Neural Expansion: A Unified Mechanism for How Deep Neural Networks Generalize

Chashi Mahiul Islam[1] **Samuel Jacob Chacko**[1] **Mao Nishino**[2]
Canlin Zhang[3] **Xiuwen Liu**[1]
[1]Department of Computer Science, Florida State University
[2]Department of Mathematics, Florida State University
[3]Independent
{cislam,sjacobchacko,mnishino,xliu}@fsu.edu
canlingrad@gmail.com

Abstract

Deep neural networks demonstrate remarkable generalization despite being highly over-parameterized, challenging traditional statistical learning theories. While several theoretical paradigms such as margin-based bounds, PAC-Bayesian analyses, and neural tangent kernels provide global explanations, they often lack mechanistic insight into how generalization emerges from training dynamics. We introduce Neural Expansion, a unified geometric framework that characterizes generalization through the expansion of directionally-selective ray coverage in the input space. Using Generalization Intervals (GIs), we quantify direction-specific regions around training samples where the model's predictions remain stable. Our extensive empirical studies across diverse architectures and datasets reveal that training leads to progressive expansion of ray coverage, particularly along directions with low Jacobian singular values. This mechanism helps explain a large fraction of correctly classified test inputs, adversarial examples, and even mislabeled or out-of-distribution samples through the geometry of ray coverage rather than instance-level memorization. By connecting input-space curvature to predictive behavior, our framework provides a mechanistic foundation that complements existing global theories of deep learning. Code and instructions for running the experiments are available at https://github.com/meetchashi/neural_expansion_neurips_2026.

1 Introduction

The remarkable generalization capability of deep neural networks presents a fundamental theoretical challenge. Despite having millions or billions of parameters, far exceeding their training dataset size, these models generalize well to unseen examples, contradicting classical learning theory predictions (Zhang et al., 2017). This generalization persists even without explicit regularization, suggesting intrinsic mechanisms beyond conventional capacity control (Neyshabur et al., 2017; Bartlett et al., 2017).

Traditional frameworks for understanding generalization, such as VC-dimension theory and uniform convergence bounds, provide vacuous guarantees for modern neural networks. While recent approaches, including spectral norm-based margins (Bartlett et al., 2017), PAC-Bayesian bounds (Neyshabur et al., 2017), compression-based complexity (Arora et al., 2018), neural tangent kernels (Jacot et al., 2018), simplicity bias (Valle-Perez et al., 2019), sharpness-aware training (Andriushchenko and Flammarion, 2022; Andriushchenko et al., 2023), and input-space geometric invariance (Movahedi et al., 2025), have advanced our understanding, a critical question remains unaddressed: why does a specific test input receive a specific classification?

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

We propose **Neural Expansion**, a novel geometric framework explaining how networks develop directionally-selective ray coverage around training samples during learning. Our approach is based on *Generalization Intervals (GIs)*, which quantify directionally-sensitive regions around training points within which a model's classification remains invariant (Salman et al., 2020). Rather than defining complex decision boundaries in high-dimensional spaces, we analyze how networks progressively expand stable classification zones, *ray coverage*, in specific directions.

Neural Expansion refers to the progressive growth of this ray coverage during training, which becomes more pronounced in directions with low Jacobian singular values. Unlike margin-based approaches that focus on decision boundary distances, Neural Expansion captures direction-specific invariance properties that explain why certain test points get assigned to particular classes. Furthermore, while kernel theories (NTK) assume a fixed feature space and neural collapse focuses on terminal phases, our framework explains the dynamic formation of ray coverage throughout the entire training process.

Our key contributions include:

- A theoretical framework that explains sample-level generalization through directional stability in input space, connecting Jacobian singular values to classification behavior
- Empirical demonstration that neural networks form wide generalization intervals in directions with low curvature, creating expansive, overlapping ray coverage around training samples
- Evidence that test-time classifications occur because test samples fall within the ray coverage of training samples, not because of instance-level memorization
- A geometric perspective showing how standard generalization, adversarial vulnerability, long-tailed misclassification, and random-label learning can be interpreted through the directional structure of generalization intervals

In this work, we establish that the local, sample-level geometry of neural network behavior is not merely an empirical curiosity but a theoretically grounded consequence of gradient-descent training. This perspective is complementary to existing theories of generalization, which provide valuable insights into optimization dynamics and global complexity measures. By formally analyzing how Jacobian structure creates and expands directional corridors of stability in input space, we provide a missing link between representation geometry and classification decisions, one that applies across diverse architectures and dataset distributions.

2 Background and Related Work

Understanding why overparameterized neural networks generalize well remains one of the central open questions in deep learning theory (Zhang et al., 2017). Bengio et al. (2013) hypothesized that effective generalization emerges when learned representations disentangle underlying factors of variation. The success of deep networks has since spurred diverse theoretical perspectives. Our work introduces a local, mechanistic framework based on Generalization Intervals (GIs) (Salman et al., 2020), focusing on individual sample-level classification behavior through Jacobian-derived directional invariance.

Classical approaches rely on complexity-based measures such as VC dimension, Rademacher complexity, and margins. Spectral norm-based bounds (Bartlett et al., 2017), PAC-Bayesian frameworks (Neyshabur et al., 2017), and compression-based approaches (Arora et al., 2018) offer compelling global insights. Our GI framework provides a local geometric lens, explaining generalization through stable directional regions around training samples where predictions remain invariant.

The Neural Tangent Kernel (NTK) approximates wide networks as kernel regression around initialization (Jacot et al., 2018), formalized and contrasted with richer feature-learning regimes (Chizat et al., 2019; Woodworth et al., 2020) and extended to explain triple descent (Adlam and Pennington, 2020). These approaches focus on global dynamics near initialization; our work instead quantifies post-training directional stability via the Jacobian, capturing piecewise-linear generalization regions in realistic regimes.

Neural Collapse describes how class means, features, and classifier weights align near convergence (Papyan et al., 2020; Han et al., 2021; Wu and Papyan, 2024), with recent work linking this to the

average gradient outer product (Beaglehole et al., 2024). Our GI framework shows that classification decisions arise from stable, low-curvature regions that generalize across architectures and tasks.

Generalization has also been attributed to a simplicity bias (Valle-Perez et al., 2019), supported by double and deep double descent (Belkin et al., 2018; Zhang et al., 2017; Nakkiran et al., 2021). Recent work shows that seeking flat loss regions improves generalization (Foret et al., 2021), SGD implicitly minimizes complexity-promoting measures (Vardi, 2023), and networks can benignly overfit noisy data under favorable signal-to-noise conditions (Frei et al., 2023), phenomena our framework explains geometrically via Jacobian curvature.

3 A Theoretical Framework for Neural Network Generalization

In this section, we develop our theoretical framework that captures the directional properties of generalization in deep neural networks. We introduce the **generalization interval (GI)**, which captures how far we can go in a given direction without changing the class, and connect it to Jacobian singular value analysis.

Definition 1. *Let $f : \mathbb{R}^d \to \mathbb{R}^c$ be a c-class classifier outputting the logits for each class, $x_0 \in \mathbb{R}^d$ be a given input data, $y \in \{1, \cdots, c\}$ be its class and $d \in \mathbb{R}^d$ be the direction to traverse. The **generalization interval (GI)** of f from the input x_0 in the direction of d is defined by*

$$GI(f, x_0, d) = \sup\{a > 0 | f^y(x_0 + td) > \max_{i \neq y} f^i(x_0 + td) \text{ for all } t \in [0, a]\} \tag{1}$$

This quantity measures how far we can go in the given direction before we reach the decision boundary. The idea is similar to the classical concept of **margins**, but a GI is direction-dependent: a model might exhibit large GIs in some directions and small GIs in others around the same point. Nevertheless, there is an important relationship between the margins and the GIs, which we will discuss later. The act of finding a GI can be described as shooting a *ray* from the given input until we reach the decision boundary. As such, it is natural to define the following:

Definition 2. *The ray coverage of x_0 is the set of all points that lie within their respective GI bounds:*

$$RC(x_0) = \left\{ x : GI\left(f, x_0, \frac{x - x_0}{\|x - x_0\|} \right) \geq \|x - x_0\| \right\}$$

3.1 Jacobian Analysis and GI Formation

To understand why neural networks exhibit directionality in ray coverages through sensitivity, we analyze the Jacobian of the network's representation function. Let $\phi : \mathbb{R}^d \to \mathbb{R}^m$ denote the mapping from input to the penultimate layer (i.e., the representation), and $W \in \mathbb{R}^{c \times m}$ be the final linear classifier, such that $f(\mathbf{x}) = W\phi(\mathbf{x})$.

The Jacobian $J_\phi(\mathbf{x}) \in \mathbb{R}^{m \times d}$ captures how the representation changes with respect to small perturbations in the input. We perform singular value decomposition:

$$J_\phi(\mathbf{x}) = U \Sigma V^T = \sum_{i=1}^{\min(m,d)} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{2}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,d)} \geq 0$ are the singular values, $\mathbf{v}_i \in \mathbb{R}^d$ are the right singular vectors (directions in input space), and $\mathbf{u}_i \in \mathbb{R}^m$ are the left singular vectors (directions in representation space).

Our empirical analysis reveals a crucial property: we observe that, in trained neural networks, the vast majority of singular values are close to zero (See Figure 2b). For instance, in a network trained on CIFAR-10 (3,072-dimensional input), typically only 10-15 singular values are significantly non-zero, while over 99.5% of the singular value spectrum is effectively zero (Appendix C). This spectral structure has implications for generalization due to Theorem 1. For a right singular vector direction v of J_ϕ, $\|J_f(x_0)v\|$ is proportional to the corresponding singular value. Therefore, the theorem predicts that the GI is large for all but a few directions due to the directional sensitivity of the model, which is consistent with the empirical observation we will present later in Figure 2a.

3.2 Linear Approximation of the GI and the Relation to the Margin

As the GI can be described as the margins of samples with directionality, they naturally have a close relationship. To describe it, we first define the margin:

Definition 3. *Let $f : \mathbb{R}^d \to \mathbb{R}^c$ be a c-class classifier, $x_0 \in \mathbb{R}^d$ be a given input data, and $y \in \{1, \cdots, c\}$ be its class. The **margin** of the sample (x_0, y) is $f^y(x_0) - \max_{i \neq y} f^i(x_0)$ and we denote it by q_0. Here, $f^i(x_0) \in \mathbb{R}$ means the ith component of $f(x_0)$.*

We now have the following relation:

Theorem 1. *Let $x_0, d \in \mathbb{R}^d$ be a given input and a given direction. Suppose $y \in \{1, \cdots c\}$ is the class of x_0. Assume that the classifier $f : \mathbb{R}^d \to \mathbb{R}^c$ is differentiable with an inhomogeneous directional derivative: the yth component of $J_f(x_0)d$ is not equal to any other component. Consider the linear approximation $f(x_0 + \epsilon d) = f(x_0) + \epsilon J_f(x_0)d + R(x_0, \epsilon, d)$ where R is the error term satisfying $\|R(x_0, \epsilon, d)\|_1 < \delta$ for some $\delta > 0$, $J_f(x_0)$ denotes the Jacobian matrix for f at x_0 and $\|\cdot\|_1$ is the ℓ^1-norm. Then*

$$\frac{q_0 - \delta}{\|J_f(x_0)d\|_1} \leq \text{GI}(f, x_0, d) \tag{3}$$

The proof can be found in Appendix J. This result tells us that, under linear approximation, a larger margin would indicate a larger generalization interval. This partially explains why the GIs should increase under gradient descent training, as previous literature such as (Lyu and Li, 2020) shows that a smooth approximation of the margin for a homogeneous network would monotonically increase under gradient descent. We note that, as we show later, the linear approximation holds surprisingly well (see Section 5.1.3). Nonlinear effects are accounted for through the assumed residual bound δ (Appendix D.1).

The lower bound has the norm of the directional derivative $\|J_f(x_0)d\|$, and thus, sensitivity in each direction would contribute to the size of the GIs.

3.3 GI Strictly Refines Spectral Margin Along Singular Directions

Theorem 1 establishes that the GI lower bound grows with margin and shrinks with directional sensitivity. This raises a natural question: does GI offer strictly more information than classical spectral margin theory (Bartlett et al., 2017), or does it merely restate the same global bound in geometric language? We show that GI strictly refines spectral margin by decomposing input space direction-by-direction in a way that spectral margin cannot.

