Topological Invariance and Breakdown in Learning Dynamics

Yongyi Yang[1,2*] **Tomaso Poggio**[3] **Isaac L. Chuang**[3] **Liu Ziyin**[3†]

[1]Harvard University [2]NTT Research [3]Massachusetts Institute of Technology

Abstract

While mainstream theories of deep learning focus on training with a small learning rate, a growing body of theoretical and empirical work suggests that neural network training dynamics can be qualitatively different at large learning rates. However, a clear and general understanding of the fundamental differences between training at small and large learning rates remains lacking. In this work, we prove that if a group of parameter vectors (neurons) in a neural network model exhibits permutation symmetry, then widely used training algorithms induce a continuous mapping on the set formed by these vectors. Moreover, when the learning rate is below a certain threshold, this mapping becomes a homeomorphism. Our result reveals a critical point of the learning rate: below it, the training dynamics are guaranteed to preserve the topology of the set of neurons, whereas above it, training may introduce simplifications to the underlying neuron manifold. This provides a topological description of a novel implicit regularization effect of small learning rates and offers a potential theoretical lens for empirical phenomena such as loss of plasticity. Notably, our theory is independent of specific network architectures and loss functions, enabling topology to be applied universally to deep learning theory.

1 Introduction

A primary difficulty in analyzing the learning dynamics of neural networks lies in their extremely high dimensionality across diverse architectural details. Modern neural networks can have more than 10^{12} parameters, inducing such complicated dynamics that conventional tools and theories of dynamical systems struggle to apply. Lessons from natural science and many fields of mathematics suggest two primary approaches [Noether, 1918, Mumford et al., 1994]: (1) study what the high-dimensional object is invariant to, and (2) decompose it into simpler parts. The first approach directly reduces the dimensionality of a problem, while the second allows us to view it as a composition of low-dimensional objects. Our theory, presented in this paper, aims to offer a crucial link between the two perspectives: whenever the learning rule is equivariant under permutations of a group of parameter vectors (i.e. neurons), the high-dimensional dynamics decomposes into a system of interacting low-dimensional neurons. This criterion covers fully connected layers, convolutional layers, and the query-key product in attention mechanisms, so the decomposition applies layer by layer to most architectures in use.

Specifically, under standard regularity conditions, we show that:

1. The permutation equivariance of common learning algorithms imposes strong topological constraints on the learning dynamics;[3]

*Email: yongyiyang@fas.harvard.edu

†Corresponding author. Email: ziyinl@mit.edu

[3]The word “topology" is sometimes used to refer to the model architecture. In our work, it always means the mathematical topology of sets.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

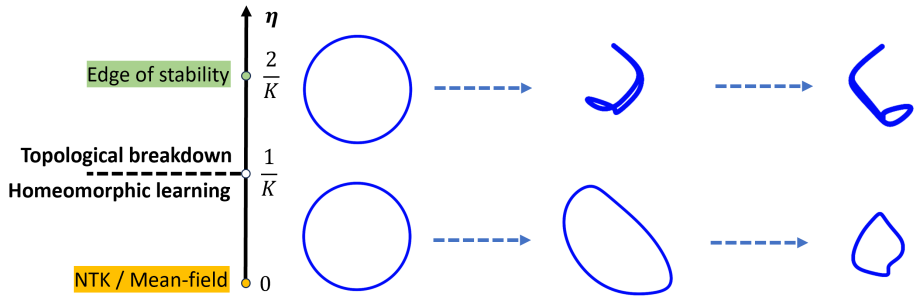


Figure 1: At a small learning rate, common learning algorithms induce a homeomorphic transformation of the neuron distribution (blue shapes in Figure), a mechanism underlying common theories including the NTK / lazy regime [Jacot et al., 2018, Chizat et al., 2018] and the mean-field / feature-learning regime [Yang and Hu, 2020]. In contrast, most neural networks in real training scenarios are known to move towards the "edge of stability," where the discrete-time updates are no longer stable at any first-order stationary point. From the perspective of topology, what separates these two regimes is the **topology invariance** in the first regime, where the learning process is strongly constrained to preserve any topological properties, and the **topological breakdown** in the second regime, where the learning ceases to preserve topology and can act as a simplifier that merges neurons and thereby reduces the model's capacity.

2. With a small learning rate η, the learning algorithm induces a bi-Lipschitz mapping between neurons at different time steps, thereby preserving the topological structure of the set formed by the neurons.
3. With a large η, this topological invariance may break down: the learning algorithm still induces a continuous surjection, thereby permitting (but not forcing) topological simplification to happen during training.

The core contribution of this paper can be summarized as the theoretical establishment of a critical point of the topological phase transition (hereinafter referred to as the **topological critical point**) for learning processes. Figure 1 illustrates the theory.

A key feature of our theory is that it relies only on several widely satisfied properties of the learning algorithm and is therefore universal across architectures and optimizers. The system-independence of our result allows us to establish a key conceptual principle: *any permutation-equivariant dynamics induces a topology between its components, and this topology is preserved at a small step size and reduced at a large step size*. This universality lends it good potential to serve as a foundation for future theories.

From the perspective of implicit regularization, our theory identifies a concrete way in which the learning rate biases the effective function class. With a small learning rate, the induced bi-Lipschitz mapping enforces topology preservation, restricting training to configurations reachable without changing neuronal topological invariants. Beyond the topological critical point, the mapping may become a continuous surjection that merges neurons (or regions of the neuron manifold), thereby reducing topology and effectively compressing the model. Thus, the learning rate modulates implicit regularization by controlling whether training preserves or simplifies neuronal topology. This introduces a new perspective on the effect of learning rate, complementing prior analyses on implicit regularization that interpret learning rate as effectively changing the optimized objective [Barrett and Dherin, 2020, Smith et al., 2021].

Why should the topology of a set of neurons matter? In neuroscience, the space of neuron labels is often the represented space itself: place cells are labeled by the locations of their place fields, and the topology of the environment can be recovered from their co-activity [Curto and Itskov, 2008]; head-direction cells are organized on a ring [Chaudhuri et al., 2019], and grid cells on a torus [Gardner et al., 2022]. An artificial analogue is a radial basis network $f(\boldsymbol{x}) = \sum_i a_i \, \phi(\|\boldsymbol{x} - \boldsymbol{c}_i\|)$, whose neurons $(\boldsymbol{c}_i, a_i)$ are receptive fields: representing a periodic variable calls for centers arranged on a circle, and merging centers removes coverage of the input domain. Similar structure emerges in trained networks; in models trained on modular arithmetic, the neurons arrange approximately into a few circles, one per key Fourier frequency [Nanda et al., 2023]. Our results imply that below the critical learning rate such organizations cannot change type, while above it they can only be coarsened.

This paper is organized as follows. Section 2 reviews the background. Section 3 introduces the problem setting. Section 4 presents our main theory. Section 5 applies the theory to common training algorithms. Section 6 presents empirical results. Finally, Section 7 discusses further implications of our theory. All proofs of the theoretical results are deferred to the appendix.

2 Background

Permutation Symmetry. Permutation symmetry refers to the invariance of a function's output under permutations of its inputs. This property is pervasive in neural networks and has been widely used to analyze their loss landscapes [Entezari et al., 2021, Brea et al., 2019, Ziyin, 2024]. For example, any neural network component (such as a layer) that has the following structure:

$$f(\boldsymbol{x}; \boldsymbol{W}_1, \boldsymbol{W}_2) = \boldsymbol{W}_2 \sigma\left(\boldsymbol{W}_1 \boldsymbol{x}\right), \tag{1}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2$ are learnable parameter matrices, $\boldsymbol{x}$ is the input vector and σ is a scalar activation function (applied element-wisely), possesses permutation symmetry, as

$$f(\boldsymbol{x}; \boldsymbol{W}_1, \boldsymbol{W}_2) = \left(\boldsymbol{P}\boldsymbol{W}_2^\top\right)^\top \sigma\left((\boldsymbol{P}\boldsymbol{W}_1)\boldsymbol{x}\right) = f(\boldsymbol{x}; \boldsymbol{P}\boldsymbol{W}_1, \boldsymbol{W}_2\boldsymbol{P}^\top) \tag{2}$$

for any permutation matrix $\boldsymbol{P}$. If we pair the i-th row of $\boldsymbol{W}_2$ with the i-th column of $\boldsymbol{W}_1$ together as a unit (which together form a "neuron" in our theory), then the symmetry can be understood as that, the model remains unchanged under exchanging two neurons $(\boldsymbol{w}_{1,i}, \boldsymbol{w}_{2,i}) \leftrightarrow (\boldsymbol{w}_{1,j}, \boldsymbol{w}_{2,j})$.

Structures in the form of eq. (1) are quite common in all types of neural networks, including convolutional layers, feed-forward layers, and the QK transformation in the self-attention layers in transformers. Specifically, the QK transformation in transformers can be represented as

$$\text{attn}(\boldsymbol{X}; \boldsymbol{W}_Q, \boldsymbol{W}_K) = \text{softmax}\left(\boldsymbol{X}\boldsymbol{W}_Q\boldsymbol{W}_k^\top\boldsymbol{X}^\top\right), \tag{3}$$

where $X \in \mathbb{R}^{n\times d}$ is the input of the self-attention layer, and it is clear that a sub-module of QK transformation satisfies the structure defined in eq. (1), with $\boldsymbol{W}_2 = \boldsymbol{W}_Q$, $\boldsymbol{W}_1 = \boldsymbol{W}_k^\top$ and σ being the identity mapping. All those components therefore possess permutation symmetry, and fall within the scope of our theory. A key consequence of the permutation symmetry is the *permutation equivariance* of the learning algorithms, such as (stochastic) gradient descent and Adam, which we will take as the starting point of our theory.

Critical Learning Rates. Recent empirical studies suggest that neural networks exhibit qualitatively different learning dynamics under small versus large learning rates. Large learning rates often lead to simpler models [Galanti and Poggio, 2022, Chen et al., 2023, Dohare et al., 2024], while such dramatic changes seem to be lacking with the use of small learning rates, where the learning dynamics are often well-approximated by the NTK or the mean-field theories [Jacot et al., 2018, Yang and Hu, 2020, Mei et al., 2019]. In this work, we establish a topological characterization of these transitions.

Topology. Topology is the mathematical study of abstract shapes and connectivity, focusing on properties that remain unchanged under continuous deformations such as stretching or bending [Kuratowski, 2014]. It provides a way to talk about local and global structures without relying on exact distances; a bijective continuous map with a continuous inverse is called a homeomorphism and preserves the topology of general sets. Manifolds are, roughly speaking, sets that locally look like Euclidean space, and their smooth structure is preserved under smooth invertible maps, called diffeomorphisms [Lang, 2012]. In the context of deep learning, topological perspectives have been widely adopted to understand the properties of neural networks [Bucarelli et al., 2024, Barannikov et al., 2020, Horoi et al., 2022, Naitzat et al., 2020, Purvine et al., 2023, Nurisso et al., 2024, Birdal et al., 2021]. However, prior studies have largely remained empirical and focused on specific networks, whereas our work connects the topology of neurons systematically to the training dynamics.

3 Preliminaries

Now, we temporarily set aside considerations of specific neural networks and learning algorithms, and instead focus on more general and abstract objects. We will consider a (possibly infinite) collection of high-dimensional particles (corresponding to neurons) and their dynamics (corresponding to learning algorithms). In the following, we use the terms "particles" and "neurons" interchangeably.

Definition 3.1 (Neurons)**.** Partition a group of learnable parameters into D-dimensional vectors $\{\boldsymbol{x}_i\}_{i\in I}$. We call these vectors *neurons* if the update rule is equivariant under every finitary permutation of them, i.e., permuting the vectors before an update gives the same result as permuting them after it (property P1 below).

Notations. Let I be an arbitrary potentially uncountable set, which we often refer to as the index set. In practice, I is usually finite (one index per hidden unit), and all results below then reduce to statements about bijections of finite point sets. We allow infinite I so that the same statements also cover the mean-field (infinite-width) limit, in which the output is an integral of neurons against a probability measure (Example 3.2). Throughout this paper, we focus on a collection of D-dimensional vectors indexed by I. We use $(\mathbb{R}^D)^I$ to represent the set of all such collections. We use calligraphic uppercase letters to denote collections indexed by I (e.g. $\mathcal{X} \in \left(\mathbb{R}^D\right)^I$), bold lowercase letters to denote vectors (e.g. $\boldsymbol{x} \in \mathbb{R}^D$), and unbold lowercase letters to denote scalars or an entry of a vector or matrix (e.g. $x_k \in \mathbb{R}$ represents the k-th entry of $\boldsymbol{x}$). For $i \in I$, and a vector $\boldsymbol{v} \in \mathbb{R}^D$, we use $\boldsymbol{e}_i \cdot \boldsymbol{v} \in \left(\mathbb{R}^D\right)^I$ to denote a collection of D-dimensional vectors where only the i-th element is $\boldsymbol{v}$ and other vectors are $\mathbf{0}$.

Let $\mathsf{FSym}(I)$ be the Finitary Permutation Group on I, i.e. the group of all permutation operators on I with a finite support [Neumann, 1976]. For an operator $P \in \mathsf{FSym}(I)$ and $\mathcal{X} = \{\boldsymbol{x}_i\}_{i\in I}$, we use $P\mathcal{X}$ to represent $\left\{\boldsymbol{x}_{P(i)}\right\}_{i\in I}$.

