv:2010.11929.

[96] Junhan Yang et al. “GraphFormers: GNN-nested Transformers for Representation Learning on Textual Graph”. In: *arXiv e-prints*, arXiv:2105.02605 (May 2021), arXiv:2105.02605.

[97] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *arXiv e-prints*, arXiv:1409.0473 (Sept. 2014), arXiv:1409.0473.

[98] Iz Beltagy, Matthew E. Peters, and Arman Cohan. “Longformer: The Long-Document Transformer”. In: *arXiv e-prints*, arXiv:2004.05150 (Apr. 2020), arXiv:2004.05150.

[99] Sinong Wang et al. “Linformer: Self-Attention with Linear Complexity”. In: *arXiv e-prints*, arXiv:2006.04768 (June 2020), arXiv:2006.04768.

[100] Manzil Zaheer et al. “Big Bird: Transformers for Longer Sequences”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 17283–17297.

[101] Yiping Lu et al. “Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View”. In: *arXiv e-prints*, arXiv:1906.02762 (June 2019), arXiv:1906.02762.

[102] Randolph F. Helfrich et al. “Neural Mechanisms of Sustained Attention Are Rhythmic”. In: *Neuron* 99.4 (2018), 854–865.e5.

[103] Anthony Zador et al. “Catalyzing next-generation Artificial Intelligence through NeuroAI”. In: *Nature Communications* 14.1 (Mar. 2023).

[104] David Applebaum. *Lévy Processes and Stochastic Calculus*. 2nd ed. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2009.

[105] Harbir Antil, Tyrus Berry, and John Harlim. “Fractional diffusion maps”. In: *Applied and Computational Harmonic Analysis* 54 (2021), pp. 145–175.

[106] James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. “A Mathematical Theory of Attention”. In: *arXiv e-prints*, arXiv:2007.02876 (July 2020), arXiv:2007.02876.

[107] James Vuckovic, Aristide Baratin, and Remi Tachet des Combes. “On the Regularity of Attention”. In: *arXiv e-prints*, arXiv:2102.05628 (Feb. 2021), arXiv:2102.05628.

[108] Borjan Geshkovski et al. “The emergence of clusters in self-attention dynamics”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 57026–57037.

[109] Michael. Renardy and Robert C. Rogers. *An Introduction to Partial Differential Equations*. eng. 2nd ed. 2004. Texts in Applied Mathematics, 13. New York, NY: Springer New York, 2004.

[110] Hugo Koubbi, Matthieu Boussard, and Louis Hernandez. “The Impact of LoRA on the Emergence of Clusters in Transformers”. In: *arXiv e-prints*, arXiv:2402.15415 (Feb. 2024), arXiv:2402.15415.

[111] Mateusz Kwaśnicki. “Ten Equivalent Definitions of the Fractional Laplace Operator”. In: *Fractional Calculus and Applied Analysis* 20.1 (2017), pp. 7–51.

[112] Mario Lezcano Casado. “Trivializations for Gradient-Based Optimization on Manifolds”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[113] Ayan Sengupta, Md Shad Akhtar, and Tanmoy Chakraborty. “Manifold-Preserving Transformers are Effective for Short-Long Range Encoding”. In: *The 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.

[114] Zhenzhong Lan et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *International Conference on Learning Representations*. 2020.

[115] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv e-prints*, arXiv:1810.04805 (Oct. 2018), arXiv:1810.04805.

[116] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: 2019.

[117] Xingyu Cai et al. “Isotropy in the Contextual Embedding Space: Clusters and Manifolds”. In: *International Conference on Learning Representations*. 2021.

[118] Andrew L. Maas et al. “Learning Word Vectors for Sentiment Analysis”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150.

[119] Jeffrey Pennington, Richard Socher, and Christopher Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

[120] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv e-prints*, arXiv:1910.01108 (Oct. 2019), arXiv:1910.01108.

[121] Xiaolong Wang et al. "Non-Local Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[122] Anatoly Kochubei and Yuri Luchko, eds. *Volume 1 Basic Theory*. Berlin, Boston: De Gruyter, 2019.