Theorem 2 (GI–Spectral Margin Decomposition)**.** *Under the same setting as Theorem 1, let $J_\phi(x_0) = U\Sigma V^\top$ be the SVD of the penultimate-layer Jacobian with singular values $\sigma_1 \geq \cdots \geq \sigma_{\min(m,d)} \geq 0$ and right singular vectors $v_i \in \mathbb{R}^d$. Then for each v_i:*

$$\text{GI}(f,\, x_0,\, v_i) \;\geq\; \frac{q_0 - \delta}{\sqrt{c}\,\|W\|_{\text{op}}\,\sigma_i} \tag{4}$$

As $\sigma_i \to 0$ with $q_0 > \delta$, this bound grows without bound, while the spectral margin bound remains fixed and direction-agnostic.

The bound decomposes input space into *sensitive directions* (large σ_i) and *invariant directions* (small σ_i). Spectral margin theory captures only the former; GI captures both. Since a significant number of singular directions in trained networks have $\sigma_i \approx 0$ (Section 3.1), GI provides nontrivial directional information in precisely the regime where spectral margin is silent. The proof is in Appendix K.

3.4 Jacobian control of the input-loss gradient

Theorem 2 reveals that directions with large singular values yield a small lower bound on GIs. A deeper question follows: do these high-sensitivity, low-GI directions correspond to the directions where the loss landscape is sharpest? To partially answer this question, we bound the input-loss gradient in terms of the largest Jacobian singular value, connecting the loss landscape and the most sensitive direction.

Theorem 3 (GI and Loss Landscape Sharpness)**.** *Under the same setting as Theorem 1, let* $\mathcal{L}(x) = -\log \sigma_y(f(x))$ *be the cross-entropy loss at* x_0 *with true class* y*, and let* $\nabla_x \mathcal{L}(x_0)$ *be its gradient with respect to the input. Then:*

$$\|\nabla_x \mathcal{L}(x_0)\|_2 \;\leq\; \sqrt{2}\,\|W\|_{\mathrm{op}} \cdot \sigma_1 \tag{5}$$

where σ_1 *is the largest singular value of* $J_\phi(x_0)$*.*

Equation (5) shows that input-space sharpness is controlled by the top Jacobian singular value σ_1, the same quantity that governs GI via Theorem 2. Empirically, the largest right singular vector v_1 also closely approximates the unconstrained steepest ascent direction, as shown in Figure 10. This connects Neural Expansion directly to the flat minima literature (Keskar et al., 2017): low-curvature directions (large index i, small σ_i) correspond to flat directions in the loss landscape and large GIs, while high-curvature directions (small index i, large σ_i) correspond to sharp directions and small GIs. The proof is in Appendix L.

3.5 GI Quantifies the Adversarial Robustness Radius

The GI provides a direct lower bound on the size of adversarial perturbations. We now show that every successful class-changing perturbation has norm at least the minimum GI over unit directions, and derive a further lower bound in terms of the margin and Jacobian norm.

Theorem 4 (Adversarial Perturbation Radius via GI)**.** *Under the same setting as Theorem 1, let* $\Delta \in \mathbb{R}^d$ *be any perturbation that changes the classification at* x_0*, i.e.,* $\arg\max_i f^i(x_0 + \Delta) \neq y$*. Then:*

$$\|\Delta\|_2 \;\geq\; \min_{\|d\|=1} \mathrm{GI}(f, x_0, d) \tag{6}$$

Furthermore, combining with Theorem 2:

$$\|\Delta\|_2 \;\geq\; \frac{q_0 - \delta}{\sqrt{c}\,\|W\|_{\mathrm{op}}\,\sigma_1} \tag{7}$$

Equation (6) establishes a bound on the adversarial robustness radius: large GIs in every direction guarantee robustness. Equation (7) makes this quantitative in terms of observable quantities: margin q_0, classifier norm $\|W\|_{\mathrm{op}}$, and the largest Jacobian singular value σ_1. The proof is in Appendix M.

4 Experimental Setup

Full implementation details are provided in Appendix A.

Datasets: We evaluate on three benchmark datasets: MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky and Hinton, 2009), and ImageNette (Deng et al., 2009). To assess generalization under more challenging regimes, we include random label variants and long-tailed variants (following Zhang et al. (2017) and Ma et al. (2022)). Full details on these datasets and their variants are provided in Appendix A.1.

Model Architectures: To demonstrate the universality of Neural Expansion, we evaluate three architecture families: Multi-Layer Perceptrons, Convolutional Networks (e.g., ResNet-20/50, VGG-16), and Vision Transformers (including ViT-B/16 and DeiT-S). All models are trained with SGD using dataset-appropriate hyperparameters until convergence. Full architecture and training details are provided in Appendix A.2.

Generalization Interval Analysis: Our primary analytical tool is the Generalization Interval (GI), which quantifies directional prediction stability in input space. GIs are computed using a step-wise search algorithm (Appendix A.4) along right singular vectors (RSVs) of the penultimate-layer Jacobian (Section 3.1), combined with ray coverage measurement toward training samples and linear approximation evaluation (Section 5.1.3). Together, these methods allow us to analyze the formation of ray coverage during training, its alignment with Jacobian structure, and its implications across standard, long-tailed, random-labeled, and adversarial test settings.

5 Results

5.1 GI Behavior Across Model Types and Training Scenarios

We conduct a systematic analysis of generalization interval (GI) behavior across model architectures, training scenarios, and datasets. GIs are computed along the RSVs of the input-to-penultimate-layer Jacobian. To ensure consistency, each RSV direction is scaled by a factor of 56.0, the maximum distance in the normalized 784-dimensional MNIST input space. Unless otherwise noted, all experiments use an MLP architecture (784–1024–512–10). Our results reveal consistent GI patterns across settings, pointing to a common geometric structure underlying neural network generalization across the settings we tested.

5.1.1 Temporal Evolution of Generalization Intervals During Training

We observe how GI-based ray coverage progressively develops during the training process in Figure 1. Both the mean GI and the proportion exceeding 1.0 increase steadily, strongly correlating with improvements in test accuracy (Pearson $r = 0.99$ across epochs, range 0.994–0.998 over three runs). While both RSV and random directions (isotropic Gaussians, unit-normalized) exhibit GI expansion, RSV directions consistently show larger GIs and faster growth rates, particularly in later epochs. This demonstrates the network's preferential expansion of ray coverage along low-curvature directions identified by the Jacobian SVD (Appendix B).

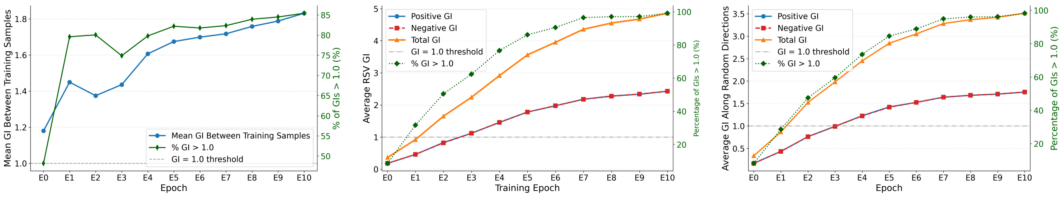


Figure 1: Evolution of ray coverage during training of an MNIST MLP over 10 epochs. **Left:** Mean positive GI growth between same-class training samples, with percentage of GIs exceeding 1.0. **Middle:** Average GI along RSV directions. **Right:** Average GI along random directions.

5.1.2 GI Expansion in Early vs. Later Training

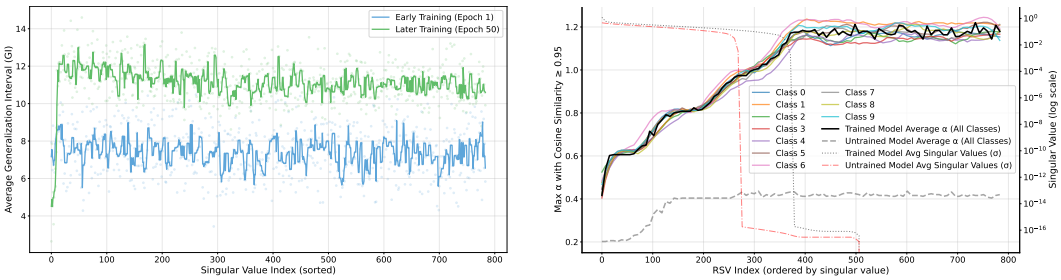
Figure 2a compares GI magnitudes across RSV directions between an early-stage model (epoch 1) and a fully trained model (epoch 50). In the early phase, GI values are relatively uniform, averaging around 7 across RSVs. After training, the distribution becomes skewed, GIs increase substantially along high-index (low-curvature) RSV directions and decrease along low-index (high-curvature) directions, with the average rising to 11.5. This indicates that training selectively expands ray coverage along directions associated with smaller singular values. We observe 512 non-zero singular values, corresponding to the representation layer dimension, while the remaining RSV directions are null due to zero singular values.

5.1.3 Linear Approximation Along Directions with Different Singular Values

Figure 2b quantifies the local linearity of the model along RSV directions by measuring the maximum distance α for which a linear approximation of the model output maintains a cosine similarity of at least 0.95 with the true output (evaluated in the penultimate-layer representation space).

In the untrained model (random initialization), linear approximation quality is uniformly low: α ranges from ~0.2 in high-curvature directions (low RSV indices) to ~0.4 in flatter directions, indicating limited local linearity across all directions. After training (49 epochs), the model exhibits a clear monotonic trend where linearity improves substantially with RSV index, starting from $\alpha \approx 0.4$ and reaching $\alpha \approx 1.2$ in low-curvature directions. This coincides with a slower decay in the Jacobian's singular values, which extend further into high-index RSVs compared to the untrained case.

The sharp transition around indices 300–400 (Appendix D), where the singular value spectrum flattens, marks the boundary where the trained model's local linearity begins to dominate. These



(a) (b)

Figure 2: (a) GI magnitudes across RSV directions in early- vs. late-stage models (epoch 1 vs. epoch 50). Training increases GIs selectively along high-index (low-curvature) directions. (b) Comparison of linear approximation quality along RSV directions before and after training.

results reinforce our hypothesis: the directions supporting the longest GIs also align with those where the model behaves most linearly. Training amplifies this structure, enabling extended ray-based coverage without changes in prediction.

5.2 Effect of Ray Coverage Interventions on Generalization

In this section, we show that intervening on ray coverage changes the model's generalization ability, highlighting their close relationship beyond the post-hoc analysis we have so far. In Appendix H.1, we show theoretically that, under what we call the *no-re-entry-assumption*, i.e., the assumption that the model does not predict the original class after leaving GI, an increase in GI implies a decrease in the prediction error. This empirical experiment expands on the theoretical result.

The effect of modifying ray coverage is evaluated using an auxiliary loss applied to perturbed training inputs. In the **expand** condition, correct classification of the perturbed input is encouraged. In the **restrict** condition, confident misclassification of the perturbed input is encouraged, while cross-entropy is retained at the original training input. The **control** condition uses cross-entropy alone. An MNIST-MLP and a Fashion-MNIST-CNN are trained under each condition using three random seeds. Full loss definitions and hyperparameters are provided in Appendix H.

Table 1: Test accuracy, mean GI, and ray coverage under control, expansion, and restriction conditions. Values are reported as mean $\pm$ standard deviation over three random seeds at the final training epoch.

Model	Condition	Test Acc. (%)	Mean GI	Ray Coverage (%)
MNIST MLP	Control	87.38 ± 0.08	1.872 ± 0.045	78.33 ± 1.04
	Expand	89.04 ± 0.08	1.943 ± 0.054	81.17 ± 1.26
	Restrict	75.18 ± 1.18	1.498 ± 0.025	60.67 ± 0.76
Fashion-MNIST CNN	Control	85.27 ± 0.18	2.005 ± 0.017	73.83 ± 0.29
	Expand	86.37 ± 0.29	2.037 ± 0.011	74.33 ± 0.58
	Restrict	77.86 ± 1.29	1.492 ± 0.092	54.33 ± 3.21

As shown in Table 1, coverage restriction reduces test accuracy relative to the control by 12.2 percentage points on MNIST-MLP ($p = 0.003$) and 7.4 percentage points on Fashion-MNIST-CNN ($p \approx 0.008$). Coverage expansion increases accuracy by 1.7 and 1.1 percentage points, respectively (both $p \leq 0.008$). Corresponding changes in GI and ray coverage are observed from the first epoch across all seeds. These intervention results are consistent with the theoretical relationship between GI expansion and population error and support a causal contribution of ray coverage to generalization.