For $\mathcal{X} = \{\boldsymbol{x}_i\}_{i\in I} \in \left(\mathbb{R}^D\right)^I$ and $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i\in I} \in \left(\mathbb{R}^D\right)^I$, if $\mathcal{X}$ and $\mathcal{Y}$ only differ in finite many terms, then we define $\|\mathcal{X} - \mathcal{Y}\| = \sqrt{\sum_{i\in I} \|\boldsymbol{x}_i - \boldsymbol{y}_i\|^2}$, and $\|\mathcal{X} - \mathcal{Y}\| = +\infty$ otherwise.

Problem Setting. Formally, we focus on an evolving collection of D-dimensional vectors $\mathcal{X}^{(t)} = \left\{\boldsymbol{x}_i^{(t)}\right\}_{i\in I} \in \left(\mathbb{R}^D\right)^I$, where $t \in \mathbb{N}$ is the time axis. $\mathcal{X}^{(t)}$ is updated by a generic update rule $U^{(t)} : \left(\mathbb{R}^D\right)^I \to \left(\mathbb{R}^D\right)^I$ with step size $\eta > 0$:

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \eta U_i^{(t)}\left(\mathcal{X}^{(t)}\right), \tag{4}$$

where $U_i^{(t)}(\mathcal{X}) = \left(U^{(t)}(\mathcal{X})\right)_i$. Here, each element in $\mathcal{X}^{(t)}$ corresponds to the weights of a neuron of a neural network at training step t, and $U^{(t)}$ corresponds to the learning algorithm at time point t, which includes regularization terms, and can also depend on other parameters that are not considered (this is also why the update rule is time-dependent, as other parameters can change over time).

Abstractly, we consider update rules $U^{(t)}$ satisfying the following properties.

- **(P1) Equivariance Property**: We say $U^{(t)}$ has equivariance property if for any $t \in \mathbb{N}$, any $\mathcal{X} \in \left(\mathbb{R}^D\right)^I$ and $P \in \mathsf{FSym}(I)$, we have $PU^{(t)}(\mathcal{X}) = U^{(t)}(P\mathcal{X})$. In deep learning, this property is a consequence of running gradient-based algorithms on permutation-symmetric loss functions, as we will show in Section 5. That many dynamics are naturally equivariant has been studied in physics [Field, 1980], but its role in deep learning is not yet clear.
- **(P2-K) K-Continuity Property**: For $K > 0$, if for any $t \in \mathbb{N}$ and any $\mathcal{X}, \mathcal{Y} \in \left(\mathbb{R}^D\right)^I$,

  $$\left\|U^{(t)}(\mathcal{X}) - U^{(t)}(\mathcal{Y})\right\| \le K\|\mathcal{X} - \mathcal{Y}\|, \tag{5}$$

  then we say $U^{(t)}$ has K-continuity property. Notice that eq. (5) makes sense only when $\mathcal{X}$ and $\mathcal{Y}$ only differ in finite entries. When I is finite and $U^{(t)}$ is gradient descent, the quantity K is the largest eigenvalue of the Hessian of the loss function, and the K-continuity property becomes an upper bound of the Lipschitz continuity of the gradient, which is commonly seen in optimization theory (See details in Section 5). We intentionally choose this form to establish this correspondence; however, it is possible to prove our theory with a weaker version of the continuity property. See Appendix B.2 for details.

Now, it is important to clarify the use of the word "neuron." With P1 in place, Definition 3.1 makes this precise and follows Ziyin [2024]. In case of fully connected networks trained with GD, this definition of a "neuron" is equivalent to the standard definition (incoming plus outgoing weights of an activation unit).

If, beyond topology, we also want to talk about the differentiable manifold structure of the neurons, we further consider a smoothness property of U.

- **(P3) Smoothness Property**: For any $i \in I$ and $t \in \mathbb{N}$, define

  $$\forall \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^D, g_i^{(t)}(\boldsymbol{y}, \boldsymbol{z}) = U_i^{(t)}\left(\mathcal{X}^{(t)} + (\boldsymbol{e}_i - \boldsymbol{e}_j)\cdot \boldsymbol{z}\right), \text{ such that } \boldsymbol{y} = \boldsymbol{x}_j^{(t)},$$

where j is arbitrarily chosen when multiple j-s satisfy the condition, and $\boldsymbol{e}_j$ is set to $\mathbf{0}$ if no j satisfies the condition. If $g_i^{(t)}$ is C^1 on $\left(\mathbb{R}^D\right)^2$ for any $i \in I$, we say $U^{(t)}$ has the (C^1-)smoothness property. Intuitively, the smoothness property requires that the response of each output entry of U with respect to a small perturbation of one entry of its input must be C^1.

Example 3.2 (A circle of neurons)**.** Consider a one-hidden-layer network in the mean-field parameterization [Mei et al., 2018, Chizat and Bach, 2018], $f(\boldsymbol{x}) = \int_I a(\theta)\,\sigma(\langle \boldsymbol{w}(\theta), \boldsymbol{x}\rangle)\,\mathrm{d}\mu(\theta)$, with $I = S^1$, μ the uniform probability measure, and neuron θ given by $(\boldsymbol{w}(\theta), a(\theta)) = (\cos\theta, \sin\theta, 1) \in \mathbb{R}^3$. Then $f(\boldsymbol{x}) = \frac{1}{2\pi}\int_0^{2\pi} \sigma(x_1\cos\theta + x_2\sin\theta)\,\mathrm{d}\theta$, which equals $\|\boldsymbol{x}\|/\pi$ for $\sigma = \mathrm{ReLU}$ and is bounded by 1 for the sigmoid. The neuron set $S = \{(\cos\theta, \sin\theta, 1)\}$ is a compact one-dimensional submanifold of $\mathbb{R}^3$ homeomorphic to S^1. A finite network is the special case where μ is an empirical measure, and under gradient flow the per-neuron velocity field is exactly an update rule U with uncountable I (the neuron distribution follows a Wasserstein gradient flow; see also Theorem B.1).

4 Evolution of Neuron Topology

In this section, we present our main theoretical results: the characterization of the change of topology and measure structures of $\mathcal{X}$ under the update rule. Before diving into the main theorems, we first establish two critical lemmas. These lemmas show that, a combination of the equivariance and continuity of the update rule implies that there is an emergent notion of distance between different neurons. Intuitively, permutation equivariance implies that two infinitesimally close neurons need to have identical updates, which implies that the motion that changes their difference must be vanishing. Thus, equivariance ensures that neurons that start close to each other remain close because dynamics that would increase or decrease their distance are suppressed.

Lemma 4.1 (Well-definedness)**.** *The following statement holds when $U^{(t)}$ satisfies P1. For any $i, j \in I$ such that $i \neq j$, if at time t we have $\boldsymbol{x}_i^{(t)} = \boldsymbol{x}_j^{(t)}$, then, $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_j^{(t+1)}$.*

Next, we strengthen the intuition behind Lemma 4.1 by incorporating the continuity property.

Lemma 4.2 (No Merging or Splitting)**.** *If $U^{(t)}$ satisfies P1 and P2-K, then for any $i, j \in I$ such that $i \neq j$,*

$$(1-\eta K)\left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\right\| \leq \left\|\boldsymbol{x}_i^{(t+1)} - \boldsymbol{x}_j^{(t+1)}\right\| \leq (1+\eta K)\left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\right\|. \tag{6}$$

This lemma implies the bi-Lipschitzness of the update rule between the manifolds formed by neurons at consecutive time steps t and $t+1$. The fact that common learning rules induce bi-Lipschitz mappings is nontrivial, as such maps are known to preserve topological invariants (as we will show in the subsequent section) and control geometric distortions [Heinonen, 2001].

Moreover, Lemma 4.2 also identifies a critical learning rate $\eta^* = 1/K$, beyond which the lower bound becomes vacuous, which we referred to as *topological critical point* hereinafter. As we will see in the next section, this marks a phase transition from bijective, homeomorphic dynamics to merely surjective continuous dynamics.

4.1 Topological Invariance

A crucial perspective is implied by the lemmas above: the entirety of the neurons can be seen as a set (or, manifold) $S \subset \mathbb{R}^D$, and the evolution of neurons can be viewed as the evolution of S. Of course, a crucial question is whether such a perspective is meaningful, which is the key question we answer in this section.

Formally, let $S^{(t)} = \left\{ \boldsymbol{x}_i^{(t)} \middle| i \in I \right\} \subseteq \mathbb{R}^D$ denote the set formed by all of the neurons in $\mathcal{X}^{(t)}$, equipped with the relative topology inherited from $\mathbb{R}^D$. Define function $\widehat{U}^{(t)} : S^{(t)} \to S^{(t+1)}$ by

$$\forall i \in I, \widehat{U}^{(t)}\left(\boldsymbol{x}_i^{(t)}\right) = \boldsymbol{x}_i^{(t+1)}. \tag{7}$$

Intuitively, $\widehat{U}^{(t)}$ describes the effect of $U^{(t)}$ on each point of $S^{(t)}$. Lemmas 4.1 and 4.2 together ensure that $\widehat{U}^{(t)}$ is well-defined and, under small learning rates, a bijection.

Lemma 4.3. *If $U^{(t)}$ satisfies P1 and P2-K, then $\widehat{U}^{(t)}$ is well-defined, and is a surjection. If additionally $\eta K < 1$, then $\widehat{U}^{(t)}$ is a bijection.*

One can show that, $\widehat{U}^{(t)}$ is not only a bijection, but also a homeomorphism between $S^{(t)}$ and $S^{(t+1)}$. Note that $S^{(t)}$ need not be finite: for infinite I it can be, e.g., a curve or an open subset of $\mathbb{R}^D$ (Example 3.2), and the openness assumption in (iv) below targets this mean-field setting. This leads to our main theorem.

Theorem 4.4 (Main). *If $U^{(t)}$ satisfies P1 and P2-K, then*

i. $\widehat{U}^{(t)}$ is a continuous surjection from $S^{(t)}$ to $S^{(t+1)}$;
ii. if $S^{(t)}$ is compact, then $S^{(t+1)}$ is also compact, and $\widehat{U}^{(t)}$ is a quotient map;
iii. if $\eta K < 1$, then $\widehat{U}^{(t)}$ is a homeomorphism;
iv. if $\eta K < 1$, and $U^{(t)}$ also satisfies P3, and $S^{(t)}$ is an open subset of $\mathbb{R}^D$, then $S^{(t+1)}$ is also open, and $\widehat{U}^{(t)}$ is a C^1-diffeomorphism.

See Appendix A.4 for the proof of Theorem 4.4. This result shows that when the learning rate is below the critical threshold $\eta^* = 1/K$, the neuronal set $S^{(t)}$ evolves through homeomorphisms (or diffeomorphisms under smoothness assumptions), so its topology is invariant throughout training: neurons initially arranged as a circle, torus, or any other manifold preserve that topological type, and separated neurons cannot merge unless identical at initialization, while merged neurons cannot split. Thus small-learning-rate training only locally deforms neuron topology by translating, expanding, or contracting local densities. This diffeomorphic evolution supports mean-field descriptions, including NTK theory, at small learning rates [Mei et al., 2019, Jacot et al., 2018]: the neuron distribution $P_t(w)$ obeys standard change-of-variable formulas and leads to Vlasov-type equations in the infinite-width limit [Spohn, 2012]. Since the argument is architecture-independent, it may point toward a general mean-field theory of deep learning. By contrast, at large learning rates, homeomorphic evolution breaks down and merging or other topology-changing events become possible, marking the *topological critical point* predicted by our theory and verified experimentally (Section 6). However, topology cannot change arbitrarily: Lemma 4.1 rules out neuron splitting, so the induced map $\widehat{U}^{(t)}$ remains a quotient map and can only produce a coarser topology. This coarsening reduces expressivity/capacity by effectively gluing neurons together; equivalently, from the perspective of permutation symmetry, merging neurons moves the network to a symmetric state and reduces the effective number of parameters by the number of weights in a neuron, as in Proposition 3 of Ziyin et al. [2025].

The theoretical results presented in this section can be further extended to measure invariance and other smoothness conditions. See Appendix B for additional theoretical results.

Role of K. So far, we have formally treated the smoothness parameter K as a global quantity, which leads to the elegant and easy-to-state results above. However, it is much better to conceptually treat K as a local quantity in a small neighborhood around the current parameter θ: $K \approx K(\theta)$. When the learning rule is SGD, this $K(\theta)$ can be approximated by the largest eigenvalue of the local Hessian $\lambda_{\max}(H(\theta))$. In this more dynamical perspective, K can be seen as a dynamically evolving quantity. When viewed with the phenomenon of the edge of stability [Cohen et al., 2021, Wu et al., 2018], this picture suggests a two-phase perspective of the learning process of common neural networks, where the first phase of training focuses on optimizing the loss and learning the task, while the second phase of learning is a simplification process, where the model tends to simpler and coarser topologies, a process that could be related to phenomena such as grokking [Power et al., 2022].