[123] L. Evans, M. K. Cameron, and P. Tiwary. "Computing committors via Mahalanobis diffusion maps with enhanced sampling data". In: *The Journal of Chemical Physics* 157.21 (Dec. 2022), p. 214107.

[124] Danfeng Hong et al. "More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.5 (2021), pp. 4340–4354.

[125] Anna Choromanska et al. "The loss surfaces of multilayer networks". In: *Artificial Intelligence and Statistics*. 2015, pp. 192–204.

[126] Hao Li et al. "Visualizing the loss landscape of neural nets". In: *Advances in Neural Information Processing Systems*. 2018, pp. 6389–6399.

[127] Daniel Jiwoong Im, Michael Tao, and Kristin Branson. "An empirical analysis of the optimization of deep network loss surfaces". In: *arXiv: Learning* (2016).

[128] Yann N Dauphin et al. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

[129] Stanisław Jastrzębski et al. "Three factors influencing minima in sgd". In: *arXiv:1711.04623* (2017).

[130] Max Welling and Yee W Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th International Conference on Machine Learning*. 2011, pp. 681–688.

[131] Pratik Chaudhari and Stefano Soatto. "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks". In: *Information Theory and Applications (ITA) Workshop*. IEEE, 2018, pp. 1–10.

[132] Marco Baity-Jesi et al. "Comparing dynamics: Deep neural networks versus glassy systems". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 314–323.

[133] Mario Geiger et al. "Jamming transition as a paradigm to understand the loss landscape of deep neural networks". In: *Physical Review E* 100.1 (2019), p. 12115.

[134] Jingzhao Zhang et al. "Why ADAM Beats SGD for Attention Models". In: *arXiv:1912.03194* (2019).

[135] Ilya Pavlyukevich. "Cooling down Lévy flights". In: *Journal of Physics A: Mathematical and Theoretical* 40.41 (Sept. 2007), pp. 12299–12313.

[136] Abhishek Panigrahi et al. "Non-gaussianity of stochastic gradient noise". In: *arXiv:1910.09626* (2019).

[137] Charles H Martin and Michael W Mahoney. "Traditional and Heavy-Tailed Self Regularization in Neural Network Models". In: *arXiv:1810.01075* (2019).

[138] Gérard B. Arous and Alice Guionnet. "The Spectrum of Heavy Tailed Random Matrices". In: *Communications in Mathematical Physics* 278 (2009), pp. 715–751.

[139] Xue Li and Yuanzhi Cheng. "Understanding the message passing in graph neural networks via power iteration clustering". In: *Neural Networks* 140 (2021), pp. 130–135.

[140] Gregory B Sorkin. "Efficient simulated annealing on fractal energy landscapes". In: *Algorithmica* 6.1 (1991), p. 367.

[141] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[142] Danfeng Hong et al. "Graph Convolutional Networks for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.7 (2021), pp. 5966–5978.

[143] Patrick Charbonneau et al. "Fractal free energy landscapes in structural glasses". In: *Nature Communications* 5 (2014), p. 3725.

[144] Hyun Joo Hwang, Robert A Riggleman, and John C Crocker. "Understanding soft glassy materials using an energy landscape approach". In: *Nature Materials* 15 (2016), p. 1031.

[145] Yuliang Jin and Hajime Yoshino. "Exploring the complex free-energy landscape of the simplest glass by rheology". In: *Nature Communications* 8.1 (2017), p. 14935.

[146] Penghui Cao, Michael P. Short, and Sidney Yip. "Potential energy landscape activations governing plastic flows in glass rheology". In: *Proceedings of the National Academy of Sciences* 116.38 (2019), pp. 18790–18797.

[147] Ido Golding and Edward C Cox. "Physical nature of bacterial cytoplasm". In: *Physical Review Letters* 96.9 (2006), p. 098102.

[148] Irena Bronstein et al. "Transient anomalous diffusion of telomeres in the nucleus of mammalian cells". In: *Physical Review Letters* 103.1 (2009), p. 018102.