5.3 Ray Coverage as a Mechanism for Test-Time Classification

Building on the directional invariance captured by GIs, we show that ray coverage provides a mechanistic explanation for test-time classification decisions. This framework unifies standard generalization, random label learning, long-tailed distribution learning, and adversarial vulnerability under a single geometric perspective.

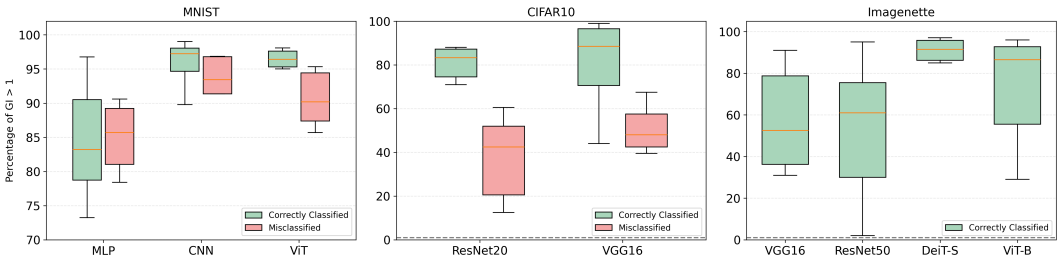


Figure 3: Test Sample Classification via Ray Coverage. Box plots show the percentage of GIs exceeding 1.0 between test and training samples of the corresponding class. Both correctly classified and misclassified samples exhibit substantial ray coverage across datasets and architectures. The gray dashed line indicates the threshold for directional coverage, where a test sample is considered covered if at least 1% of training samples from the same class exhibit a GI greater than 1.0. Misclassified bars are absent for the ImageNette models because no misclassified samples appeared in the evaluated subset (k=20) of test samples per class, likely due to the limited sample size.

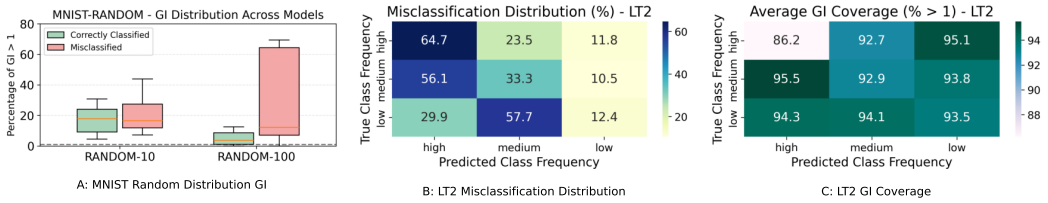


Figure 4: **Left:** Random Label Learning with GI ray coverage for RANDOM-10 and RANDOM-100 models. The gray dashed line (as defined in Figure 3) indicates the 1% directional coverage threshold. **Middle:** LT2 Misclassification Distribution showing percentage distribution from true class frequency to predicted class frequency. **Right:** LT2 GI Coverage showing average percentage of GIs exceeding 1.0 across different frequency categories.

5.3.1 Test Sample Classification via Ray Coverage

We analyze the relationship between test sample classification and the ray coverage formed by training samples across MNIST, CIFAR-10, and ImageNette datasets using multiple architectures. For each test sample, we compute GIs along the direction to all training samples of the predicted class and report the percentage of GIs exceeding 1.0, representing how often the test sample falls within the ray coverage region of the predicted class.

We observe consistent patterns across datasets and models; Figure 3 reveals consistently high coverage across datasets and architectures for both correctly classified and misclassified samples.

These findings suggest that test-time predictions arise from the geometry of class-level ray coverage rather than instance-level memorization. Even misclassified inputs often fall within the invariant region of the predicted class, supporting Neural Expansion as a general mechanism underlying classification across varying accuracy regimes. We include more details about the experiment in Appendix E. Class trajectory visualizations along train-to-test paths are shown in Appendix G. These measurements exhibit seed stability and class-discriminability (Appendices F.1, F.2).

5.3.2 Neural Expansion as a Unified Framework for Challenging Cases

We examine whether ray coverage remains the operative mechanism in challenging settings of traditional learning: random label learning, long-tailed distributions, and adversarial perturbations.

Random Label Learning: Even when label semantics are randomized (RANDOM-10, RANDOM-100 [1]), we find that test samples consistently fall within ray coverage zones around training samples with the same assigned label. The plot in Figure 4a reveals nontrivial GI overlap between test and training samples of the same (randomly assigned) label, indicating that the model still forms geometric

[1] Since the labels are randomized, we are free to choose the total number of classes as well.

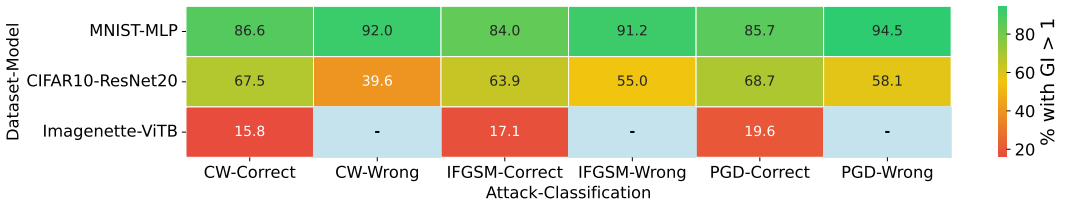


Figure 5: Adversarial Examples Analysis (misclassified samples indicate successful attacks; correct classifications indicate failures). The heatmap shows the percentage of adversarial examples with GI > 1 to training samples of the predicted class across different dataset-model combinations and attack methods (CW, IFGSM, PGD).

zones for each label, albeit smaller and noisier. This finding challenges the common assumption that networks simply "memorize" random labels without forming structure. (See Appendix I.1 for dataset-specific coverage values.)

Long-Tailed Distributions: In highly imbalanced datasets (e.g., MNIST-LT2), test samples from underrepresented classes are disproportionately misclassified into medium- or high-frequency classes (Figure 4b). The ray coverage plot in Figure 4c shows that this is not due to failure in learning low-frequency classes, but rather to the expansive coverage of frequent classes dominating the input space. Misclassifications align with the regions where dominant classes extend their influence. (Detailed statistics in Appendix I.2.)

Adversarial Examples: We analyze how adversarial examples interact with ray coverage using three attack methods: Carlini-Wagner (CW) (Carlini and Wagner, 2017), Iterative Fast Gradient Sign Method (IFGSM) (Kurakin et al., 2017), and Projected Gradient Descent (PGD) (Madry et al., 2018). Figure 5 reveals the consistent presence of ray coverage regardless of whether adversarial perturbations successfully change the classification. In many cases, the attack appears to shift the test input from one class's coverage zone to another's, rather than entirely escaping the structure. This trend persists across MNIST and CIFAR-10.

For ImageNette-ViTB, correctly classified adversarial examples show low ray coverage. Notably, we could not compute coverage values for misclassified ImageNette examples, as they were often pushed outside our 10-class subset into the broader 1000-class ImageNet space. This illustrates how high-dimensional natural images provide more pathways for adversarial examples to escape class boundaries than constrained datasets like MNIST. Appendix I.3 provides more details regarding the geometry of adversarial perturbations. To further examine model behavior under distribution shift, we also compute GI coverage for out-of-distribution samples (see Appendix I.4). The anisotropic growth of GIs during training, driven by margin increase and directional restructuring of Jacobian singular values, is analyzed in Appendix N.

6 Discussion

Our results show that ray coverage is present in every setting we tested, including standard training, random labels, long-tailed data, and adversarial inputs, although its extent varies. It appears even where the labels carry no semantic meaning: networks trained on random labels still form coverage zones around training samples, and misclassified test samples still fall inside the coverage of the (wrong) predicted class. This suggests that gradient descent does not merely fit labels; it appears to organize input space into directional corridors, regardless of whether those corridors reflect semantic structure. Without semantic supervision, however, these corridors become substantially sparser and more fragmented (17%–24% coverage under random labels, versus roughly 83%–97% for MNIST models trained on true labels). Memorization and generalization may therefore both operate through ray coverage, with generalizing networks building wider, overlapping corridors along low-sensitivity directions that capture class-relevant structure.

This picture complements existing theories in ways the global bounds do not. PAC-Bayesian (Neyshabur et al., 2017) and margin-based bounds (Bartlett et al., 2017) certify that a model generalizes globally, but cannot say *why* a specific test point is classified correctly. NTK theory (Jacot et al., 2018) and Neural Collapse (Papyan et al., 2020) describe either the initialization

regime or the terminal phase. Our framework instead examines the geometry around individual training samples as training progresses, which offers a sample-level view that global theories do not provide.

Limitations and Future Work. GI computation remains expensive for high-dimensional inputs, and while our sampling-based approach scales to architectures such as ViTs, full coverage requires further approximation strategies. The framework is currently most natural for classification; extending it to regression, generation, or structured prediction requires reformulating GIs for settings without discrete decision boundaries. A promising future direction is to integrate these insights directly into training objectives, encouraging networks to explicitly maximize ray coverage in low-curvature directions to enhance generalization and robustness. Finally, the consistent emergence of ray coverage across model classes suggests that gradient-based learning may implicitly align with the intrinsic geometry of natural data distributions, a connection that could yield new principles for representation learning and regularization. A rigorous proof that individual Jacobian singular values restructure anisotropically during gradient descent training remains an open theoretical question; our current account in Appendix N relies on empirical observations.

7 Conclusion

We present Neural Expansion, a geometric framework that reinterprets generalization in deep neural networks through the lens of directionally-sensitive invariance pathways. Through extensive empirical analysis spanning diverse architectures, datasets, and training regimes, we uncover a coherent geometric signature: a compact set of high-curvature directions sculpts class boundaries, while an abundance of low-curvature directions carves out invariance corridors that link samples belonging to the same class. This dual structure provides a unifying account of seemingly disparate phenomena, namely standard generalization, memorization of random labels, misclassification of tail-class samples, and susceptibility to adversarial perturbations, all through a common geometric lens. The resulting picture reframes our understanding of what neural networks learn: rather than constructing intricate decision surfaces, they organically develop stable classification pathways whose collective geometry mirrors the latent structure of the data.

References

Adlam, B. and Pennington, J. (2020). The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*, pages 74–84. PMLR.

Andriushchenko, M., Bahri, D., Mobahi, H., and Flammarion, N. (2023). Sharpness-aware minimization leads to low-rank features. In *Advances in Neural Information Processing Systems*, volume 36.

Andriushchenko, M. and Flammarion, N. (2022). Towards understanding sharpness-aware minimization. In *Proceedings of the 39th International Conference on Machine Learning*, pages 639–668. PMLR.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. (2018). Stronger generalization bounds for deep nets via a compression approach. In *International conference on machine learning*, pages 254–263. PMLR.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30.

Beaglehole, D., Súkeník, P., Mondelli, M., and Belkin, M. (2024). Average gradient outer product as a mechanism for deep neural collapse. *Advances in Neural Information Processing Systems*, 37:130764–130796.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2018). Reconciling modern machine learning practice and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118 v2*, 14.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE.

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

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2021). Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*.

Frei, S., Vardi, G., Bartlett, P., and Srebro, N. (2023). Benign overfitting in linear classifiers and leaky relu networks from kkt conditions for margin maximization. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 3173–3228. PMLR.

Han, X., Papyan, V., and Donoho, D. L. (2021). Neural collapse under mse loss: Proximity to and dynamics on the central path. *arXiv preprint arXiv:2106.02073*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kurakin, A., Goodfellow, I., and Bengio, S. (2017). Adversarial examples in the physical world. In *ICLR Workshop*.

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

Lyu, K. and Li, J. (2020). Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*.

Ma, Y., Jiao, L., Liu, F., Li, Y., Yang, S., and Liu, X. (2022). Delving into semantic scale imbalance. *arXiv preprint arXiv:2212.14613*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Movahedi, S., Orvieto, A., and Moosavi-Dezfooli, S.-M. (2025). Geometric inductive biases of deep networks: The role of data and architecture. In *International Conference on Learning Representations*, volume 2025, pages 80383–80416.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2021). Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. (2017). A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*.

Papyan, V., Han, X., and Donoho, D. L. (2020). Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663.