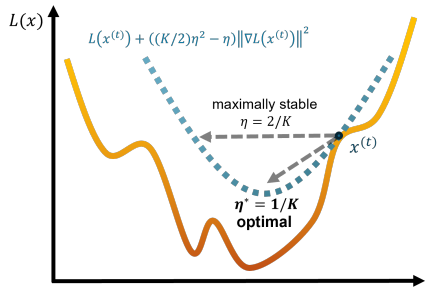


Figure 2: **An optimization perspective of the topological critical point**. The topological critical point $\eta^* = 1/K$ corresponds to the step size that reduces the loss optimally, while the critical step size found by Cohen et al. [2021] corresponds to the largest one ensuring loss decay.

4.2 "Approximate" Topology

So far, our results are about the precise notion of mathematical topology of the neurons, which primarily addresses the case in which the number of neurons is considered infinite. While this is

especially useful for existing theories that study infinite-size neural networks [Jacot et al., 2018, Xiao et al., 2018, Karakida et al., 2019], it is not so useful for more practical scenarios where there is only a finite number of neurons and exact merging of neurons is a probability-zero event for generic trajectories, so that in practice one observes neurons becoming extremely close rather than exactly coincident. In this section, we establish a form of topology that is more applicable to realistic systems. To formalize this, we first define the r-expansion of a set.

Definition 4.5. For a set $P \subseteq \mathbb{R}^d$ and a scalar $r > 0$, the r**-expansion** of P is defined as

$$P^r = \left\{ y \in \mathbb{R}^d \middle| \exists x \in P, \|x - y\| < r \right\}. \tag{8}$$

The r-expansion formalizes the intuition that two neurons must be sufficiently separated to be regarded as encoding distinct information. The definition used here coincides with the standard notion of expansion widely used in the study of metric spaces, and this definition is conceptually similar to the Vietoris-Rips complex widely used in topological data analysis [Zomorodian and Carlsson, 2004]. For any set $P \subseteq \mathbb{R}^d$, its r-expansion is naturally an open set and inherits the relative topology from $\mathbb{R}^d$.

Corollary 4.6. *If $U^{(t)}$ satisfies P1 and P2-K and $\eta K < 1$, then for any $r < \frac{1}{2} \inf_{\substack{x,y \in S^{(t)} \\ x \neq y}} \|x - y\|$, we have $\left(S^{(t)}\right)^r$ is homeomorphic to $\left(S^{(t+1)}\right)^{(1-\eta K)r}$.*

See Appendix A.10 for the proof of Corollary 4.6. Note that when $r = 0$, Corollary 4.6 reduces to Theorem 4.4.iii. When $r > 0$, however, it provides a quantitative description of how the scale of the set changes.

4.3 "Approximate" Expressivity

Now, we show that the approximate topology defined in the previous section can be directly linked to the expressivity of the model being studied.

Definition 4.7. Let $\epsilon > 0$ be a positive real number. Define the ϵ**-covering number** at time t to be the minimum number m, such that there exists a subset P with $|P| = m$ and $S^{(t)} \subseteq P^\epsilon$.

The following corollary is an irreversibility statement: neurons that are close at some time cannot quickly become distinguishable again.

Corollary 4.8. *Suppose $U^{(t)}$ satisfies P1 and P2-K. Let $T \in \mathbb{N}_+$, $\epsilon < \exp(-\eta K T)/2$, let $E_\epsilon(t)$ be the ϵ-covering number at time t, then we have*

$$E_1(t + T) \leq E_\epsilon(t). \tag{9}$$

The following Lemma illustrates how the ϵ-covering number is related to the expressivity of the network.

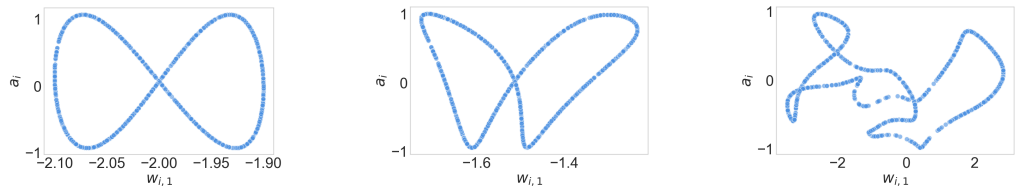
Lemma 4.9. *Suppose the index set I is finite, and the ϵ-covering number at time t is m, then there exists a function $h : \left(\mathbb{R}^D\right)^I \to \left(\mathbb{R}^D\right)^m$ such that the following statement holds. For any R-Lipschitz continuous function $f : \left(\mathbb{R}^D\right)^I \to \mathbb{R}$ that maps the neurons to a scalar output (i.e. the network), there exists another function $g : \left(\mathbb{R}^D\right)^m \to \mathbb{R}$, such that*

$$\left| g\left(h\left(\mathcal{X}^{(t)}\right)\right) - f\left(\mathcal{X}^{(t)}\right) \right| < R|I|\epsilon. \tag{10}$$

The proofs of Corollary 4.8 and Lemma 4.9 are provided in Appendix A.11 and Appendix A.12 respectively. The dependence on ϵ can be further tightened with extra smoothness conditions [Wang et al., 2026]. Notice that the result in this section is, in a sense, spiritually dual to Corollary 4.6, as the former pins down the lower end of the scale (how close two neurons must be to be treated as the same neuron), while the latter pins down the upper end (how far apart two neurons must be to be treated as separated).

5 Examples: Gradient Descent and Adam

Starting from this section, we connect our abstract theoretical discussions to actual optimization algorithms, by proving that the properties of the update rule $U^{(t)}$ we have used are satisfied by a wide range of optimization algorithms. Here we analyze two of the most popular ones, namely Gradient Descent (GD) and Adam, as illustrative cases.



(a) Initialization (b) End of training, small η (c) End of training, large η

Figure 3: **Topology of a 2D neural network with GD.** The neurons are initialized on a genus-2 surface and optimized with GD. We visualize the topology of 2D and 3D networks before and after training under different step sizes η. For small step sizes, the training may deform the geometric arrangement of the neurons but the topology remains unchanged. In contrast, for large step sizes, the topological structure can change substantially. These results consistently verify our theoretical predictions that while the geometry of the neurons can be affected by training, the underlying topology is stable under small learning rates but fragile under large ones.

Gradient Descent. Let us assume that there is a loss function $L : \left(\mathbb{R}^D\right)^I \to \mathbb{R}$ that maps the neurons to a scalar loss value, and the update rule $U^{(t)}$ is the gradient descent update[4]:

$$U^{(t)}(\mathcal{X}) = -\nabla L(\mathcal{X}). \tag{11}$$

Now we prove that, the equivariance property of U comes from the permutation symmetry of L and the continuity property comes from the smoothness of L.

Proposition 5.1. *If L has $\mathsf{FSym}(I)$-symmetry, i.e.*

$$\forall \mathcal{X} \in \left(\mathbb{R}^D\right)^I, \forall P \in \mathsf{FSym}(I), L(\mathcal{X}) = L(P\mathcal{X}), \tag{12}$$

then $U^{(t)}$ defined in eq. (11) satisfies P1.

Proposition 5.2. *If there exists a constant $K > 0$, such that for any $\mathcal{X}, \mathcal{Y} \in \left(\mathbb{R}^D\right)^I$ and any $i \in I$, we have*

$$\|\nabla L(\mathcal{X}) - \nabla L(\mathcal{Y})\| \le K \|\mathcal{X} - \mathcal{Y}\|, \tag{13}$$

then $U^{(t)}$ defined in eq. (11) satisfies P2-K.

Adam. Besides GD, other more complicated and modern optimizers are usually stateful – they need to keep track of some values in the process of training, which at first sight seems incompatible with our definition of $U^{(t)}$, since our $U^{(t)}$ is stateless by definition. This seeming difficulty can be resolved by a small trick: we can view the state of an optimizer as a part of the neurons, thereby rewriting the update rule in a stateless form.

Specifically, suppose $\boldsymbol{\theta}_i^{(t)}$ is the i-th neuron in the neural network at time t, the collection of particles is defined as $\mathcal{X}^{(t)} = \left\{\left(\boldsymbol{\theta}_i^{(t)}, \boldsymbol{m}_i^{(t)}, \boldsymbol{v}_i^{(t)}\right)\right\}_{i\in I}$, where $\boldsymbol{m}_i^{(t)}$ and $\boldsymbol{v}_i^{(t)}$ are the first order and second order moment estimators in Adam. The update rule is then defined as[5]

$$U^{(t)} : \left\{\begin{pmatrix}\boldsymbol{\theta}_i\\ \boldsymbol{m}_i\\ \boldsymbol{v}_i\end{pmatrix}\right\}_{i\in I} \mapsto \left\{\begin{pmatrix} -\frac{\boldsymbol{m}_i/(1-\beta_1^t)}{\epsilon+\sqrt{\boldsymbol{v}_i/(1-\beta_2^t)}}\\ \frac{1-\beta_1}{\eta}\left[\nabla_i L(\Theta) - \boldsymbol{m}_i\right]\\ \frac{1-\beta_2}{\eta}\left[\nabla_i L(\Theta)^2 - \boldsymbol{v}_i\right]\end{pmatrix}\right\}_{i\in I}, \tag{14}$$

where β_1 and β_2 are the decay rates for the first- and second-order moment estimators, respectively; $\Theta = \{\boldsymbol{\theta}_i\}_{i\in I}$ is the collection of the neurons; and all scalar operations (square, division, square root) are taken element-wise. It is straightforward to check that the neuron update in eq. (14) is equivalent to the standard Adam rule. One can now prove the following theorem.

[4]For this case, the right-hand side of eq. (11) is independent of time t, and therefore $U^{(t)}$ is the same at each time. However, we would love to keep this redundancy of notation, because here we have actually made a subtle (but harmless) simplification, that we implicitly assume all neurons that are to be updated have permutation symmetry, while in practice there can be a part of learnable parameters that are not permutation-invariant, absorbing which into L, although does not affect our discussion here, will make the loss function L time-dependent and so does the update rule.

[5]Here and in the appendix, we optionally write the tuple in the tall form for better presentation.

Proposition 5.3. *If L has $\mathsf{FSym}(I)$-symmetry (eq. (12)), then $U^{(t)}$ defined in eq. (14) satisfies P1.*

The K-continuity of Adam can be proved in a bounded set of the finite-dimensional parameter space. The boundedness is generally satisfied as long as the learning does not diverge.

Proposition 5.4. *Suppose $|I| = M \in \mathbb{N}$. Let $L : (R^D)^M \to \mathbb{R}$ be a smooth function, such that $\nabla L(\boldsymbol{\theta})$ satisfies Equation* (13)*. Let $U = U^{(t)}$ be defined as in eq. (14), with $\beta_1, \beta_2 \in [0, 1), \epsilon > 0$, and $\eta > 0$. Let $B \subseteq (R^D)^M$ be a bounded set, then there exists a constant number $K' > 0$, such that for any $\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}, \boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}' \in B$ where $\boldsymbol{v}, \boldsymbol{v}' \geq \epsilon$, we have*

$$\|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\| \leq K' \|(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - (\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\|. \tag{15}$$

Scope. The theory is criterion-based rather than architecture-based: it applies to any pair of architecture and learning rule that satisfies P1 and P2-K (and P3 for the smooth statements). P1 holds whenever the loss is invariant under permuting the relevant parameter groups. Besides GD and Adam, this covers SGD, since every mini-batch loss keeps the symmetry, and momentum, via the same state-augmentation argument as for Adam (Figures 5 and 7). On the architecture side, it covers fully connected and convolutional layers and the query-key product in attention, applied layer by layer (Definition 3.1).

6 Experiments

Our experimental results include both direct visualizations of the topological structure in low-dimensional networks and quantitative measurements that capture topological properties in networks trained on standard tasks. Additional results and detailed experimental settings are in Appendices C and D.

Low-dimensional Distributions. We first train neural networks with low-dimensional neurons (2- or 3-dimensional) in order to directly visualize their topology. Specifically, consider a two-layer neural network $F : \mathbb{R}^d \to \mathbb{R}$ with hidden layer size h, defined as

$$F\left(\boldsymbol{z}; \{(\boldsymbol{w}_i, a_i)\}_{i=1}^h\right) = \sum_{i=1}^h a_i \sigma\left(\langle \boldsymbol{w}_i, \boldsymbol{z} \rangle\right), \tag{16}$$

where $\boldsymbol{w}_i \in \mathbb{R}^d$ and $a_i \in \mathbb{R}$ are learnable parameters, and σ denotes the sigmoid function. The network is trained on data generated by a random teacher network (See Appendix D for details). In this setting, the loss function has the permutation symmetry as described in eq. (12), with $I = [h]$ and $\mathcal{X} = \{(\boldsymbol{w}_i, a_i)\}_{i \in I} \in (\mathbb{R}^{d+1})^I$.

For visualization purposes, we focus on $d = 1$ and $d = 2$, so that each element in $\mathcal{X}$ lies in $\mathbb{R}^2$ or $\mathbb{R}^3$ (referred to below as 2D and 3D networks, respectively), which enables a straightforward visualization of their topology. Moreover, we initialize the elements in $\mathcal{X}$ with specific topological structures to highlight potential topological (in)variance. See Figure 3 for the results with GD and 2D networks, and Appendix C for extra results with other optimizers and other dimensions.