[149] V Zaburdaev, S Denisov, and J Klafter. "Lévy walks". In: *Reviews of Modern Physics* 87.2 (2015), p. 483.

[150] Ralf Metzler and Joseph Klafter. "The random walk's guide to anomalous diffusion: a fractional dynamics approach". In: *Physics Reports* 339.1 (2000), pp. 1–77.

[151] T H Solomon, Eric R Weeks, and Harry L Swinney. "Observation of anomalous diffusion and Lévy flights in a two-dimensional rotating flow". In: *Physical Review Letters* 71.24 (1993), p. 3975.

[152] Gandimohan M Viswanathan et al. "Optimizing the success of random searches". In: *Nature* 401.6756 (1999), p. 911.

[153] Peter Dieterich et al. "Anomalous dynamics of cell migration". In: *Proceedings of the National Academy of Sciences* 105.2 (2008), pp. 459–463.

[154] Luiz GA Alves et al. "Transient superdiffusion and long-range correlations in the motility patterns of trypanosomatid flagellate protozoa". In: *PLoS One* 11.3 (2016), e0152092.

[155] Karthik A Sankararaman et al. "The impact of neural network overparameterization on gradient confusion and stochastic gradient descent". In: *Proceedings of the 37th International Conference on Machine Learning*. 2020.

[156] Joseph Klafter and Igor M Sokolov. *First Steps in Random Walks: from Tools to Applications*. Oxford University Press, 2011.

[157] Levent Sagun, Leon Bottou, and Yann LeCun. "Eigenvalues of the Hessian in Deep Learning: Singularity and Beyond". In: *arXiv e-prints*, arXiv:1611.07476 (Nov. 2016), arXiv:1611.07476.

[158] Eugene P. Wigner. "Random Matrices in Physics". In: *SIAM Review* 9.1 (1967), pp. 1–23.

[159] Marc Potters and Jean-Philippe Bouchaud. *A First Course in Random Matrix Theory: for Physicists, Engineers and Data Scientists*. Cambridge University Press, Nov. 2020.

[160] Hanxiao Liu et al. "Pay Attention to MLPs". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 9204–9215.

[161] Ilya O. Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision*. 2021.

[162] Maithra Raghu et al. "On the Expressive Power of Deep Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 2847–2854.

[163] Patrick Kidger and Terry Lyons. "Universal Approximation with Deep Narrow Networks". In: *Proceedings of Thirty Third Conference on Learning Theory*. Ed. by Jacob Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. PMLR, Sept. 2020, pp. 2306–2327.

[164] Zdzislaw Burda et al. "Dysonian Dynamics of the Ginibre Ensemble". In: *PRL* 113.10 (Sept. 2014), p. 104102.

[165] Jacek Grela and Piotr Warchoł. "Full Dysonian dynamics of the complex Ginibre ensemble". In: *Journal of Physics A: Mathematical and Theoretical* 51.42 (2018), p. 425203.

[166] Sandy H. Huang et al. "Adversarial Attacks on Neural Network Policies". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[167] David A. Patterson et al. "Carbon Emissions and Large Neural Network Training". In: *ArXiv* abs/2104.10350 (2021).

[168] E. Racker and Herman Kabat. "The metabolism of the central nervous system in experimental Poliomyelitis". In: *Journal of Experimental Medicine* 76.6 (Dec. 1942), pp. 579–585.

[169] James A. Henderson and Pulin Gong. "Functional mechanisms underlie the emergence of a diverse range of plasticity phenomena". In: *PLOS Computational Biology* 14.11 (Nov. 2018). Ed. by Abigail Morrison, e1006590.

[170] Demis Hassabis et al. "Neuroscience-Inspired Artificial Intelligence". In: *Neuron* 95.2 (July 2017), pp. 245–258.

[171] Dan Voiculescu, Nicolai Stammeier, and Moritz Weber. "Free Probability and Operator Algebras". In: 2016.