Salman, S., Zhang, C., Liu, X., and Mio, W. (2020). Towards quantifying intrinsic generalization of deep relu networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Valle-Perez, G., Camargo, C. Q., and Louis, A. A. (2019). Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*.

Vardi, G. (2023). On the implicit bias in deep-learning algorithms. *Communications of the ACM*, 66(6):86–93.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. (2020). Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR.

Wu, R. and Papyan, V. (2024). Linguistic collapse: Neural collapse in (large) language models. *Advances in Neural Information Processing Systems*, 37:137432–137473.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

A Detailed Experimental Setup

A.1 Datasets and Variants

We use four benchmark datasets: **MNIST** (LeCun et al., 1998) (28×28 grayscale, 10 classes), **Fashion-MNIST** (Xiao et al., 2017) (28×28 grayscale, 10 classes), **CIFAR-10** (Krizhevsky and Hinton, 2009) (32×32 RGB, 10 classes), and **ImageNette** (Deng et al., 2009), a 10-class subset of ImageNet with 224×224 RGB images. Fashion-MNIST is used for the causal intervention experiment (Section 5.2) and the robustness analyses in Appendix F.

To test generalization under different distributions, we created variants including:

- **Random Label Variants**: RANDOM-10 (random labels across 10 classes) and RANDOM-100 (random labels across 100 classes), following Zhang et al. (2017).
- **Long-tailed Variants**: MNIST-LT1 (imbalance ratio 6.5:1) and MNIST-LT2 (78.6:1), following Ma et al. (2022).

Images are normalized to $[-1, 1]$ using dataset-specific preprocessing. For ImageNette, we apply standard ImageNet normalization, random cropping, horizontal flipping, and color jitter.

Licenses. MNIST is released under CC BY-SA 3.0, Fashion-MNIST under the MIT license, CIFAR-10 under the MIT license, and ImageNette and the ViT models under the Apache 2.0 license.

A.2 Model Architectures

We evaluate three architecture families:

- **Multi-Layer Perceptrons (MLPs)**: For MNIST, we use a 784–1024–512–10 MLP with ReLU and batch normalization. For CIFAR-10, a 3072–2048–1024–10 MLP is used.
- **Convolutional Neural Networks (CNNs)**: We use custom CNNs for MNIST and CIFAR-10, along with pre-trained ResNet-20, ResNet-50, and VGG-16 (He et al., 2016; Simonyan and Zisserman, 2015).
- **Vision Transformers (ViTs)**: Lightweight ViTs for MNIST (patch size 7×7, embedding dim 64) and CIFAR-10 (patch size 8×8, embedding dim 192). For ImageNette, we use ViT-B/16 and DeiT-S (Dosovitskiy et al., 2021).

A.3 Training Protocols

All custom models are trained using SGD until convergence with dataset-specific hyperparameters:

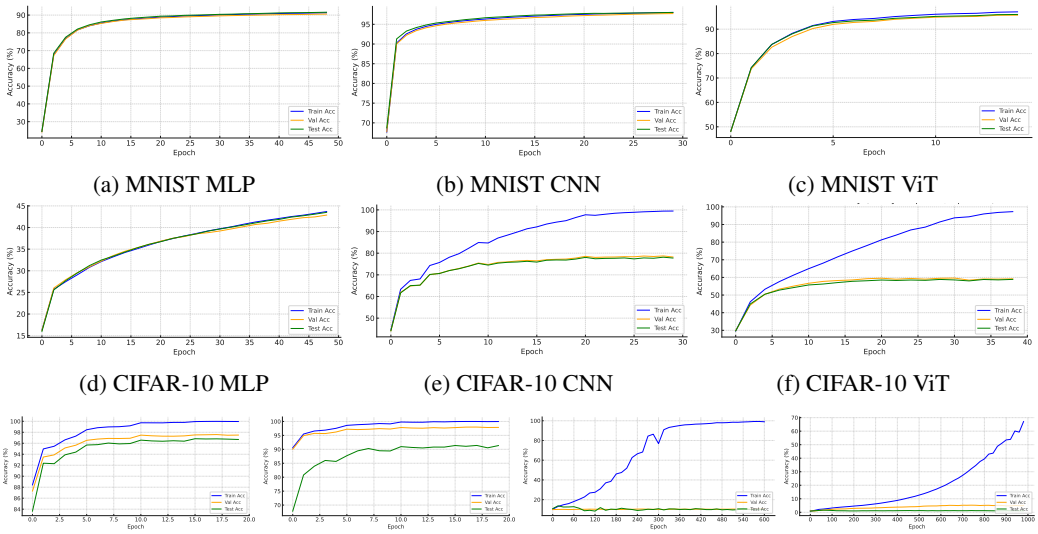
- **MNIST**: 30 epochs, learning rate 10^{-3}/10^{-2}, no momentum or weight decay.
- **CIFAR-10**: 50 epochs, learning rate 10^{-3}/10^{-2}, momentum 0.9, weight decay 10^{-4}.
- **ImageNette**: 30 epochs, learning rate 10^{-4}, momentum 0.9, weight decay 10^{-4}.

We include accuracy for all the custom-trained models in Figure 6.

A.4 Analysis Methods

Generalization Interval Calculation: We implemented a step-wise search algorithm to compute GIs between sample pairs. For direction vector $\mathbf{v} = \mathbf{x}_2 - \mathbf{x}_1$ (the unnormalized difference vector), we incrementally moved along both positive and negative directions ($\mathbf{x}_1 \pm \epsilon\mathbf{v}$) using step size $\delta = 0.001$-0.01 until either a decision boundary was crossed or a maximum $\epsilon = 3.0$ was reached. Under this convention, $\epsilon = 1.0$ exactly reaches $\mathbf{x}_2$, and the threshold $\mathrm{GI}(f, x_i, x_{\text{test}} - x_i) \geq 1$ in Equation (8) is dimensionally consistent with the unnormalized direction. Our vectorized implementation enabled batch processing (MNIST: 6000 pairs, CIFAR-10: 2000 pairs, ImageNette: 100 pairs). Table 2 contains the compute resources and runtime details of the GI coverage experiments.

Jacobian Analysis: For each sample, we computed the Jacobian matrix $J = \frac{\partial f(x)}{\partial x}$ of the penultimate layer features with respect to the input and performed SVD: $J = U\Sigma V^T$. We evaluated linear



(a) MNIST MLP (b) MNIST CNN (c) MNIST ViT

(d) CIFAR-10 MLP (e) CIFAR-10 CNN (f) CIFAR-10 ViT

(g) MNIST MLP LT1 (h) MNIST MLP LT2 (i) MLP Random-10 (j) MLP Random-100

Figure 6: Epoch-wise accuracy curves across models and datasets for vanilla as well as LT1, LT2, Random-10 and Random-100 variations.

Table 2: Compute resources and runtime details for test-to-train GI coverage experiments using a step-wise directional search.

Dataset	Architecture	GPU(s)	Runtime (hh:mm)	Memory Used (GB)
MNIST (6000 pairs)	**MLP**	A5000	00:30	3.0
	CNN	A5000	00:55	5.0
	ViT	A5000	01:25	6.5
CIFAR-10 (2000 pairs)	**ResNet20**	A5000	00:40	4.2
	VGG16-BN	A5000	00:55	5.1
ImageNette (100 pairs)	**VGG16**	A5000	01:40	6.8
	ResNet50	A5000	01:30	6.5
	ViT-B/16	A5000	01:55	7.8
	DeiT-Small	A5000	01:35	7.0

approximation quality by measuring cosine similarity between $f(x_0 + \alpha v_i)$ and $f(x_0) + \alpha W \sigma_i u_i$ for different α values, where both terms are c-dimensional logit vectors.

Ray Coverage Analysis: For each test sample with predicted class c, we calculated GIs to training samples of class c and computed the percentage exceeding 1.0. This quantified test samples' membership in training samples' ray coverage zones, enabling comparisons between correct and incorrect classifications across datasets, architectures, and distribution variants.

B GI Evolution Experimental Details

To analyze the temporal evolution of ray coverage during training, we tracked GI development in an MNIST MLP (784-1024-512-10) across 10 epochs. Figure 1 captures three key aspects of this evolution.

For train-to-train GI analysis (left panel), we calculated GIs between 100 randomly selected same-class training sample pairs for each digit class at checkpoints from epoch 0 to epoch 10. We measured both mean GI values and the percentage exceeding 1.0, which grew from 0.76 to 1.24 and 39% to 82% respectively, exhibiting strong correlation with test accuracy improvement (86.2% to 91.7%).

For directional analysis (middle and right panels), we computed GIs along RSVs of the Jacobian matrix and along random directions, using the same scaling factor of 56.0 for both to ensure fair comparison. This factor was derived from the maximum possible distance in the 784-dimensional normalized input space of MNIST. Along RSV directions, mean GI increased from 0.53 to 1.18, with the percentage exceeding 1.0 growing from 22% to 78%. In contrast, random directions showed more modest growth from 0.40 to 0.82, with the percentage exceeding 1.0 reaching only 34% by epoch 10.

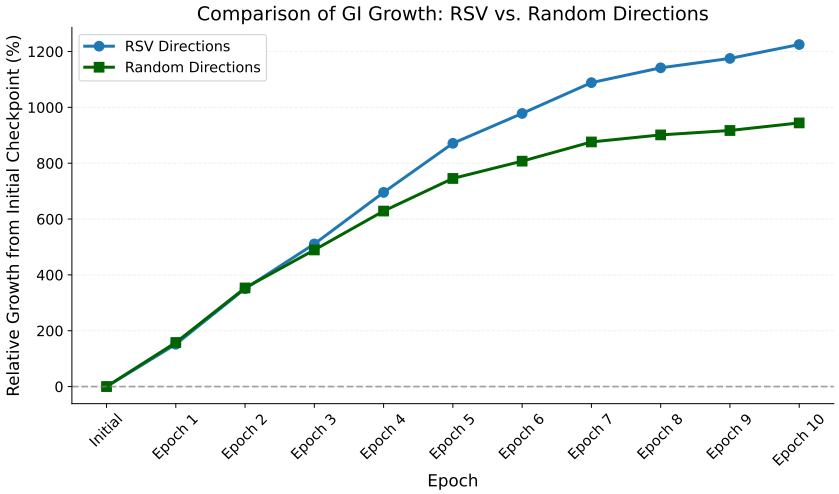


Figure 7: Relative GI growth along RSV vs. random directions during training. Both direction types show similar growth patterns through epoch 3, after which RSV directions exhibit substantially accelerated expansion, reaching 1220% growth by epoch 10 compared to 940% for random directions.

Figure 7 provides further insight into the differential growth rates of GIs along RSV versus random directions. When normalized to show relative growth from the initial checkpoint, both direction types exhibit similar expansion through epoch 3 (approximately 500% growth). However, from epoch 4 onward, a clear divergence emerges, with RSV directions showing accelerated expansion. By epoch 10, RSV directions achieve 1220% growth relative to initialization, while random directions reach only 940%. This growing disparity indicates the network progressively develops anisotropic ray coverage that preferentially expands along directions aligned with the underlying data manifold.

To complement these directional and train-to-train analyses, we further examined how ray coverage evolves *within the training set*. At each checkpoint, we computed GIs between randomly selected same-class training sample pairs. Table 3 summarizes GI statistics separately for training pairs where the prediction remains correct versus incorrect.

Table 3: Summary of GI statistics across training checkpoints for correctly and incorrectly classified training sample pairs. *Mean GI* is the arithmetic mean of all GI values including those at the boundary ($\mathrm{GI} = 0$); *Avg GI* is the mean computed only over sample pairs with $\mathrm{GI} > 0$ (i.e., where a nonzero interval exists).