Topological Invariants. For higher-dimensional models, it is generally harder to directly visualize the geometry of the neurons, but we can verify the theoretical prediction by measuring the topological invariants. In Appendix C.2, we provide experiment results with higher-dimensional models trained on real image classification tasks. We measure the Betti numbers b_1, b_2 of the point cloud formed by the neurons. Betti numbers are fundamental topological invariants that count the number of connected components, loops, and higher-dimensional voids in a topological space; they are widely used in topological data analysis as compact descriptors of shape and structure [Edelsbrunner and Harer, 2010, Naitzat et al., 2020].

Loss of Plasticity. One recently discovered phenomenon that can be directly related to topology is the loss of plasticity [Abbas et al., 2023]. We define a random task-generating process. In each epoch, we regenerate the data with a new random task and continue training the same model. See Appendix D.3 for the details of the setting. In Figure 14, we present the loss and the number of collapsed neuron pairs after each epoch. It is clear that, in the large learning rate setting, the model is not able to learn, and the number of collapsed neurons is increasing, showing the loss of plasticity of the neural network; while in the small learning rate setting, there are no collapsed neurons and the model is able to continually learn the tasks even when they are changed at each epoch.

7 Discussion

We have studied the interplay between permutation symmetries, learning rates, and neuron topology in neural-network training. Our main message is that permutation symmetry, whether arising from architecture or learning algorithms, imposes strong topological constraints on learning dynamics. This perspective helps explain several empirical phenomena and suggests new directions for algorithm design. A limitation of this work is that it is theoretical: we do not test the predictions in large-scale experiments, and we only discuss a subset of the possible implications of a topological theory of deep learning.

Topological Description of Dynamics. Our results provide a topological characterization of learning-rate regimes: small learning rates preserve the topology of the neuron manifold, whereas large learning rates may enable topology-changing events such as neuron merging. This distinction offers a structural view of learning-rate scheduling. For example, a large initial learning rate may help training move across different topological configurations, while later decay may stabilize optimization within a fixed topology. This complements existing explanations of large-learning-rate phenomena, such as the "catapult mechanism" [Lewkowycz et al., 2020], though further work is needed to identify when such transitions occur in practical networks.

Phase Transition. From a physics perspective, topology changes are naturally interpreted as phase transitions, as in systems distinguished by different Chern numbers. In our setting, these transitions also correspond to changes in parameter symmetry and are therefore Landau-type phase transitions. For instance, a transition from genus-1 to genus-2 topology can be viewed as neuron merging, which restores permutation symmetry by moving the network from a symmetry-broken state to a symmetric state [Ziyin, 2024].

Deep Learning Theory. Our results also clarify the limitations of conventional theories of learning dynamics. The EOS phenomenon suggests that GD often reaches solutions with sharpness approximately $2/\eta$, sometimes early in training. Thus practical dynamics may quickly enter a regime near or beyond the topological critical point, where theories designed for small learning rates, such as NTK and mean-field theory, may no longer approximate the true dynamics. Developing a theory for this topological-breakdown regime remains an important open problem.

Practical Implications. Our results point to several concrete next steps: topology-aware learning-rate schedules, which use a large rate to move between topological configurations and then decay it to freeze one; monitoring the number of collapsed neuron pairs as a diagnostic of loss of plasticity (Figure 14); and connecting topological coarsening to network compression, since the same permutation symmetry underlies recent results showing that permutation-invariant functions of d objects can be compressed to functions of $\mathrm{polylog}\, d$ objects, with implications for the lottery ticket hypothesis [Wang et al., 2026].

Acknowledgments and Disclosure of Funding

The authors thank Hong-Yi Wang for discussion.

Funding. ILC acknowledges support in part from the Institute for Artificial Intelligence and Fundamental Interactions (IAIFI) through NSF Grant No. PHY-2019786. This work was also supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

Competing interests. The authors declare no competing interests.

References

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. *arXiv preprint arXiv:2303.07507*, 2023.

Serguei Barannikov, Daria Voronkova, Ilya Trofimov, Alexander Korotin, Grigorii Sotnikov, and Evgeny Burnaev. Topological obstructions in neural networks learning. *arXiv preprint arXiv:2012.15834*, 2020.

David GT Barrett and Benoit Dherin. Implicit gradient regularization. *arXiv preprint arXiv:2009.11162*, 2020.

Tolga Birdal, Aaron Lou, Leonidas J Guibas, and Umut Simsekli. Intrinsic dimension, persistent homology and generalization in neural networks. *Advances in neural information processing systems*, 34:6776–6789, 2021.

Johanni Brea, Berfin Simsek, Bernd Illing, and Wulfram Gerstner. Weight-space symmetry in deep networks gives rise to permutation saddles, connected by equal-loss valleys across the loss landscape. *arXiv preprint arXiv:1907.02911*, 2019.

Maria Sofia Bucarelli, Giuseppe Alessio D'Inverno, Monica Bianchini, Franco Scarselli, and Fabrizio Silvestri. A topological description of loss surfaces based on betti numbers. *Neural Networks*, 178: 106465, 2024.

Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature Neuroscience*, 22(9):1512–1520, 2019.

Feng Chen, Daniel Kunin, Atsushi Yamamura, and Surya Ganguli. Stochastic collapse: How gradient noise attracts sgd dynamics towards simpler subnetworks. *arXiv preprint arXiv:2306.04251*, 2023.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, 2018.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Carina Curto and Vladimir Itskov. Cell groups reveal structure of stimulus space. *PLoS Computational Biology*, 4(10):e1000205, 2008.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024.

Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

MJ561832 Field. Equivariant dynamical systems. *Transactions of the American Mathematical Society*, 259(1):185–205, 1980.

Tomer Galanti and Tomaso Poggio. Sgd noise and implicit low-rank bias in deep neural networks. *arXiv preprint arXiv:2206.05794*, 2022.

Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. Toroidal topology of population activity in grid cells. *Nature*, 602:123–128, 2022.

Noah Golmant, Zhewei Yao, Amir Gholami, Michael Mahoney, and Joseph Gonzalez. pytorch-hessian-eigenthings: efficient pytorch hessian eigendecomposition, October 2018. URL `https://github.com/noahgolmant/pytorch-hessian-eigenthings`.

Walter Helbig Gottschalk and Gustav Arnold Hedlund. *Topological dynamics*, volume 36. American Mathematical Soc., 1955.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

Juha Heinonen. *Lectures on analysis on metric spaces*. Springer Science & Business Media, 2001.

Stefan Horoi, Jessie Huang, Bastian Rieck, Guillaume Lajoie, Guy Wolf, and Smita Krishnaswamy. Exploring the geometry and topology of neural network loss landscapes. In *International Symposium on Intelligent Data Analysis*, pages 171–184. Springer, 2022.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

Dayal Singh Kalra and Maissam Barkeshli. Phase diagram of early training dynamics in deep neural networks: effect of the learning rate, depth, and width. *Advances in Neural Information Processing Systems*, 36:51621–51662, 2023.

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1032–1041. PMLR, 2019.

Kazimierz Kuratowski. *Topology: Volume I*, volume 1. Elsevier, 2014.

Serge Lang. *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media, 2012.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Clément Maria, Pawel Dlotko, Vincent Rouvreau, and Marc Glisse. Rips complex. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.11.0 edition, 2025. URL `https://gudhi.inria.fr/doc/3.11.0/group__rips__complex.html`.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *arXiv preprint arXiv:1902.06015*, 2019.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory*, volume 34. Springer Science & Business Media, 1994.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023.

Peter M Neumann. The structure of finitary permutation groups. *Archiv der Mathematik*, 27(1):3–17, 1976.

Emmy Noether. Invariante variationsprobleme. *Königlich Gesellschaft der Wissenschaften Göttingen Nachrichten Mathematik-physik Klasse*, 2:235–267, 1918.

Marco Nurisso, Pierrick Leroy, and Francesco Vaccarino. Topological obstruction to the training of shallow relu neural networks. *Advances in Neural Information Processing Systems*, 37:35358–35387, 2024.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL `https://arxiv.org/abs/2201.02177`.

Emilie Purvine, Davis Brown, Brett Jefferson, Cliff Joslyn, Brenda Praggastis, Archit Rathore, Madelyn Shapiro, Bei Wang, and Youjia Zhou. Experimental observations of the topology of convolutional neural network activations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9470–9479, 2023.

Samuel L Smith, Benoit Dherin, David GT Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. *arXiv preprint arXiv:2101.12176*, 2021.

Herbert Spohn. *Large scale dynamics of interacting particles*. Springer Science & Business Media, 2012.

Hong-Yi Wang, Di Luo, Tomaso Poggio, Isaac L. Chuang, and Liu Ziyin. A universal compression theory for lottery ticket hypothesis and neural scaling laws. In *International Conference on Learning Representations (ICLR)*, 2026.

Lei Wu, Chao Ma, et al. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31, 2018.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *arXiv preprint arXiv:1806.05393*, 2018.

Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

Liu Ziyin. Symmetry induces structure and constraint of learning. In *Forty-first International Conference on Machine Learning*, 2024.

Liu Ziyin, Yizhou Xu, and Isaac Chuang. Remove symmetries to control model expressivity and improve optimization. *ICLR*, 2025.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 347–356, 2004.

A Proofs of Theoretical Results

In this section, we prove all the theoretical results in the main paper. Before starting, we first define some additional notation. For a collection, we use subscripts to denote its elements. For example, if $\mathcal{X} = \{\boldsymbol{x}_i\}_{i\in I} \in \left(\mathbb{R}^D\right)^I$ is a collection of D-dimensional vectors, then $\mathcal{X}_i$ represents $\boldsymbol{x}_i$ by default.

For an operator P on I, and a subset $J \subseteq I$, we use P_J to denote the operator obtained by constraining P on J.

For a statement ψ, we use $\mathbb{1}_{\{\psi\}}$ to represent its indicator, i.e. $\mathbb{1}_{\{\psi\}} = \begin{cases} 1 & \psi \text{ is true} \\ 0 & \text{otherwise} \end{cases}$.

A.1 Proof of Lemma 4.1

Let $\mathcal{X} = \mathcal{X}^{(t)}$ for convenience. Define $P : I \to I$ as switching i and j:

$$\forall k \in I, P(k) = \begin{cases} j & k = i \\ i & k = j \\ k & \text{otherwise} \end{cases}. \tag{17}$$

Obviously $P \in \mathsf{FSym}(I)$. Since $\boldsymbol{x}_i^{(t)} = \boldsymbol{x}_j^{(t)}$, we have $\mathcal{X} = P\mathcal{X}$.

Applying $\mathcal{X} = P\mathcal{X}$ and the equivariance property, we can obtain that

$$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} + \eta U_i(\mathcal{X}) = \boldsymbol{x}_j^{(t)} + \eta U_i(P\mathcal{X}) = \boldsymbol{x}_j^{(t)} + \eta (PU(\mathcal{X}))_i = \boldsymbol{x}_j^{(t)} + \eta U_j(\mathcal{X}) = \boldsymbol{x}_j^{(t+1)}, \tag{18}$$

which proves the proposition.

A.2 Proof of Lemma 4.2

We first prove a lemma showing that two neurons that are close must remain close.

Lemma A.1 (No Splitting)**.** *The following statement holds when U has the equivariance property and K-continuity property. For any $t \in \mathbb{N}$ and $i, j \in I$ such that $i \neq j$, we have*

$$\left\|U_i(\mathcal{X}^{(t)}) - U_j(\mathcal{X}^{(t)})\right\| \le K \left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\right\|. \tag{19}$$

Proof. Let $\mathcal{X} = \mathcal{X}^{(t)}$ for convenience. Define $P \in \mathsf{FSym}(I)$ as switching i and j (as in eq. (17)). Then using the equivariance property, we have

$$\begin{aligned} U_i(\mathcal{X}) - U_j(\mathcal{X}) &= U_i(\mathcal{X}) - (PU(\mathcal{X}))_i && (20)\\ &= U_i(\mathcal{X}) - U_i(P\mathcal{X}). && (21) \end{aligned}$$

Notice that $\mathcal{X}$ and $P\mathcal{X}$ only differ in entries i and j. Using the K-continuity property, we have

$$\begin{aligned} \sqrt{2}\left\|U_i(\mathcal{X}) - U_j(\mathcal{X})\right\| &= \sqrt{\left\|U_i(\mathcal{X}) - U_j(\mathcal{X})\right\|^2 + \left\|U_i(\mathcal{X}) - U_j(\mathcal{X})\right\|^2} && (22)\\ &= \sqrt{\left\|U_i(\mathcal{X}) - U_i(P\mathcal{X})\right\|^2 + \left\|U_j(\mathcal{X}) - U_j(P\mathcal{X})\right\|^2} && (23)\\ &\le \left\|U(\mathcal{X}) - U(P\mathcal{X})\right\| && (24)\\ &\le K\left\|\mathcal{X} - P\mathcal{X}\right\| && (25)\\ &= \sqrt{2}K\left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\right\|. && (26) \end{aligned}$$

The proposition is thus proved by shifting the terms. □

Next, we prove Lemma 4.2 using Lemma A.1.

Proof of Lemma 4.2 Notice that

$$\begin{aligned} \left\|\boldsymbol{x}_i^{(t+1)} - \boldsymbol{x}_j^{(t+1)}\right\| &= \left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)} + \eta\left(U_i(\mathcal{X}^{(t)}) - U_j(\mathcal{X}^{(t)})\right)\right\| && (27)\\ &\in \left\|\boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)}\right\| + [-\eta, +\eta] \times \left\|U_i(\mathcal{X}^{(t)}) - U_j(\mathcal{X}^{(t)})\right\|. && (28) \end{aligned}$$

The proposition is thus directly proved by applying Lemma A.1. □

A.3 Proof of Lemma 4.3

For simplicity in the proof we fix a time t and denote $\widehat{U}^{(t)}$ by f. From Lemma 4.1, we have if $\boldsymbol{x}_i^{(t)} = \boldsymbol{x}_j^{(t)}$ then $\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_j^{(t+1)}$, therefore f is well-defined. Moreover, from the definition of $S^{(t)}$ and $S^{(t+1)}$, it is obvious that f is a surjection.

Now suppose $\eta K < 1$. If $\boldsymbol{x}_i^{(t)} \neq \boldsymbol{x}_j^{(t)}$, the left-hand-side of Lemma 4.2 and the condition that $\eta K < 1$ together shows that $\boldsymbol{x}_i^{(t+1)} \neq \boldsymbol{x}_j^{(t+1)}$, following from which we have f is an injection. Therefore, f is a bijection.

A.4 Proof of Theorem 4.4

We fix a time point t and denote $\widehat{U}^{(t)}$ by f. Lemma 4.3 has already proved that f is a surjection. In this proof, we first prove the topological properties (i., ii. and iii.), and then the differentiable manifold property (iv.).

Topological properties. For any pair of two different points $\boldsymbol{x}_i^{(t)}$ and $\boldsymbol{x}_j^{(t)}$, from the right-hand-side of Lemma 4.2, we have

$$\left\| f\left(\boldsymbol{x}_i^{(t)}\right) - f\left(\boldsymbol{x}_j^{(t)}\right) \right\| \leq (1+\eta K) \left\| \boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)} \right\|, \tag{29}$$

which shows that f is $(1+\eta K)$-Lipschitz continuous. Since all Lipschitz continuous functions are continuous, we have f is also continuous.

If, additionally, $S^{(t)}$ is compact, then from the continuity of f we immediately know $S^{(t+1)} = f\left(S^{(t)}\right)$ is compact. Moreover, since $S^{(t+1)}$ is a metric space, it is automatically Hausdorff, and it is known that a surjective mapping from a compact space to a Hausdorff space is a quotient map.

Now, suppose $\eta K < 1$ (without the compactness of $S^{(t)}$). Lemma 4.3 has proved that f is a bijection. Consider the inversion of f. Let $g = f^{-1}$. It is obvious that for any $i \in I$, we have $g\left(\boldsymbol{x}_i^{(t+1)}\right) = \boldsymbol{x}_i^{(t)}$. Using the left-hand-side of Lemma 4.2, we have

$$\left\| g\left(\boldsymbol{x}_i^{(t+1)}\right) - g\left(\boldsymbol{x}_j^{(t+1)}\right) \right\| \leq \frac{1}{1-\eta K} \left\| \boldsymbol{x}_i^{(t+1)} - \boldsymbol{x}_j^{(t+1)} \right\| \tag{30}$$

for any $i, j \in I$, and therefore $g = f^{-1}$ is also continuous. This proves that f is a homeomorphism.

Differentiable manifold properties. Now, with the condition that $\eta K < 1$ and $U^{(t)}$ satisfies the smoothness property (P3), we prove that f is a diffeomorphism from $S^{(t)}$ to $S^{(t+1)}$. The Invariance of Domain Theorem (See e.g. Theorem 2B.3 in Hatcher [2002]) guarantees that $S^{(t+1)}$ is also an open set in $\mathbb{R}^D$. Therefore, we only need to prove that f and its inverse both have continuous derivatives.

Fix a point $\boldsymbol{x}_i^{(t)} \in S^{(t)}$. Since $S^{(t)}$ is open, there must be a scalar $r_i > 0$, such that for any $\boldsymbol{\Delta} \in \mathbb{R}^d$ with $\|\boldsymbol{\Delta}\| \leq r_i$, we have $\boldsymbol{x}_i^{(t)} + \boldsymbol{\Delta} \in S^{(t)}$. Consider such a perturbation $\boldsymbol{\Delta}$, then there must be a $j \in I$ such that $\boldsymbol{x}_j^{(t)} = \boldsymbol{x}_i^{(t)} + \boldsymbol{\Delta}$. Let $P \in \mathsf{FSym}(I)$ be the permutation operator that exchanges i and j (as defined in eq. (17)). We have

$$\begin{aligned}
f(\boldsymbol{x}_i^{(t)} + \boldsymbol{\Delta}) &= f\left(\boldsymbol{x}_j^{(t)}\right) && (31)\\
&= U_j\left(\mathcal{X}^{(t)}\right) && (32)\\
&= \left(PU\left(\mathcal{X}^{(t)}\right)\right)_i && (33)\\
&= U_i\left(P\mathcal{X}^{(t)}\right) && \text{(Equivariance Property)} \quad (34)\\
&= U_i\left[\mathcal{X}^{(t)} + (\boldsymbol{e}_i - \boldsymbol{e}_j)\left(\boldsymbol{x}_j^{(t)} - \boldsymbol{x}_i^{(t)}\right)\right] && (35)\\
&= g_i^{(t)}\left(\boldsymbol{\Delta}, \boldsymbol{x}_j^{(t)}\right) && (36)\\
&= g_i^{(t)}\left(\boldsymbol{\Delta}, \boldsymbol{x}_i^{(t)} + \boldsymbol{\Delta}\right) && (37)
\end{aligned}$$

Since from P3 we know $g_i^{(t)}$ is C^1 with respect to its two parameters, from the chain rule we have $g_i^{(t)}\left(\boldsymbol{\Delta}, \boldsymbol{x}_i^{(t)} + \boldsymbol{\Delta}\right)$ is also C^1 with respect to $\boldsymbol{\Delta}$, and therefore f is also C^1 at point $\boldsymbol{x}_i^{(i)}$. Since i is arbitrarily chosen, f is thus C^1 on entire $S^{(t)}$.

Next, we prove that f^{-1} is also C^1. Again consider $\boldsymbol{x}_i^{(t)} \in S^{(t)}$. Since we already know f is C^1, let its gradient at point $\boldsymbol{x}_i^{(t)}$ be $\boldsymbol{G}$ and we have for any unit vector $\boldsymbol{v}$, the directional derivative satisfies

$$\boldsymbol{G}\boldsymbol{v} = \lim_{\substack{\delta \to 0 \\ \delta \neq 0}} \frac{f\left(\boldsymbol{x}_i^{(t)} + \delta \boldsymbol{v}\right) - f\left(\boldsymbol{x}_i^{(t)}\right)}{\delta}. \tag{38}$$

Let $\alpha = 1 - \eta K, \beta = 1 + \eta K$. From Lemma 4.2, for any $\delta < r_i$ we have

$$\alpha \leq \frac{\left\| f\left(\boldsymbol{x}_i^{(t)} + \delta \boldsymbol{v}\right) - f\left(\boldsymbol{x}_i^{(t)}\right) \right\|}{|\delta|} \leq \beta. \tag{39}$$

Subtracting the bounds into eq. (38), we get

$$\|\boldsymbol{G}\boldsymbol{v}\| \in [\alpha, \beta], \tag{40}$$

which further implies that all singular-values of $\boldsymbol{G}$ are in $[\alpha, \beta]$, which means $\boldsymbol{G}$ is invertible. Since $\widehat{U}$ is C^1, inverse function theorem therefore shows f^{-1} is also C^1.

A.5 Proof of Theorem B.1

We fix a time point t and denote $\widehat{U}^{(t)}$ by f. Lemma 4.3 has already proved that f is a bijection. For any open set $A \subseteq S^{(t+1)}$, we have

$$\begin{aligned}
\mu^{(t+1)}(A) &= \mu^{(t+1)}\left\{ \boldsymbol{x}_i^{(t+1)} \middle| i \in I, \boldsymbol{x}_i^{(t+1)} \in A \right\} && (41)\\
&= m\left\{ i \in I \,|\boldsymbol{x}_i^{(t+1)} \in A \right\} && (42)\\
&= m\left\{ i \in \left| f\left(\boldsymbol{x}_i^{(t)}\right) \in A \right\} \right. && (43)\\
&= m\left\{ i \in \left| \boldsymbol{x}_i^{(t)} \in f^{-1}(A) \right\} \right. && (44)\\
&= \mu^{(t)}(f^{-1}(A)). && (45)
\end{aligned}$$

This proves that f is measure-preserving. Following the same process one can easily prove that f^{-1} is also measure-preserving.

A.6 Proof of Proposition 5.1

In this proof we prove a slightly stronger version of the proposition originally stated in Proposition 5.1, without using the condition that I is finite. The result for finite I is thus a direct corollary.

We only need to prove that for any $\mathcal{X} \in \left(\mathbb{R}^D\right)^I$ and $P \in \mathsf{FSym}(I)$, we have

$$P\nabla L(\mathcal{X}) = \nabla L(P\mathcal{X}). \tag{46}$$

Now we fix $P \in \mathsf{FSym}(I)$ and consider any $\mathcal{X} \in \left(\mathbb{R}^D\right)^I$. Let

$$J = \{i \in I | Pi \neq i\}, \tag{47}$$

be the support set of P. Since P is finitary, J is a finite set. Therefore, we only need to prove the proposition of entries in J. define $L_J : \left(\mathbb{R}^D\right)^J \to \mathbb{R}$ such that

$$\forall \mathcal{Y} = \{\boldsymbol{y}_j\}_{j \in J}, L_J(\mathcal{Y}) = L\left(\left\{\mathbb{1}_{\{i \in J\}}\boldsymbol{y}_i + \mathbb{1}_{\{i \notin J\}}\boldsymbol{x}_i\right\}_{i \in I}\right). \tag{48}$$

The symmetry gives us

$$\forall \mathcal{Y} \in \left(\mathbb{R}^D\right)^J, L_J(\mathcal{Y}) = L_J(P|_J\mathcal{Y}), \tag{49}$$

where $P_J = P|_J$ is restriction of P on J. Taking derivative of both sides gives

$$\nabla L_J(\mathcal{Y}) = P_J^\intercal \nabla L_J(P_J\mathcal{Y}), \tag{50}$$

shifting the terms and eq. (46) is proved.

A.7 Proof of Proposition 5.2

The proposition is directly proved by noticing that

$$\left\|U^{(t)}(\mathcal{X}) - U^{(t)}(\mathcal{Y})\right\| = \|\nabla L(\mathcal{Y}) - \nabla L(\mathcal{X})\| \le K\|\mathcal{Y} - \mathcal{X}\|. \tag{51}$$

A.8 Proof of Proposition 5.3

We use eq. (46) proved before. Let $P \in \mathsf{FSym}(I)$. We have

$$U^{(t)}\left[\left\{\begin{pmatrix}\boldsymbol{\theta}_{P(i)}\\ \boldsymbol{m}_{P(i)}\\ \boldsymbol{v}_{P(i)}\end{pmatrix}\right\}_{i\in I}\right] = \left\{\begin{pmatrix} -\frac{\boldsymbol{m}_{P(i)}/(1-\beta_1^t)}{\epsilon+\sqrt{P\boldsymbol{v}_{P(i)}/(1-\beta_2^t)}}\\ \frac{1-\beta_1}{\eta}\left[\nabla_i L(P\Theta) - \boldsymbol{m}_{P(i)}\right]\\ \frac{1-\beta_2}{\eta}\left[\nabla_i L(P\Theta)^2 - \boldsymbol{v}_{P(i)}\right]\end{pmatrix}\right\}_{i\in I} \tag{52}$$

$$= \left\{\begin{pmatrix} -\frac{\boldsymbol{m}_{P(i)}/(1-\beta_1^t)}{\epsilon+\sqrt{P\boldsymbol{v}_{P(i)}/(1-\beta_2^t)}}\\ \frac{1-\beta_1}{\eta}\left[\nabla_{P(i)} L(\Theta) - \boldsymbol{m}_{P(i)}\right]\\ \frac{1-\beta_2}{\eta}\left[\nabla_{P(i)} L(\Theta)^2 - \boldsymbol{v}_{P(i)}\right]\end{pmatrix}\right\}_{i\in I} \tag{53}$$

$$= P\left\{\begin{pmatrix} -\frac{\boldsymbol{m}_i/(1-\beta_1^t)}{\epsilon+\sqrt{P\boldsymbol{v}_i/(1-\beta_2^t)}}\\ \frac{1-\beta_1}{\eta}\left[\nabla_i L(\Theta) - \boldsymbol{m}_i\right]\\ \frac{1-\beta_2}{\eta}\left[\nabla_i L(\Theta)^2 - \boldsymbol{v}_i\right]\end{pmatrix}\right\}_{i\in I} \tag{54}$$

$$= PU^{(t)}\left[\left\{\begin{pmatrix}\boldsymbol{\theta}_i\\ \boldsymbol{m}_i\\ \boldsymbol{v}_i\end{pmatrix}\right\}_{i\in I}\right]. \tag{55}$$

A.9 Proof of Proposition 5.4

Since in Equation (14) scalar operations applied to vectors are applied element-wise, and here we only concern the continuous property of the learning algorithm instead of its internal structure, WLOG we can vectorize the parameters and states, denoting them as MD-dimensional vectors. That is, below we assume $\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v} \in \mathbb{R}^N$, where $N = MD$.