Epoch	Batch	Correct					Incorrect				
		Count	Mean GI	% GI > 1	Avg GI	Median GI	Count	Mean GI	% GI > 1	Avg GI	Median GI
0	0	50	1.1798	48.21	1.1638	0.41	50	1.1922	48.49	1.1816	0.49
0	200	50	1.4496	79.60	1.4708	1.38	50	1.5397	84.00	1.5635	1.48
0	399	50	1.3748	80.06	1.3850	1.21	50	1.3375	82.87	1.3553	1.28
1	399	50	1.4358	74.89	1.4523	1.23	50	1.2922	81.24	1.3065	1.30
3	200	50	1.6066	79.80	1.6202	1.36	50	1.2574	85.17	1.2656	1.20
4	200	50	1.6746	82.20	1.6863	1.49	50	1.2213	85.07	1.2286	1.16
5	399	50	1.6989	81.77	1.7175	1.54	50	1.1515	82.79	1.1615	1.13
6	399	50	1.7177	82.36	1.7352	1.56	50	1.1150	81.84	1.1250	1.11
8	200	50	1.7584	83.94	1.7737	1.61	50	1.1169	82.16	1.1260	1.14
9	200	50	1.7881	84.51	1.8039	1.65	50	1.1172	82.54	1.1245	1.15
10	399	50	1.8315	85.43	1.8474	1.72	50	1.1167	83.95	1.1219	1.14

C Directional Generalization Intervals and Jacobian SVD Analysis

Our experiment examines how Generalization Intervals (GIs) along Jacobian RSV directions evolve during training. The Singular Value Decomposition (SVD) decomposes the Jacobian matrix into orthogonal components, where singular values represent the network's sensitivity along corresponding input directions. For each model (early-trained after 1 epoch and later-trained after 50 epochs), we computed Jacobians at 50 correctly classified MNIST samples, performed SVD, and measured GIs along each RSV direction with maximum perturbation of 20.0 and step size of 0.01.

The results reveal a significant transformation in directional sensitivity. Early training (epoch 1) shows a relatively uniform GI profile ($\sim$7) across all RSV directions, whereas after extended training (epoch 50), GIs decrease along directions with large singular values (low-indexed RSVs) while dramatically increasing along directions with small singular values (high-indexed RSVs), reaching an average of $\sim$11.5. This pattern supports the theory that neural networks develop a natural split between an "information space" (directions with large singular values) where the model remains sensitive, and a "nuisance space" (directions with small singular values) where the model becomes increasingly invariant.

D Linear Approximation Analysis Along Singular Vector Directions

We analyze the local linear behavior of neural networks by measuring how far a linear approximation remains valid along different input directions. For each RSV direction from the Jacobian SVD, we compute the maximum perturbation magnitude α at which the cosine similarity between the true network output and its linear approximation remains above 0.95. This threshold represents the distance over which the linear Taylor expansion of the network function serves as a reliable proxy for the true nonlinear function.

Our analysis contrasts untrained and trained MLP models on MNIST, using 50 samples (5 per class) for reliable statistics. For each sample, we compute the Jacobian at the penultimate layer, perform SVD, and measure α values along 100 evenly distributed RSV directions. The results reveal a striking transformation in local linearity through training. The untrained model (random initialization) exhibits uniformly poor linear approximations, with maximum α ranging only from 0.2-0.4 across all directions. In contrast, the trained model develops a clear monotonic relationship: directions with higher singular values (lower RSV indices) support shorter linear approximations ($\alpha \approx 0.4$), while directions with lower singular values (higher RSV indices) maintain linearity over much greater distances ($\alpha \approx 1.2$). The transition around indices 300-400, where the singular value spectrum flattens, corresponds to where trained models achieve significantly better linearity than untrained ones.

D.1 Remark: Theorem 1 and the Residual Bound

Theorem 1 may be interpreted as requiring the network f is approximately linear, although approximate linearity is not required for the large, non-adversarial displacements involved in test-time classification. A residual bound is assumed. The theorem follows from the exact identity $f(x_0 + td) = f(x_0) + tJ_f(x_0)d + R(t)$, which holds for any differentiable f without requiring the network be linear; nonlinearity is fully absorbed into the residual $R(t)$, bounded by δ over the ray. A larger nonlinear residual requires a correspondingly larger δ, which reduces the magnitude of the resulting GI lower bound; it does not invalidate the theorem. Correspondingly, all GI and ray-coverage values reported in this paper are obtained through direct evaluation of the nonlinear network along each ray, without substituting a linearized network model.

Empirically, a first-order approximation is found to represent the network's behavior over extended distances along directions associated with the largest GIs. This relationship is examined through representation analysis reported in this section. The analysis above shows that the distance α over which a linear approximation of the penultimate-layer representation ϕ maintains cosine similarity ≥ 0.95 with the true representation grows from $\alpha \approx 0.2$–0.4 (untrained, roughly uniform across directions) to $\alpha \approx 1.2$ after training, specifically along the low-curvature directions with the largest GIs. Since $f = W\phi$ for a fixed linear classifier head W, a small residual in ϕ implies a residual in f scaled by $\|W\|_{\mathrm{op}}$ (the same step used in the proof of Theorem 2), relating these observations about ϕ transfers to the quantity Theorem 1 explicitly bounds.

E Test-to-Train Generalization Interval Experimental Protocol

Our ray coverage analysis examined whether test samples consistently fall within the GI zones of training samples. For each dataset, we computed:

$$\text{Coverage}(x_{\text{test}}) = \frac{1}{|S_p|} \sum_{x_i \in S_p} \mathbf{1}[\text{GI}(f, x_i, x_{\text{test}} - x_i) \geq 1] \tag{8}$$

where S_p is the set of training samples from the predicted class, and $\mathbf{1}[\cdot]$ is the indicator function. This measures the percentage of training samples whose ray coverage contains the test sample.

Dataset-specific protocols were as follows:

- **MNIST:** All 10,000 test samples compared against all training samples of the predicted class ($\sim$6,000 per class), using MLP, CNN, and ViT models with step size $\delta = 0.01$.
- **CIFAR-10:** 50 test samples per class compared against 200 randomly sampled training samples of the predicted class, using ResNet20 and VGG16-BN with step size $\delta = 0.01$.
- **ImageNette:** 20 test samples per class compared against 100 randomly sampled training samples of the predicted class, using VGG16, ResNet50, ViT-B, and DeiT-S with step size $\delta = 0.01$.

For computational efficiency, we implemented batch processing tailored to each dataset's dimensionality (6,000/2,000/100 sample pairs per batch for MNIST/CIFAR-10/ImageNette respectively). The consistently high coverage percentages across datasets support our Neural Expansion hypothesis, confirming that classification operates through ray coverage rather than instance-level memorization.

F Robustness and Discriminative Validity of Ray Coverage

The ray-coverage measurements used throughout the paper are evaluated with respect to training variability and class specificity. Stability across sources of training randomness is examined, and class-specific structure is assessed through comparisons involving untrained networks and rays directed toward different classes.

F.1 Discriminative Baselines: Untrained Networks and Wrong-Class Rays

A 2×2 study was conducted (untrained vs. trained checkpoint × same-class vs. different-class rays) on MLP-MNIST, CNN-MNIST, and CNN-Fashion-MNIST, using $n = 100$ test samples per architecture and the step-search protocol of Appendix A.4. For the different-class condition, the correct class is held fixed at the test sample's predicted class c, while rays are evaluated toward 3 random wrong-class training pools per test sample; this ensures a consistent comparison, since directly reusing the per-class coverage columns from Appendix E is invalid (it makes GI ≤ 1 tautological, and empirically yields exactly 0%).

Table 4: Ray coverage (%) under a 2×2 untrained/trained × same/different-class design. Training substantially increases same-class coverage, and trained same-class coverage substantially exceeds trained different-class coverage, confirming coverage is class-discriminative rather than an artifact of the measurement.

Model	Untrained, same	Untrained, diff.	Trained, same	Trained, diff.
MLP, MNIST	46.8%	30.6%	88.9%	1.2%
CNN, MNIST	35.6%	42.7%	96.3%	0.3%
CNN, Fashion-MNIST	32.3%	34.6%	88.5%	1.0%

At $p < 10^{-11}$, training substantially increases same-class coverage (32–47%→89–96%), and trained same-class coverage substantially exceeds trained different-class coverage (89–96% vs. 0.3–1.2%). The untrained same-vs-different contrast is nonsignificant on Fashion-MNIST ($p = 0.19$) but statistically significant on MNIST ($p = 3.9 \times 10^{-12}$ MLP, $p = 0.02$ CNN), indicating a mild

same-class pixel bias in untrained networks, with a substantially larger class-specific effect being observed following network training.

F.2 Statistical Robustness Across Seeds and Optimizers

The same-class ray-coverage measurement was also repeated (MNIST MLP/CNN, percentage of pairs with GI > 1, protocol as in Appendix E) across 5 random seeds and 3 optimizers (SGD, Adam, RMSprop; learning rate 0.001) for both architectures, 16 models total, recomputing coverage at several checkpoints from untrained through epoch 20 on a fixed $n = 50$ evaluation split.

Table 5: Final-epoch ray coverage (%) is nearly seed-invariant and only mildly optimizer-sensitive, whereas substantial variability is observed among untrained baselines (13–99.96% MLP, 24–95% CNN).

Architecture	Factor	n runs	Mean	Std	CV (%)	Min–max
MNIST MLP	Seed	5	88.91	0.07	0.1	88.82–89.02
MNIST MLP	Optimizer	3	91.20	1.64	1.8	89.04–93.03
MNIST CNN	Seed	5	96.86	0.11	0.1	96.74–97.04
MNIST CNN	Optimizer	3	98.52	1.22	1.2	96.79–99.43

The untrained-vs-final difference is significant ($p < 0.05$, Mann-Whitney U) in 15 of 16 runs, most at $p < 10^{-14}$; the one exception (CNN seed 2, $p = 0.061$) exhibits the same trend, with elevated coverage already observed at the untrained checkpoint for this seed (95.5%) at initialization. The untrained baseline is itself seed-dependent, while the trained endpoint converges tightly regardless of seed or optimizer (CV $\leq 1.8\%$), supporting the stability of coverage measurements used for the results in the main text.

F.3 Orthogonal Tube Test: Full Protocol and Results

A class-preserving straight-line ray from a training sample to a test sample establishes stability only along that one direction; the extent of prediction stability in the surrounding neighborhood must therefore be assessed through additional directional measurements. This property is evaluated using an orthogonal "tube" probe: for 100 previously identified rays (GI > 1) per model on a trained MNIST-CNN and a trained Fashion-MNIST-CNN, we sample 20 directions exactly orthogonal to each ray at 5 evenly spaced points along it, apply the established step-search algorithm (Appendix A.4) along directions perpendicular to the ray, and record the distance to the first class change (the *tube radius*), normalized by the ray's own hop distance $\|x_{\text{test}} - x_i\|$. CNNs are specifically selected, whose local receptive fields make an orthogonal perturbation more likely to induce an early transition than in a fully-connected network, as a challenging case for this test.

Table 6: Orthogonal tube test: orthogonal stability radius around high-coverage rays, normalized by the ray's own hop distance. A ratio ≥ 1 indicates the stable region is at least as wide as the ray is long, in every perpendicular direction tested.

Model	Median hop distance	Median tube radius	Median ratio (tube/hop)
MNIST CNN	17.9	23.1	1.39×
Fashion-MNIST CNN	16.2	15.4	1.01×

On both CNNs, the typical high-coverage ray lies within a region at least as wide as the ray itself (Table 6), across every one of the 20 perpendicular directions sampled at every one of the 5 points along the ray. Ray coverage therefore reflects an extended multidimensional same-class region, not a one-dimensional artifact of the specific train-test direction measured.

G Class Region Analysis Along Train-to-Test Paths

This section investigates how a model's class predictions evolve along linear paths from training samples to test samples of the same class. We conduct this analysis using a Multi-Layer Perceptron

(MLP) trained on the MNIST dataset. We define $\varepsilon \in [0, \varepsilon_{\max}]$ as a scalar interpolation parameter that traces a linear trajectory from a starting input x_1 to a target input x_2 in input space. Each point along the path is given by $x(\varepsilon) = x_1 + \varepsilon(x_2 - x_1)$. The generalization interval (GI) is the maximum value of ε for which the model's predicted class remains consistent with the initial prediction at x_1. We group path analyses into two categories:

GI $>$ 1.0: The test sample lies within the ray coverage of the training point. These paths typically show a single, uninterrupted class prediction (matching the true class) along the entire trajectory (see Figure 8).