For $\boldsymbol{u} = U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})$, let $\boldsymbol{u}_\theta, \boldsymbol{u}_m, \boldsymbol{u}_v \in \mathbb{R}^N$ be the $\boldsymbol{\theta}$-, $\boldsymbol{m}$-, and $\boldsymbol{v}$- component of $\boldsymbol{u}$ respectively.

Let $\gamma_1 = \frac{1-\beta_1}{\eta} > 0$, and $\gamma_2 = \frac{1-\beta_2}{\eta} > 0$. Since B is a bounded set, let $R > 0$ be the radius of B. For notation simplicity, let $g(\boldsymbol{\theta}) = \nabla L(\boldsymbol{\theta})$.

We first prove that, the gradient $g(\boldsymbol{\theta})$ is bounded. Since $\|\boldsymbol{\theta}\| \le R$ and g is Lipschitz-continuous, we have

$$\|g(\boldsymbol{\theta})\| \le \|g(\boldsymbol{\theta}) - g(\mathbf{0})\| + \|g(\mathbf{0})\| \le K\|\boldsymbol{\theta}\| + \|g(\mathbf{0})\| \le (K+1)R. \tag{56}$$

In the following, let $G = (K+1)R$ be the upper bound of $\|g(\boldsymbol{\theta})\|$.

Next, we prove the Lipschitz continuity of U w.r.t. each of the three components.

i) *Lipschitz continuity of the m-component of U.*
We have

$$\|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})_m - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')_m\| = \gamma_1 \|g(\boldsymbol{\theta}) - \boldsymbol{m} - g(\boldsymbol{\theta}') - \boldsymbol{m}'\| \tag{57}$$

$$\le \gamma_1\|g(\boldsymbol{\theta}) - g(\boldsymbol{\theta}')\| + \gamma_1\|\boldsymbol{m} - \boldsymbol{m}'\| \tag{58}$$

$$\le \gamma_1 K\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| + \gamma_1\|\boldsymbol{m} - \boldsymbol{m}'\| \tag{59}$$

$$\le \sqrt{3}\gamma_1 \max\{K, 1\}\|(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - (\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\| \tag{60}$$

Therefore, U is $\sqrt{3}\gamma_1 \max\{K, 1\}$-Lipschitz continuous w.r.t. $\boldsymbol{m}$.

ii) *Lipschitz continuity of the v-component of U.*

Similar to the proof for the m-component. Only need to notice that in a bounded area B, the mapping $\boldsymbol{\theta} \mapsto g(\boldsymbol{\theta})^2$ is also Lipschitz-continuous since

$$\|g(\boldsymbol{\theta})^2 - g(\boldsymbol{\theta}')^2\| = \sum_{i=1}^{m} g(\boldsymbol{\theta})_i^2 - g(\boldsymbol{\theta}')_i^2 \tag{61}$$

$$= \sum_{i=1}^{m} (g(\boldsymbol{\theta})_i + g(\boldsymbol{\theta}')_i)(g(\boldsymbol{\theta})_i - g(\boldsymbol{\theta}')_i) \tag{62}$$

$$\leq \|g(\boldsymbol{\theta}) + g(\boldsymbol{\theta}')\| \|g(\boldsymbol{\theta}) - g(\boldsymbol{\theta}')\| \tag{63}$$

$$\leq 2GK\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|. \tag{64}$$

Then, repeating the proof for the m-component, we obtain a Lipschitz constant $\sqrt{3}\gamma_2 \max\{2GK, 1\}$.

iii) *Lipschitz continuity of of the $\boldsymbol{\theta}$-component of U.*

Notice that for $a, b, c, d \in \mathbb{R}$ where $a, c \neq 0$, we have

$$\frac{b}{a} - \frac{d}{c} = \frac{b-d}{a} + d\left(\frac{1}{a} - \frac{1}{c}\right). \tag{65}$$

Let $\psi_1 = (1-\beta_1)^t$ and $\psi_2 = (1-\beta_2)^t$, we have

$$\|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})_\theta - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')_\theta\| = \left\| \frac{\boldsymbol{m}\psi_1}{\epsilon + \sqrt{\boldsymbol{v}/\psi_2}} - \frac{\boldsymbol{m}'\psi_1}{\epsilon + \sqrt{\boldsymbol{v}'/\psi_2}} \right\| \tag{66}$$

$$= \left\| \frac{\psi_1(\boldsymbol{m} - \boldsymbol{m}')}{\epsilon + \sqrt{\boldsymbol{v}/\psi_2}} + \boldsymbol{m}'\psi_1 \left(\frac{1}{\epsilon + \sqrt{\boldsymbol{v}/\psi_2}} - \frac{1}{\epsilon + \sqrt{\boldsymbol{v}'/\psi_2}} \right) \right\| \tag{67}$$

$$\leq \left\| \frac{\psi_1(\boldsymbol{m} - \boldsymbol{m}')}{\epsilon + \sqrt{\boldsymbol{v}/\psi_2}} \right\| + \|\boldsymbol{m}'\|\psi_1 \left\| \frac{1}{\epsilon + \sqrt{\boldsymbol{v}/\psi_2}} - \frac{1}{\epsilon + \sqrt{\boldsymbol{v}'/\psi_2}} \right\| \tag{68}$$

$$\leq \frac{\psi_1}{\epsilon}\|\boldsymbol{m} - \boldsymbol{m}'\| + R\psi_1 \left\| \frac{\sqrt{\boldsymbol{v}'/\psi_2} - \sqrt{\boldsymbol{v}/\psi_2}}{\left(\epsilon + \sqrt{\boldsymbol{v}/\psi_2}\right)\left(\epsilon + \sqrt{\boldsymbol{v}'/\psi_2}\right)} \right\| \tag{69}$$

$$\leq \frac{\psi_1}{\epsilon}\|\boldsymbol{m} - \boldsymbol{m}'\| + \frac{R\psi_1}{\epsilon^2\psi_2}\|\sqrt{\boldsymbol{v}} - \sqrt{\boldsymbol{v}'}\| \tag{70}$$

Notice that

$$\|\sqrt{\boldsymbol{v}} - \sqrt{\boldsymbol{v}'}\| = \left\| \frac{\boldsymbol{v} - \boldsymbol{v}'}{\sqrt{\boldsymbol{v}} + \sqrt{\boldsymbol{v}'}} \right\| \leq \frac{\|\boldsymbol{v} - \boldsymbol{v}'\|}{2\sqrt{\epsilon}}. \tag{71}$$

Therefore, we have

$$\|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})_\theta - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')_\theta\| \leq \frac{\psi_1}{\epsilon}\|\boldsymbol{m} - \boldsymbol{m}'\| + \frac{R\psi_1}{2\epsilon^2\sqrt{\epsilon}\psi_2}\|v - v'\| \tag{72}$$

$$\leq \frac{\sqrt{3}\psi_1}{\epsilon} \max\left\{1, \frac{R}{2\epsilon\sqrt{\epsilon}\psi_2}\right\} \|(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - (\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\|. \tag{73}$$

Thus, U is $\left(\frac{\sqrt{3}\psi_1}{\epsilon} \max\left\{1, \frac{R}{2\epsilon\sqrt{\epsilon}\psi_2}\right\}\right)$-Lipschitz for its $\boldsymbol{\theta}$-component.

Now we have proved that U is Lipschitz continuous w.r.t. each of the components. Let their Lipschitz constant be K_1, K_2, K_3 respectively. Let $K' = \sqrt{K_1^2 + K_2^2 + K_3^2}$, we have

$$\begin{aligned}
&\|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\| \\
=&\sqrt{\|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})_\theta - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')_\theta\|^2 + \|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})_m - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')_m\|^2 + \|U(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v})_v - U(\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')_v\|^2} \\
\leq&\sqrt{K_1^2 + K_2^2 + K_3^2}\|(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - (\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\| \\
=&K'\|(\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{v}) - (\boldsymbol{\theta}', \boldsymbol{m}', \boldsymbol{v}')\|.
\end{aligned}$$

This proves the proposition.

A.10 Proof of Corollary 4.6

Let $\delta = \inf_{\substack{x,y\in S^{(t)}\\ x\neq y}}$. When $\delta = 0$, the result is a direct corollary of Theorem 4.4. Below we assume $\delta > 0$.

Since $2r < \delta$, $\left(S^{(t)}\right)^r$ is a union of pairwise disjoint open balls with radius r. Using Lemma 4.2, the minimal distance at time $t+1$ is at least $(1-\eta K)\delta > 2(1-\eta K)r$, so $\left(S^{(t+1)}\right)^{(1-\eta K)r}$ is a union of pairwise disjoint open balls with radius $(1-\eta K)r$. Since any two open balls are homeomorphic, we have $\left(S^{(t)}\right)^r$ is homeomorphic to $\left(S^{(t+1)}\right)^{(1-\eta K)r}$.

A.11 Proof of Corollary 4.8

Suppose at time t, $S^{(t)}$ is covered by P^ϵ, where $P = \{\boldsymbol{p}_k\}_{k=1}^m$, and $m = E_\epsilon(t)$. Let I_k be the indices of neurons that are covered by $\{\boldsymbol{p}_k\}^\epsilon$. We only need to prove that the diameter of the set of neurons in I_k is at most $1/2$ at time $t+T$, so that they can be covered by a ball with diameter 1.

If $|I_k| = 1$, then it is obvious that the diameter of the set of neurons in I_k is less than $1/2$. Now, suppose $|I_k| \geq 2$, consider two indices $i, j \in I_k$, with $i \neq j$. Using Lemma 4.2, we have

$$\begin{aligned}
\left\|\boldsymbol{x}_i^{(t+T)} - x_j^{(t+T)}\right\| &\leq (1+\eta K)^T \left\|\boldsymbol{x}_i^{(t)} - x_j^{(t)}\right\| && (74)\\
&\leq (1+\eta K)^T \epsilon && (75)\\
&\leq \exp(T\log(1+\eta K) - T\eta K)/2 && (76)\\
&< \exp(0)/2 && (77)\\
&= 1/2. && (78)
\end{aligned}$$

This proves the proposition.

A.12 Proof of Lemma 4.9

The function h is defined such that it maps $\mathcal{X}^{(t)}$ to the set P such that $|P| = m$ and P^ϵ covers $\mathcal{S}^{(t)}$. Let $P = \{\boldsymbol{p}_k\}_{k=1}^m$. Define $\pi_i \in [m]$ to be the index of ball where the i-th neuron is covered by, i.e. $\boldsymbol{x}_i^{(t)} \in \{\boldsymbol{p}_{\pi_i}\}^\epsilon$.

Define function $\psi : (\mathbb{R}^D)^m \to (\mathbb{R}^D)^I$, such that

$$\psi\left(\{\boldsymbol{q}_k\}_{k=1}^m\right) = \{\boldsymbol{q}_{\pi_i}\}_{i\in I}. \tag{79}$$

Let $g = f \circ \psi$. Now we upper bound $\left|(g\circ h)\left(\mathcal{X}^{(t)}\right) - f\left(\mathcal{X}^{(t)}\right)\right|$. We have

$$\begin{aligned}
\left|(g\circ h)\left(\mathcal{X}^{(t)}\right) - f\left(\mathcal{X}^{(t)}\right)\right| &= \left|f\left(\psi\circ h\left(\mathcal{X}^{(t)}\right)\right) - f\left(\mathcal{X}^{(t)}\right)\right| && (80)\\
&\leq R\left\|\psi\circ h\left(\mathcal{X}^{(t)}\right) - \mathcal{X}^{(t)}\right\| && (81)\\
&= R\sqrt{\sum_{i\in I}\|\boldsymbol{q}_{\pi_i} - \boldsymbol{x}_i\|^2} && (82)\\
&\leq R\epsilon|I|, && (83)
\end{aligned}$$

which proves the proposition.

B Extra Theoretical Results

In this section, we introduce two extensions of the theoretical result presented in the main text.

B.1 Measure Invariance

Beyond the invariance of topology, one can also ask "how many" neurons are stacked at a single point of $S^{(t)}$ and how their density evolves over time. Formally speaking, this corresponds to studying

the probability distribution on $S^{(t)}$ obtained by pushing forward a universal probability distribution defined on the index set I. In this subsection, we show that this distribution is also preserved under the update rule.