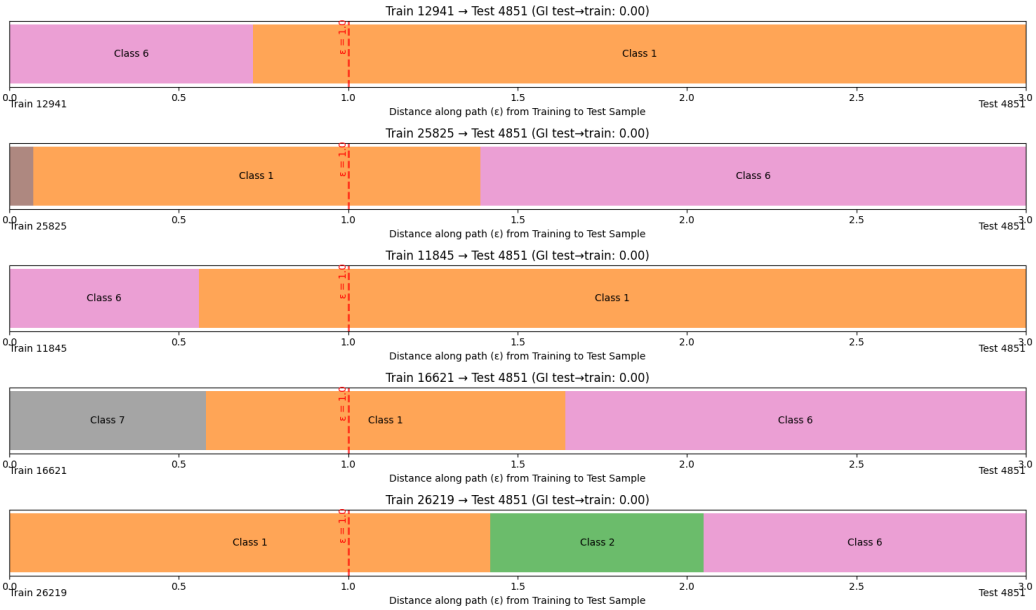
(a)



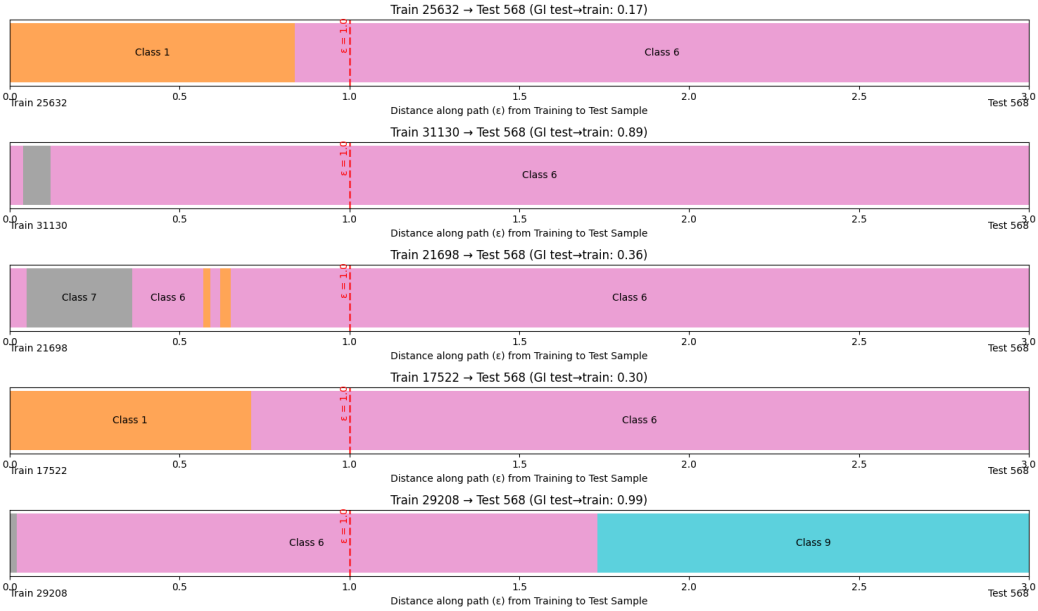
(b)

Figure 8: Predicted class trajectories along paths from training to test samples (Train $\rightarrow$ Test Class 6) with GI $>$ 1.0. Each subplot shows class-stable behavior across the entire interpolation path, indicating the test point lies within the ray coverage of the training sample.

GI < 1.0: The test sample lies outside the ray coverage region. These paths frequently exhibit multiple class transitions, indicating that the model's classification is unstable along the direction from train to test (see Figure 9).



(a)



(b)

Figure 9: Predicted class trajectories along paths from training to test samples (Train $\rightarrow$ Test Class 6) with GI < 1.0. Multiple class transitions are observed along the interpolation, indicating unstable or fragmented class regions.

Together, these visualizations highlight a key behavioral difference: when GI > 1.0, the classification is stable and coherent between the training and test sample. In contrast, GI < 1.0 is associated with fragmented decision regions and unstable class behavior, even between samples of the same true class.

H Causal Intervention Experiment: Full Protocol

H.1 Derivation of the Population Error Identity

We consider binary classification task with labels $0, 1$ and fix the training set $S = \{(x_i, y_i)\}_{i=1}^N$ sampled from the population. We denote a random sample from the population by $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$. For each sample (X, Y) from the population, we assign a sample from the training set (x_I, y_I) so that $y_I = Y$. The random index I keeps track of which GI to keep track of at sample (X, Y), since GIs from multiple samples may reach the sample (X, Y). It is just a device to make sure that there is no ambiguity about which GI we are considering, and the choice is generally interchangeable. We now define, for $B \subset \mathbb{S}^{d-1}$, where $\mathbb{S}^{d-1}$ is the $d-1$-dimensional sphere in $\mathbb{R}^d$,

$$\mu_i(B) = P\left(I = i, \quad \frac{X - x_I}{\|X - x_I\|} \in B\right) \tag{9}$$

This is the probability that a random sample X assigned to x_i points to the set of directions B when we see x_i as the origin. In other words, this tells us which directions the population is most concentrated in, when seen from x_i. We assume that the random variable $R = \|X - x_I\|_2$ admits a density conditional to the direction $U = \frac{X - x_I}{\|X - x_I\|}$, denoted $p_i(r|u)$:

$$P(I = i, U \in du, R \in dr) = p_i(r|u)dr\mu_i(du) \tag{10}$$

Now, let $f_n : \mathbb{R}^n \to \{0, 1\}$ denote the prediction model at training time n. $f_n(x)$ denotes the predicted class of x by the model. One nontrivial assumption we make is the following *no-re-entry assumption*:

$$f_n(x_I + RU) = y_I \Leftrightarrow R < \mathrm{GI}(f_n, x_I, U), \quad \text{almost surely.} \tag{11}$$

This statement says that, along the ray from the assigned sample x_I, the model predicts the correct original class y_I up to the GI boundary and not beyond. Thus, the prediction does not "re-enter" the class y_I. We note that this does not mean samples of class y_I do not occur beyond GI. In such a case, the model misclassifies these samples.

Now, we will consider the population error of the model: $\epsilon_n = P(f_n(X) \neq Y)$. We note that the statement $f_n(x_I + RU) = y_I$ is equivalent to $f_n(X) = Y$ by definitions of I, R, U, so the no-entry assumption means that

$$\epsilon_n = P(R \geq \mathrm{GI}(f_n, x_I, U)) \tag{12}$$

$$= \sum_i \int_{\mathbb{S}^{d-1}} \int_{\mathrm{GI}(f_n, x_i, u)}^{\infty} p_i(r|u)dr\mu_i(du) \tag{13}$$

Therefore, the difference in error between f_{n+1} and f_n is

$$\epsilon_{n+1} - \epsilon_n = -\sum_i \int_{\mathbb{S}^{d-1}} \int_{\mathrm{GI}(f_n, x_i, u)}^{\mathrm{GI}(f_{n+1}, x_i, u)} p_i(r|u)dr\mu_i(du) \tag{14}$$

If the GI increased, i.e., $\mathrm{GI}(f_n, x_i, u) \leq \mathrm{GI}(f_{n+1}, x_i, u)$ almost surely, then the right-hand side of (14) is nonpositive, so the error decreases (or does not change). We remark that without the no-re-entry assumption, the conclusion is generally false. For example, for the population distribution of $X \sim \mathrm{Unif}([0, 3])$ and $Y = +1$, the single training example of $(x_1, y_1) = (0, 1)$, and a training sequence of $f_0(x) = \mathbf{1}\{(x-1)(x-2) > 0\}$ and $f_1(x) = \mathbf{1}\{1 - x > 0\}$, the GI at x_1 to the positive direction is unchanged at 1, but the population error increases because the interval $[2, 3]$, which was correctly classified, turns into incorrectly classified interval, while not affecting GI because $[2, 3]$ is disconnected with the region covered by GI, i.e., $[0, 1]$.

We will complement this theoretical observation with the experiment below.

H.2 Interventional Experiment: Design

For both the MNIST MLP (784–1024–512–10, Appendix A.2) and a two-block convolutional Fashion-MNIST CNN with BatchNorm, an auxiliary loss term is added, active in two of three training arms,

that acts directly on what a GI measures: whether the prediction remains unchanged under local perturbations around a training point x. For each (x, y) we draw a random unit direction u and perturb by $\varepsilon(x) = 0.08 \times \|x\|_2$, placing the step in the 2.0–2.2 range in this normalized space, comparable to standard ℓ_2 adversarial-perturbation budgets for MNIST-scale data: a local perturbation that is applied independently of another sample location. The hinge objective uses the margin at perturbed inputs, defined as the model's output for the true class minus its highest output among the other classes:

- **Expand** arm: $\mathcal{L} = \text{CE} + \lambda \cdot \text{ReLU}(\tau - \text{margin}(x + \varepsilon u))$, encouraging confident correct classification of the perturbed input.
- **Restrict** arm: $\mathcal{L} = \text{CE} + \lambda \cdot \text{ReLU}(\text{margin}(x + \varepsilon u) + \tau)$, encouraging confident misclassification of perturbed inputs, while cross-entropy encourages correct classification at x.
- **Control** arm: standard cross-entropy, $\lambda = 0$.

We set $\tau = 2.5$ to match the mean margin measured on our own trained checkpoint (2.48 over 1000 training images), a target calibrated using observations. At every epoch we measure GI and ray coverage with our existing, unmodified step-search routine (Appendix A.4), on a fixed pool of 200 random same-class train-to-train pairs, independent of the training seed and identical across arms. Each configuration is trained with 3 seeds, reported both individually and averaged.

H.3 Interventional Experiment: Full Results

Table 1 in the main text reports the final-epoch (epoch 10) results averaged over 3 seeds. Individually, restricting ray coverage on MNIST-MLP lowers test accuracy to 74.7%, 76.5%, and 74.3% across the 3 seeds ($p = 0.003$ vs. control), while expansion yields consistently higher accuracy across seeds (std 0.08, $p < 10^{-4}$). On Fashion-MNIST-CNN, restriction produces a statistically significant reduction (-7.4pp accuracy, -19.5pp ray coverage, $p \approx 0.008$ on both); expanding increases test accuracy by 1.1pp ($p = 0.008$), with a smaller corresponding increase in ray coverage on this architecture. In both datasets, GI and ray coverage separate in the intended direction from epoch 1 onward, on every individual seed, with no overlap between arms. The reported MNIST-MLP restrict arm's train accuracy (73.9%) is nearly as low as its test accuracy (75.2%), indicating the effect is a slower formation of classification structure overall, rather than merely a widened train/test gap.

I Ray Coverage in Challenging Cases

I.1 Random Label Experiment Protocol

We evaluate ray coverage in two randomized settings:

- **RANDOM-10**: Training labels are reassigned within 10 classes.
- **RANDOM-100**: Both training and test labels are reassigned across 100 classes.

Despite the lack of semantic label structure, ray coverage remains active in both settings. Using an MLP architecture (784-1024-512-n, where n is 10 or 100) with a GI step size of $\delta = 0.01$, we evaluated all 10,000 test samples against all training samples of the predicted class.

Table 7: Ray coverage percentages for correctly and incorrectly classified test samples under random label supervision.

Setting	Correctly Classified	Misclassified
RANDOM-10 (MNIST-MLP)	17%	24%
RANDOM-100 (MNIST-MLP)	18%	22%

Although label randomization reduced ray coverage (17%-24% on average) as shown in Table 7, it remained the classification mechanism, in contrast to the higher coverage seen with semantically meaningful labels.

I.2 Long-tailed Distribution Analysis Details

We analyze neural network behavior under two long-tailed MNIST variants: **MNIST-LT1** and **MNIST-LT2**, which differ in the severity of class imbalance (6.5:1 and 78.6:1, respectively) (see details in Table 8). In **MNIST-LT2**, which has a more extreme imbalance, we observe systematic migration of misclassified samples. As shown in Figure 4(b), samples from low-frequency classes tend to be absorbed into the ray coverage of higher-frequency classes.