Formally, in this subsection we assume a concrete structure on the index set I. Assume there is a σ-algebra $\mathcal{F}$ on I and a probability measure $m : \mathcal{F} \to [0, 1]$. At any time $t \in \mathbb{N}$, define the mapping $r^{(t)} : i \mapsto \boldsymbol{x}_i^{(t)}$, and let it be a measurable function from I to $S^{(t)}$, where $S^{(t)}$ carries the corresponding Borel σ-algebra.[6] For each time t, we define a measure $\mu^{(t)}$ on $S^{(t)}$ as the push-forward of m under $r^{(t)}$, i.e.

$$\forall \text{open set } A \text{ on } S^{(t)},\ \mu^{(t)}(A) = m\left({r^{(t)}}^{-1}(A)\right). \tag{84}$$

Clearly, $\mu^{(t)}$ is also a probability measure.

Theorem B.1. *Suppose $U^{(t)}$ satisfies P1 and P2-K, and suppose $\eta K < 1$, then $\widehat{U}^{(t)}$ is a probability isomorphism between $\left(S^{(t)}, \mu^{(t)}\right)$ and $\left(S^{(t+1)}, \mu^{(t+1)}\right)$, i.e. $\widehat{U}^{(t)}$ and ${\widehat{U}^{(t)}}^{-1}$ are both measure-preserving bijections.*

Theorem B.1 shows that the update rule preserves not only the topology of $S^{(t)}$, but also the density of neurons across it. Theorem 4.4 and Theorem B.1 might remind readers of the topological dynamical systems and measure-preserving dynamical systems [Gottschalk and Hedlund, 1955]. However, what we study is more general because in this context the update rule $\widehat{U}^{(t)}$ as well as the space $S^{(t)}$ itself are both time-dependent, which violates the definition of the topological/measure-preserving dynamical systems. Therefore, the classical recurrence theorems in those fields cannot be directly applied.

B.2 A Weaker Version of Continuity Property

In Section 3, we mentioned that the specific form of K-continuity property is only for establishing a correspondence with the smoothness property used in optimization theory, and in our theory this property can actually be weaker. Specifically, when I is an infinite set, the K-continuity property defined in Section 3 implicitly requires that $U(\mathcal{X})$ and $U(\mathcal{Y})$ only differ in finite number of entries, which might not always hold. This condition can be relaxed to the upper bound on each entry of $U(\mathcal{X}) - U(\mathcal{Y})$.

(P2′) Altered K-Continuity Property: If there exists a constant $K > 0$, such that for any $t \in \mathbb{N}$, $\mathcal{X}, \mathcal{Y} \in \left(\mathbb{R}^D\right)^I$ and $i \in I$, we have

$$\sup_{i \in I} \left\| U_i^{(t)}(\mathcal{X}) - U_i^{(t)}(\mathcal{Y}) \right\| \le \frac{K}{2} \left\| \mathcal{X} - \mathcal{Y} \right\|, \tag{85}$$

then we say $U^{(t)}$ has altered K-continuity property.

The proofs of the main theorems are built upon Lemma A.1. Here we prove a variation of Lemma A.1 with the altered K-continuity property.

Lemma B.2 (Altered - No Splitting)**.** *The following statement holds when U has the equivariance property and altered K-continuity property. For any $t \in \mathbb{N}$ and $i, j \in I$ such that $i \neq j$, we have*

$$\left\| U_i(\mathcal{X}^{(t)}) - U_j(\mathcal{X}^{(t)}) \right\| \le K \left\| \boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)} \right\|. \tag{86}$$

Proof. Let $\mathcal{X} = \mathcal{X}^{(t)}$ for convenience. Define $P \in \mathsf{FSym}(I)$ as switching i and j (as in eq. (17)). Then using the equivariance property, we have

$$U_i(\mathcal{X}) - U_j(\mathcal{X}) = U_i(\mathcal{X}) - \left(PU(\mathcal{X})\right)_i = U_i(\mathcal{X}) - U_i(P\mathcal{X}). \tag{87}$$

Notice that $\mathcal{X}$ and $P\mathcal{X}$ only differ in entries i and j. Using the K-continuity property, we have

$$\begin{aligned} \left\| U_i(\mathcal{X}) - U_j(\mathcal{X}) \right\| &= \left\| U_i(\mathcal{X}) - U_i(P\mathcal{X}) \right\| && (88)\\ &\le \frac{K}{2} \left\| \mathcal{X} - P\mathcal{X} \right\| && (89)\\ &\le K \left\| \boldsymbol{x}_i^{(t)} - \boldsymbol{x}_j^{(t)} \right\|. && (90) \end{aligned}$$

[6] These assumptions are automatically satisfied with a finite I.

□

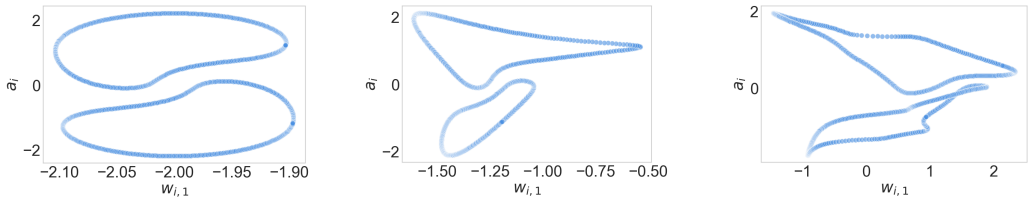
In all the presented theories, the K-continuity property can be replaced by the altered K-continuity property. The proofs directly apply by replacing Lemma A.1 with Lemma B.2.

C Extra Experiment Results

In this section, we provide extra experiment results performed with more optimizers, complementing those in Section 6.

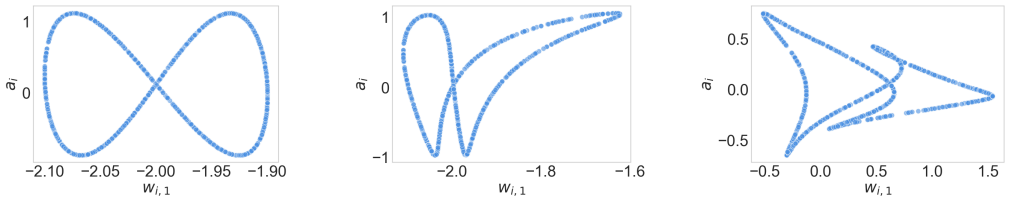
C.1 Low Dimensional Neural Networks

We extend the low-dimensional experiments to additional optimizers. Figure 4 presents addition results for the 2D network trained with GD, under a different initialization. Figures 8 and 9 present the results for 2D and 3D networks trained with Adam, and Figures 5 and 7 present the corresponding results with momentum gradient descent.



(a) Initialization (b) End of training, small η (c) End of training, large η

Figure 4: **Topology of a 2D neural network with GD and disjoint genus-1 initialization.** The neurons are initialized on the disjoint union of two genus-1 surfaces and optimized with GD.



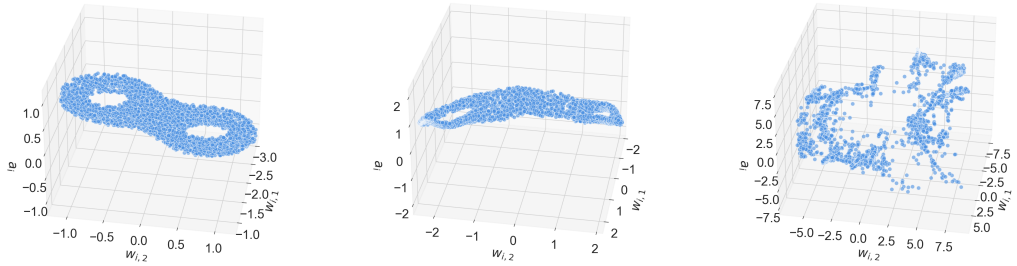
(a) (b) (c)

Figure 5: **Topology of a 2D neural network with momentum gradient descent.** The neurons are initialized on a genus-2 surface and optimized with momentum GD.

C.2 Experiments on Realistic Models and Tasks

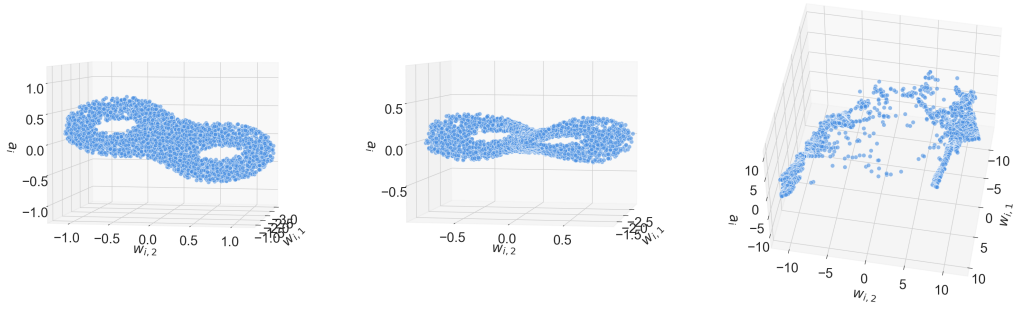
In order to verify the theory on more realistic settings, we train a two-layer MLP on the MNIST dataset [LeCun, 1998] for a classification task using standard cross-entropy loss, and track the evolution of Betti numbers. The network is initialized with neurons uniformly sampled from the surface of a 3D unit ball, which has Betti numbers $(b_1, b_2) = (0, 1)$. Figures 10 and 11 shows the results with both GD and Adam, respectively. These results are consistent with our theoretical predictions: with small learning rates, the model learns without changing the topological structure of the neurons, while with large learning rates, the topology can change. Importantly, in all cases the model can achieve a significant test accuracy, ruling out the possibility that with small step sizes the model simply stays near initialization without meaningful updates.

Notice that in the small step-size setting of Figure 11, the topology remains unchanged initially but begins to change after a certain period of training. We attribute this to a key difference in the dynamics of Adam under small versus large learning rates. Specifically, with small learning rates, the



(a) Initialization (b) End of training, small η (c) End of training, large η

Figure 6: **Topology of a 3D neural network with GD.** The neurons are initialized on a genus-2 surface and optimized with GD.



(a) Initialization (b) End of training, small η (c) End of training, large η

Figure 7: **Topology of a 3D neural network with momentum gradient descent.** The neurons are initialized on a genus-2 surface and optimized with momentum GD. The camera angle is manually adjusted to better visualize the structure of the point cloud.

network undergoes progressive sharpening: the sharpness steadily increases and eventually surpasses the topological critical point. Beyond this point, the step size becomes relatively "large," and the topology of the neurons starts to change. In contrast, with large learning rates, the sharpness remains small. Similar phenomena have also been reported in the literature [Kalra and Barkeshli, 2023].

To verify this explanation, Figure 12 shows the evolution of the sharpness inversion ($1/K$, where K denotes the largest eigenvalue of the Hessian matrix[7]) under Adam. Comparing these results with those in Figure 11, it is evident that the topology begins to change once $1/K$ becomes sufficiently small, matching our theoretical prediction.

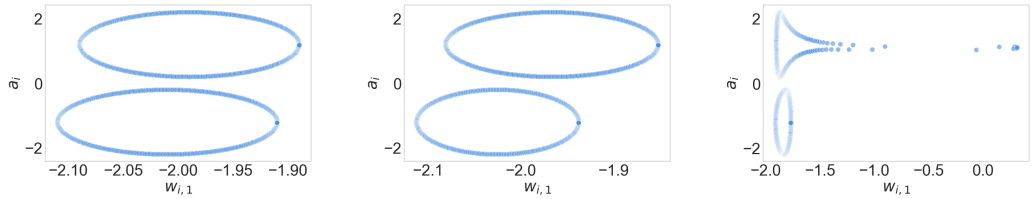
C.3 Extra Results with Multi-Layer Neural Networks

In order to further verify our theoretical results, we also conduct experiments with multi-layer neural networks. Specifically, we use a four-layer neural network, where the outputs of the first and third layers are wide (corresponding to the number of neurons), and the output of the second layer is 1-d (corresponding to the dimensionality of neurons). In this network, the weights in the first and second layer constitute a set of neurons, and so do the weights in the third and fourth layer, where the dimensionality of each neuron is 2. In Figure 13, we plot the initialization and end of training neuron distribution for both layers. Other than the model, the rest of the settings are the same as those in Appendix D.1. The results clearly support our theoretical claims.

D Experiment Details

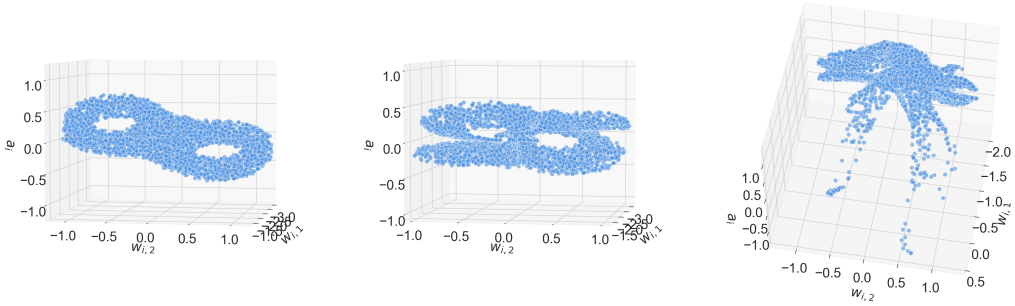
In this section, we provide the experimental details.

[7]The sharpness is calculated with the hessian-eigenthings library [Golmant et al., 2018].



(a) (b) (c)

Figure 8: **Topology of a 2D neural network with Adam.** The neurons are initialized on the disjoint union of two genus-1 surfaces and optimized with Adam.



(a) (b) (c)

Figure 9: **Topology of a 3D neural network with Adam.** The neurons are initialized on a genus-2 surface and optimized with Adam. The camera angle is manually adjusted to better visualize the structure of the point cloud.

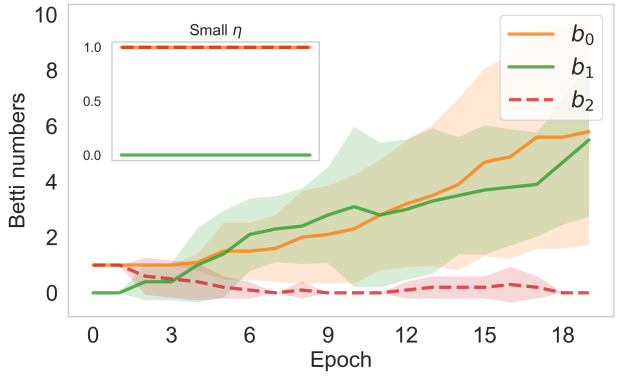


Figure 10: **Evolution of Betti numbers during training with GD.** The main panel shows results for the large learning rate, while the inset shows results for the small one. Each curve is obtained by averaging over 10 runs; the shaded regions indicate the standard deviations.

D.1 Experiments with Low-dimensional Neural Networks

As described in Section 6, we use a two-layer neural network with sigmoid activation, with input dimension $d = 1$ (referred to as the 2D case) or $d = 2$ (referred to as the 3D case).

Given input dimension d, the input data are denoted by $\mathcal{D} = \{(\boldsymbol{z}_s, y_s^*)\}_{s=1}^n \in \left(\mathbb{R}^d \times \mathbb{R}\right)^n$, where n is the dataset size. Each input $\boldsymbol{z}_s$ is sampled from a Gaussian distribution with variance 4, i.e., $\boldsymbol{z}_{s,j} \sim \mathcal{N}(0, 4)$ for $j \in \{1, 2\}$. The labels y_s^* are generated by a teacher model

$$y_s^* = \langle \boldsymbol{a}^*, \sigma\left(\boldsymbol{W}^* \boldsymbol{z}_s\right) \rangle, \tag{91}$$

where $\boldsymbol{a}^* \in \mathbb{R}^{h^*}$, $\boldsymbol{W}^* \in \mathbb{R}^{h^* \times d}$, and h^* is the hidden size of the teacher model. Both $\boldsymbol{a}^*$ and $\boldsymbol{W}^*$ are randomly sampled at the beginning and fixed when constructing the dataset, with $a_j^* \sim \mathcal{N}(0, 1)$ and

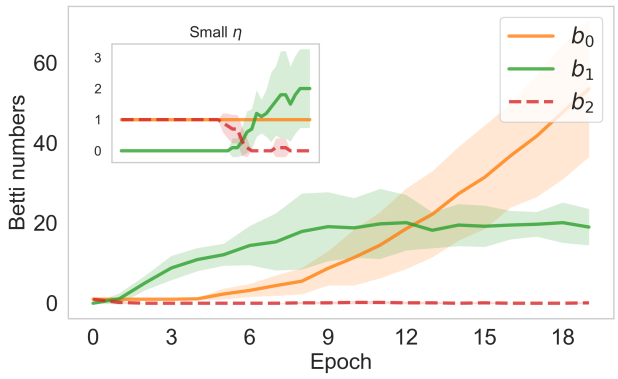
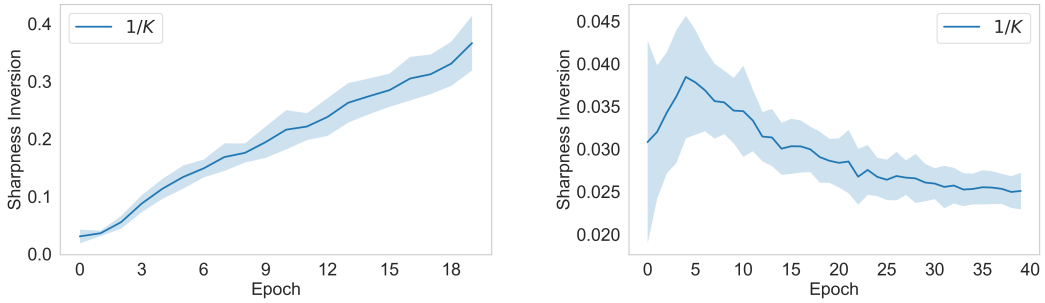


Figure 11: **Evolution of Betti numbers during training with Adam.** The plots show the Betti numbers b_1, and b_2 over time. The main panels correspond to large learning rates, while the insets show the results for small learning rates. Each curve is obtained by averaging over 10 runs with different random seeds; the curves denote the means and the shaded regions indicate the standard deviations. When the step size is small, the topology eventually changes after a certain training time. We attribute this to increasing sharpness: as training progresses, the network becomes sharper and the threshold for topological changes correspondingly decreases.



(a) Small step size

(b) Large step size

Figure 12: **Evolution of Betti numbers and sharpness inversion under Adam.** Here K denotes the largest eigenvalue of the Hessian matrix. The small step-size setting is trained for a longer time to ensure convergence.

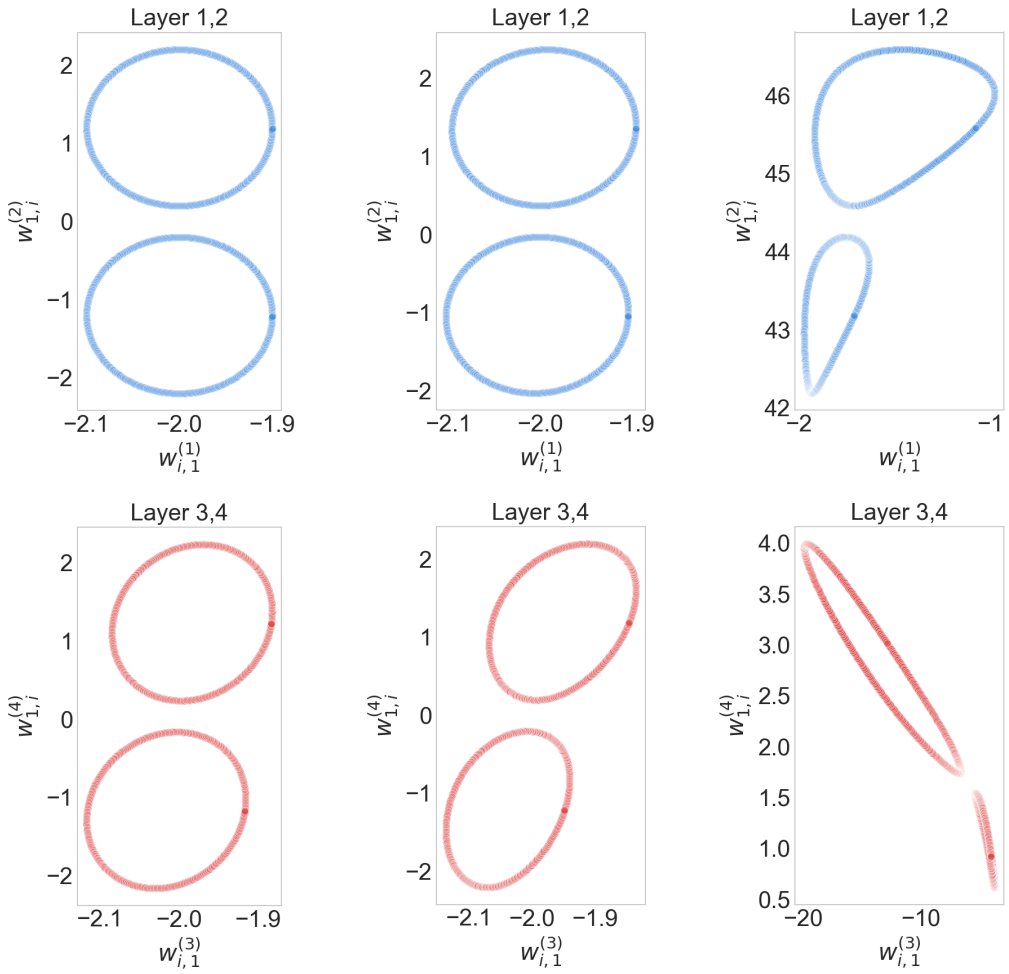
$w^*_{j,k} \sim \mathcal{N}(0, 0.36)$. In all experiments, n is set to 5000, with 70% of the data used for training. The model is trained using mini-batches of size 128. For GD with momentum, the momentum coefficient is set to 0.9.

Since different methods admit different thresholds for effective learning rates, we manually tuned the step sizes for each optimizer. The learning rates used to generate the reported results are summarized in Table 1. In all cases, we train the model until the training loss converges.[8]

D.2 Experiments with Large Neural Networks

In the experiments on MNIST (Appendix C.2), we use a two-layer MLP with sigmoid activation and hidden size 1024. The model is trained for classification using cross-entropy loss, without any additional regularization (e.g., weight decay). The batch size is set to 1024. For GD, the small and large learning rates are 0.02 and 0.5, respectively. For Adam, the corresponding values are 10^{-5} and 10^{-3}.

[8]Although in some cases the small and large step sizes appear close, we observed that low-dimensional networks are highly sensitive to the learning rate when trained with GD, possibly due to a degenerated loss landscape. For instance, in the 2D case, if $\eta = 2 \times 10^{-3}$ the neurons remain nearly unchanged, whereas for $\eta = 4 \times 10^{-3}$ the loss diverges. Thus we must compare within a relatively narrow range of learning rates.



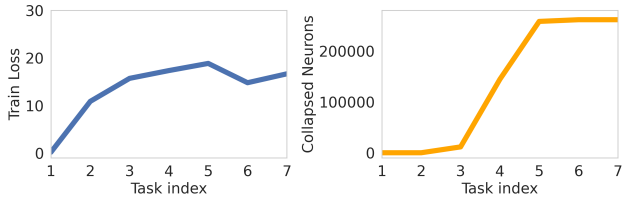
(a) Initialization (b) End of training, small η (c) End of training, large η

Figure 13: **Topology of a multi-layer 2D neural network with gradient descent.** The neurons are initialized on a genus-2 surface and optimized with GD.

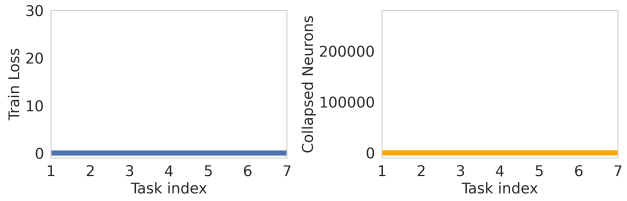
The Betti numbers are calculated with the GUDHI library [Maria et al., 2025]. When computing Betti numbers for the neuron-induced point cloud, a minimal distance (i.e. scale) must be chosen to decide how close two points need to be for them to be considered as neighbors. To ensure robustness to scale changes during training, we adopt a self-adaptive strategy for deciding the minimal distance: the minimal distance is set to $1/4$ times the diameter of the point cloud.

D.3 Experiments with Loss of Plasticity

In the experiment with multi-task training, the input data are i.i.d. standard Gaussian vectors with dimensionality 8. For each task, we sample a random linear mapping from the input data to an 8-dimensional space, as the teacher model, that acts on the input data to generate the target data. The input data and teacher model are resampled every epoch to change the task. Each task contains 50000 training samples and 10000 evaluation samples. The model is trained with GD on the mean-square-error (MSE) loss. The model structure is the same as that in Appendix D.2, with random initialization. For large and small learning rate settings, the learning rates are set to 0.25 and 0.01, respectively, and the weight decay rate is set to 0.01. The first task (with index 0) is viewed as a warm-up and discarded.



(a) Large learning rate



(b) Small learning rate

Figure 14: **The Multi-task learning results.** The neurons are initialized on a genus-2 surface and optimized with GD. After each epoch, the teacher model is resampled and the data is regenerated. "# of collapsed neurons" is calculated by counting the number of neuron pairs whose distance are smaller than 0.01.

Optimizer	Network dimension	Small learning rate	Large learning rate
GD	2D	2.5×10^{-3}	3×10^{-3}
GD	3D	8×10^{-4}	9×10^{-4}
Adam	2D	10^{-4}	10^{-2}
Adam	3D	3×10^{-2}	10^{-1}
Momentum	2D	5×10^{-4}	4.5×10^{-3}
Momentum	3D	10^{-3}	1.25×10^{-3}
Multi-Layer	N/A	0.01	0.05

Table 1: Learning rates used in the experiments for low-dimensional neural networks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See Sections 4 to 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Assumptions and problem settings are provided in Section 3; proofs are provided in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiment details are provided in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is provided in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experiment details are provided in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard deviations are shown in the figures as shaded areas in quantitative experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: This is a theory-oriented paper, and the experiments are only for the verification of the theoretical results. In this context, computational resources are not a critical factor for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Packages used in the experiment code have been properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.