Despite the imbalance, ray coverage remains strong across all class groups, as shown in Figure 4(c), but the ray coverage from high-frequency classes appears to dominate test-time decisions. In **MNIST-LT1**, misclassified samples also showed similar migration patterns: low-frequency classes tended to migrate to medium (47.2%) and high (35.2%) frequencies, with only 17.6% staying within the low-frequency group. In both settings, the GI coverage remained consistently high (86.6%-94.3%) across all frequency combinations, confirming that **Neural Expansion** operates reliably across varying imbalance conditions.

Table 8: Class distributions in MNIST long-tailed variants. H=High, M=Medium, L=Low frequency bands based on 33rd and 66th percentiles.

Class	0	1	2	3	4	5	6	7	8	9	**Ratio**
LT1	4717	2866	2370	2015	1806	1590	1346	1034	886	730	6.5:1
Band	H	H	H	M	M	M	M	L	L	L	
LT2	4716	2504	2026	1464	1096	761	383	218	93	60	78.6:1
Band	H	H	H	M	M	M	L	L	L	L	

I.3 Adversarial Examples Experiment Protocol

We conducted a systematic analysis of how adversarial examples interact with ray coverage across three datasets and architectures:

- **MNIST-MLP**: Custom MLP with architecture 784-1024-512-10
- **CIFAR10-ResNet20**: Pre-trained ResNet20 from PyTorch hub
- **ImageNette-ViTB**: Pre-trained Vision Transformer Base (patch size 16)

For each configuration, we generated adversarial examples using three established attack methods:

- **Carlini-Wagner (CW) (Carlini and Wagner, 2017)**: Optimization-based attack that minimizes perturbation magnitude with parameters $c = 1e$-2, learning rate $1e$-2, and 1000 iterations.
- **Iterative Fast Gradient Sign Method (IFGSM) (Kurakin et al., 2017)**: Iterative gradient-based attack with $\epsilon = 0.03$, $\alpha = 0.005$, and 10 iterations.
- **Projected Gradient Descent (PGD) (Madry et al., 2018)**: Extension of IFGSM with random initialization, using $\epsilon = 0.03$, $\alpha = 0.005$, and 30 iterations.

Our dataset-specific protocols involved:

- **MNIST**: 20 samples/class (total 200), compared against all training samples of predicted class
- **CIFAR-10**: 10 samples/class (total 100), compared against 200 training samples of predicted class
- **ImageNette**: 5 samples/class (total 50), compared against 100 training samples of predicted class

- **MNIST-MLP:** Correctly classified adversarial examples show 83.9–86.6% coverage with their predicted class, while misclassified examples show 91.2–94.5%
- **CIFAR10-ResNet20:** Correctly classified adversarial examples have moderate coverage (63.9–68.7%), and misclassified examples vary between 39.6–58.1% depending on attack strength

- **ImageNette-ViT-B:** Correctly classified adversarial examples exhibit low ray coverage (15.8–19.6%). For misclassified examples, coverage could not be computed reliably due to class predictions falling outside the 10-class subset and into the full 1000-class ImageNet space

Adversarial Alignment Analysis: We computed the Jacobian SVD at each source image and measured the cumulative projection of both adversarial gradients and image-to-image difference vectors onto the right singular vectors (see Figure 10). Loss gradients (blue lines) concentrate heavily in the first few singular vector directions, often reaching 80-95% cumulative projection within the top 5 directions. In contrast, natural image difference vectors (orange lines) show a more gradual accumulation, typically reaching only 20-40% in the top 5 directions. This contrast explains why adversarial perturbations can be much smaller than natural image differences: they selectively target the model's most sensitive directions, precisely those with the shortest generalization intervals.

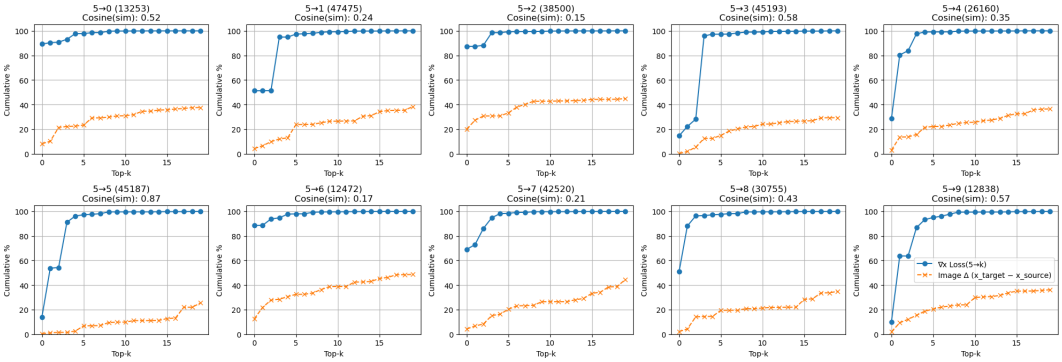


Figure 10: Gradient & Image Δ Projections on Top-k Right Singular Vectors. Blue lines show the cumulative percentage of the loss gradient projected onto the top-k RSVs. Orange lines show the projection of the image difference vector between target and source images.

Our sample counts across dataset/model/attack combinations were: MNIST-MLP: CW (132 correct, 68 wrong), IFGSM (81 correct, 19 wrong), PGD (74 correct, 26 wrong); CIFAR10-ResNet20: CW (37 correct, 63 wrong), IFGSM (25 correct, 75 wrong), PGD (35 correct, 65 wrong); ImageNette-ViTB: CW (53 correct), IFGSM (47 correct), PGD (42 correct).

I.4 Out-of-Distribution Sample Analysis

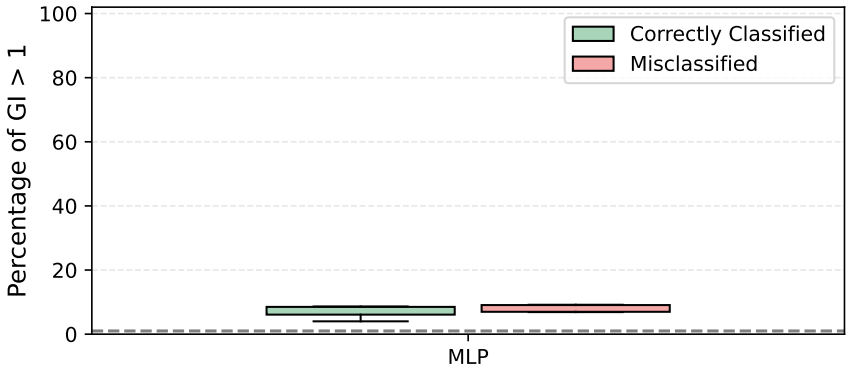


Figure 11: Box plot illustrating the GI ray coverage for out-of-distribution (OOD) samples from CIFAR-10, evaluated on an MLP trained on MNIST. The gray dashed line (as defined in Figure 3) indicates the 1% directional coverage threshold.

To assess whether out-of-distribution (OOD) inputs fall within learned class-preserving regions, we compute GI ray coverage for CIFAR-10 test samples evaluated on an MLP trained on MNIST. Since the model expects 28×28 grayscale inputs, CIFAR-10 images are first transformed to grayscale 28×28 images. We follow the MNIST evaluation protocol described in Appendix E, using 50 test

samples per CIFAR-10 class. Despite being OOD, CIFAR-10 samples exhibit nontrivial GI coverage of $\sim$10%, showing that the model's learned ray coverage regions extend into OOD input space. We note that since CIFAR-10 object classes have no semantic correspondence to MNIST digit classes, the "correct/incorrect" stratification here reflects only whether the predicted digit index happens to match the CIFAR-10 class index, and we report aggregate coverage values accordingly.

Potential extensions of this analysis include evaluation under distribution shift (e.g., ImageNet→ImageNet-Sketch/-R), self-supervised representations (e.g., DINO or CLIP linear probes), and next-token prediction in language models, where the same GI and ray-coverage formalism applies directly to context-level prediction stability.

J Proof of Theorem 1

By definition of the linear approximation, we have $f(x_0+\epsilon d) = f(x_0)+\epsilon J_f(x_0)d+R$. In particular, for each component, we have $f^i(x_0+\epsilon d) = f^i(x_0) + \epsilon[J_f(x_0)d]^i + R^i$. If $\epsilon = \text{GI}(f, x_0, d)$, then since we are at the decision boundary, we must have $f^y(x_0+\epsilon d) = f^i(x_0+\epsilon d)$ where y is the class of x_0 and i is another class. Substituting the linear approximation, we obtain

$$f^y(x_0) + \epsilon[J_f(x_0)d]^y + R^y = f^i(x_0) + \epsilon[J_f(x_0)d]^i + R^i \tag{15}$$

Reorganizing the equation, we obtain

$$\epsilon([J_f(x_0)d]^y - [J_f(x_0)d]^i) = f^i(x_0) - f^y(x_0) + R^i - R^y \tag{16}$$

By the inhomogeneous assumption, $[J_f(x_0)d]^y - [J_f(x_0)d]^i \neq 0$ and thus

$$\epsilon = \frac{f^i(x_0) - f^y(x_0) + R^i - R^y}{[J_f(x_0)d]^y - [J_f(x_0)d]^i} \tag{17}$$

By the triangle inequality,

$$|\epsilon| = \epsilon \geq \frac{|f^i(x_0) - f^y(x_0)| - |R^i - R^y|}{|[J_f(x_0)d]^y| + |[J_f(x_0)d]^i|} \tag{18}$$

Note that, by the triangle inequality again,

$$|R^i - R^y| \leq \sum_{j=1}^{c} |R^j| = \|R\|_1 \tag{19}$$

and similarly $|[J_f(x_0)d]^y| + |[J_f(x_0)d]^i| \leq \|J_f(x_0)d\|_1$. Thus we have

$$\epsilon \geq \frac{|f^i(x_0) - f^y(x_0)| - \|R\|_1}{\|J_f(x_0)d\|_1} \tag{20}$$

We note that we must have $f^y(x_0) \geq f^i(x_0)$ so that x_0 is classified as y. Thus

$$\epsilon \geq \frac{f^y(x_0) - f^i(x_0) - \|R\|_1}{\|J_f(x_0)d\|_1} \tag{21}$$

Here $i = i_*$ denotes the particular wrong class whose logit ties f^y at the point $x_0 + \epsilon d$ where the decision boundary is first reached. Since $i_* \neq y$ is one of the $c-1$ wrong classes, we have $f^{i_*}(x_0) \leq \max_{i\neq y} f^i(x_0)$, and therefore

$$f^y(x_0) - f^{i_*}(x_0) \;\geq\; f^y(x_0) - \max_{i\neq y} f^i(x_0) \;=\; q_0. \tag{22}$$

Since the margin is $q_0 = f^y(x_0) - \max_{i\neq y} f^i(x_0)$ and also $\|R\|_1 < \delta$, we have

$$\epsilon \geq \frac{q_0 - \delta}{\|J_f(x_0)d\|_1} \tag{23}$$

K Proof of Theorem 2

We establish the bound $\text{GI}(f, x_0, v_i) \geq \frac{q_0-\delta}{\sqrt{c}\,\|W\|_{\text{op}}\sigma_i}$ by starting from Theorem 1 and evaluating its denominator explicitly at $d = v_i$ using the SVD structure of $J_\phi(x_0)$.

Step 1: Apply Theorem 1 at $d = v_i$. Setting $d = v_i$ (a unit vector by definition of the SVD) in Theorem 1:

$$\mathrm{GI}(f, x_0, v_i) \;\geq\; \frac{q_0 - \delta}{\|J_f(x_0)\, v_i\|_1} \tag{24}$$

It suffices to upper-bound $\|J_f(x_0)\, v_i\|_1$, since a larger denominator yields a smaller (still valid) lower bound on GI.

Step 2: Convert ℓ^1 to ℓ^2 norm. For any $z \in \mathbb{R}^c$, the Cauchy–Schwarz inequality gives $\|z\|_1 = \langle |z|, \mathbf{1} \rangle \leq \|z\|_2\, \|\mathbf{1}\|_2 = \sqrt{c}\, \|z\|_2$. Applying this:

$$\|J_f(x_0)\, v_i\|_1 \;\leq\; \sqrt{c}\, \|J_f(x_0)\, v_i\|_2 \tag{25}$$

Step 3: Apply the chain rule. Since $f(x) = W\phi(x)$, the chain rule gives $J_f(x_0) = W J_\phi(x_0)$, so:

$$\|J_f(x_0)\, v_i\|_2 \;=\; \|W\, J_\phi(x_0)\, v_i\|_2 \tag{26}$$

Step 4: Use the SVD identity $J_\phi(x_0)\, v_i = \sigma_i u_i$. By definition of the SVD $J_\phi(x_0) = U \Sigma V^\top$, the right singular vectors are mapped as $J_\phi(x_0)\, v_i = \sigma_i\, u_i$ where $u_i \in \mathbb{R}^m$ is the i-th left singular vector with $\|u_i\|_2 = 1$. Substituting:

$$\|W\, J_\phi(x_0)\, v_i\|_2 \;=\; \|W\, \sigma_i\, u_i\|_2 \;=\; \sigma_i\, \|W\, u_i\|_2 \tag{27}$$

Step 5: Bound by the operator norm. By definition, $\|W\|_{\mathrm{op}} = \max_{\|v\|=1} \|Wv\|_2$. Since $\|u_i\|_2 = 1$:

$$\|W\, u_i\|_2 \;\leq\; \|W\|_{\mathrm{op}} \tag{28}$$

Step 6: Chain the bounds and conclude. Combining Steps 2–5:

$$\|J_f(x_0)\, v_i\|_1 \;\leq\; \sqrt{c}\, \sigma_i\, \|W\|_{\mathrm{op}} \tag{29}$$

Substituting into Step 1:

$$\mathrm{GI}(f, x_0, v_i) \;\geq\; \frac{q_0 - \delta}{\sqrt{c}\, \|W\|_{\mathrm{op}}\, \sigma_i} \tag{30}$$

□

L Proof of Theorem 3

We prove that the input-space loss gradient is bounded by $\sqrt{2}\|W\|_{\mathrm{op}} \cdot \sigma_1$.

Step 1: Express the input gradient via the chain rule. The cross-entropy loss is $\mathcal{L}(x) = -\log \sigma_y(f(x))$, where σ_y denotes the softmax probability of the true class y. Differentiating with respect to the input x via the chain rule:

$$\nabla_x \mathcal{L}(x_0) \;=\; J_f(x_0)^\top\, g \tag{31}$$

where $g \in \mathbb{R}^c$ is the gradient of the loss with respect to the logits $f(x_0)$, with entries $g_i = \sigma_i(x_0) - \mathbf{1}[i = y]$.

Step 2: Bound $\|g\|_2$. Since $g_y = \sigma_y(x_0) - 1$ and $g_i = \sigma_i(x_0)$ for $i \neq y$, with $\sigma_i(x_0) \in [0, 1]$ and $\sum_i \sigma_i = 1$, the squared norm satisfies $\|g\|_2^2 = (\sigma_y - 1)^2 + \sum_{i \neq y} \sigma_i^2 \leq (\sigma_y - 1)^2 + (1 - \sigma_y)^2 = 2(1 - \sigma_y)^2 \leq 2$, so $\|g\|_2 \leq \sqrt{2}$.

Step 3: Bound the input gradient norm. Taking the ℓ^2 norm and applying submultiplicativity, then using the chain rule decomposition $J_f(x_0) = W J_\phi(x_0)$ and $\|J_\phi(x_0)\|_2 = \sigma_1$:

$$\|\nabla_x \mathcal{L}(x_0)\|_2 \;\leq\; \|W\|_{\mathrm{op}} \cdot \sigma_1 \cdot \sqrt{2} \;=\; \sqrt{2}\, \|W\|_{\mathrm{op}} \cdot \sigma_1 \tag{32}$$

This establishes Equation (5). □

M Proof of Theorem 4

We prove two claims: (1) any successful adversarial perturbation Δ must satisfy $\|\Delta\|_2 \geq \min_{\|d\|=1} \mathrm{GI}(f, x_0, d)$, and (2) the lower bound $\frac{q_0-\delta}{\sqrt{c}\,\|W\|_{\mathrm{op}}\sigma_1}$.

Step 1: Decompose Δ into direction and magnitude. Any nonzero perturbation $\Delta \in \mathbb{R}^d$ can be written as $\Delta = \|\Delta\|_2 \cdot d_\Delta$ where $d_\Delta = \Delta/\|\Delta\|_2$ and $\|d_\Delta\| = 1$.

Step 2: Derive the lower bound by contradiction. Suppose the perturbation Δ successfully changes the classification: $\arg\max_i f^i(x_0 + \Delta) \neq y$. Suppose for contradiction that $\|\Delta\|_2 < \mathrm{GI}(f, x_0, d_\Delta)$. By definition of GI, moving distance $\|\Delta\|_2$ in direction d_Δ does not change the predicted class, which contradicts the assumption that the attack succeeds. Therefore:

$$\|\Delta\|_2 \;\geq\; \mathrm{GI}(f, x_0, d_\Delta) \tag{33}$$

Step 3: Optimize over the adversary's choice of direction. Since the bound holds for any direction d_Δ, the tightest constraint the adversary faces is:

$$\|\Delta\|_2 \;\geq\; \min_{\|d\|=1} \mathrm{GI}(f, x_0, d) \tag{34}$$

This establishes Equation (6).

Step 4: Apply Theorem 1. Observe that for any unit d, by Cauchy-Schwarz and the definition of operator norm,

$$\|J_f(x_0)d\|_1 \leq \sqrt{c}\|W\|_{\mathrm{op}}\|J_\phi d\|_2 \leq \sqrt{c}\|W\|_{\mathrm{op}}\sigma_1 \tag{35}$$

Then, applying this bound to Theorem1, we obtain the desired bound.

□

N Understanding GI Growth During Training

Our experiments in Section 5.1 (and specifically Figure 2a, which compares epoch 1 to epoch 50) show consistently that GIs expand during training, and that this expansion is anisotropic: GIs grow most along high-index RSV directions (small σ_i) and shrink along low-index RSV directions (large σ_i). Here we provide a unified understanding of why this happens, grounded in two mechanisms that our experiments and prior work support.

The GI lower bound from Theorem 2 is:

$$B_i(t) \;=\; \frac{q_0(t) - \delta}{\sqrt{c}\,\|W(t)\|_{\mathrm{op}}\,\sigma_i(t)}$$

This bound grows when the numerator (margin) increases and when the denominator (sensitivity) decreases. We identify two mechanisms driving this:

(i) Margin growth. Lyu and Li (2020) prove that for homogeneous networks trained with cross-entropy loss, a smooth approximation of the normalized margin increases monotonically during training. This supports the numerator $q_0(t)$ being non-decreasing, consistent with what we observe empirically.

(ii) Anisotropic restructuring of Jacobian singular values. As seen in Figure 2(b), training does not uniformly shrink all singular values. Instead, it *sharpens the anisotropy*: high-index singular values (which are already small) shrink further, while low-index singular values (which are already large) may grow.

This observation provides a finer-grained directional decomposition of the aggregate findings by Novak et al. (2018). They report that the overall Jacobian Frobenius norm, $\|J\|_F = \sqrt{\sum_i \sigma_i^2}$, decreases during training. Because there are many more high-index (small σ_i) directions than low-index ones, the net effect on $\|J\|_F$ is still a decrease, even while a few low-index σ_i grow. It is also important to note a structural distinction: while Novak et al. (2018) measure the input-output

Jacobian $J_f = \frac{\partial f(x)}{\partial x}$, our analysis relies on the penultimate layer Jacobian $J_\phi = \frac{\partial \phi(x)}{\partial x}$. These are related by $J_f = W J_\phi$, meaning their singular value spectra are connected but not identical. Our empirical results directly measure J_ϕ, making them specifically relevant to our framework.

The consequence of this restructuring is that $B_i(t)$ grows fastest in directions where σ_i is already small (high-index RSVs), because $\partial B_i / \partial \sigma_i \propto -1/\sigma_i^2$ is largest in magnitude there. This directly explains the skewed GI distribution seen in Figure 2(a): after training, GIs are disproportionately large along high-index RSV directions. We note that a fully rigorous proof of individual σ_i decay during training remains an open theoretical question; our account here is grounded in our own empirical observations of J_ϕ and the aggregate sensitivity results of Novak et al. (2018).

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: The abstract and introduction clearly state the claims made, their contributions, assumptions, and limitations. In particular, the abstract uses careful language ("helps explain a large fraction") rather than overclaiming.

   Guidelines:

   - The answer NA 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 NA 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: Limitations are clearly discussed in the Discussion section (Section 6), including computational cost, restriction to classification tasks, and the open theoretical question of individual singular value decay during training.

   Guidelines:

   - The answer NA 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.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

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: The assumptions of all four theorems are provided clearly, and the proofs are provided in Appendices J–M. Each theorem is numbered and cross-referenced throughout the paper.

   Guidelines:

   - The answer NA 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: The paper discloses all the information required to reproduce the experimental results, including datasets, model architectures, training hyperparameters, and GI computation algorithms, provided in Appendix A.

   Guidelines:

   - The answer NA 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.
   - 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.

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: The code to reproduce all experiments, including model training, Generalization Interval and ray coverage computation, the Jacobian SVD analysis, the causal intervention experiments, and the adversarial and out-of-distribution analyses, is publicly available at `https://github.com/meetchashi/neural_expansion_neurips_2026`, together with instructions for running each experiment. All datasets (MNIST, Fashion-MNIST, CIFAR-10, ImageNette) are public and can be obtained from their standard sources. The random-label and long-tailed variants are generated from these datasets by scripts in the repository, and the pretrained models we use (ResNet-20, VGG-16, ViT-B/16, DeiT-S) are from PyTorch Hub and timm.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the training and test details necessary to understand the results in Appendix A, including dataset preprocessing, model architectures, training hyperparameters, and GI computation parameters.

Guidelines:

- The answer NA 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: The experiments that support the main claims report variability and significance tests. The causal intervention results (Table 1, Section 5.2, Appendix H.3) report mean $\pm$ standard deviation over three random seeds, together with p-values for the control-vs-expand and control-vs-restrict comparisons. The robustness of ray coverage is assessed over 5 random seeds and 3 optimizers (Appendix F.2, Table 5), with standard deviation, coefficient of variation, and Mann-Whitney U tests against untrained baselines. The discriminative baselines in Appendix F.1 (Table 4) also report p-values. The reported variability reflects random initialization and data ordering across seeds, and, in Table 5, the choice of optimizer. Some descriptive analyses (e.g., Figures 2, 3 and 5 and Table 3) are computed on a single trained model per setting, with the spread across test samples shown through box plots rather than error bars across training runs.

   Guidelines:

   - The answer NA 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 (e.g., 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.).

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: [Yes]

   Justification: The paper includes the type of GPU used, memory usage, and total runtime of GI calculation per model in Table 2. This information enables reproducibility and offers insight into the computational cost of our framework.

   Guidelines:

   - The answer NA 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.

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. This is foundational theoretical and empirical work on understanding neural network generalization.

Guidelines:

- The answer NA 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: This is foundational theoretical research on understanding neural network generalization mechanisms. There is no direct path to societal impact, positive or negative, beyond general advances in understanding deep learning.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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, fairness considerations, privacy considerations, and security considerations.

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., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks. We evaluate on standard benchmark datasets (MNIST, CIFAR-10, ImageNette) and release an analysis framework, not a deployable model with misuse potential.

    Guidelines:

    - The answer NA 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.

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: The paper uses existing models and datasets and provides the necessary citations and licenses in Appendix A.1. Licenses: MIT (CIFAR-10), CC BY-SA 3.0 (MNIST), Apache 2.0 (ImageNette, ViT).

    Guidelines:

    - The answer NA 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.

13. **New assets**

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

Answer: [Yes]

Justification: The paper does not release new datasets or pretrained models. The new asset is the code implementing our Generalization Interval (GI) and ray coverage analysis framework, released at `https://github.com/meetchashi/neural_expansion_neurips_2026`. The repository includes a README with environment requirements, instructions for running each experiment, and a description of the scripts and their configuration options.

Guidelines:

- The answer NA 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.
- 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 NA means that the paper does not involve crowdsourcing nor research with human subjects.

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 NA means that the paper does not involve crowdsourcing nor research with human subjects.

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 rigorousness, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as an important, original, or non-standard component of the core methods of this research. Any LLM usage was limited to writing assistance only and did not impact the scientific content, proofs, or experimental design.

Guidelines:

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