Does Sparse Connectivity Improve Generalization? Convolutional Networks Below the Edge of Stability

Tongtong Liang
UC San Diego
ttliang@ucsd.edu

Esha Singh
UC San Diego
e3singh@ucsd.edu

Rahul Parhi
UC San Diego
rahul@ucsd.edu

Alexander Cloninger
UC San Diego
acloninger@ucsd.edu

Yu-Xiang Wang
UC San Diego
yuxiangw@ucsd.edu

Abstract

Gradient descent on overparameterized neural networks typically operates at the Edge of Stability (EoS), where the largest Hessian eigenvalue hovers around a step-size-dependent threshold. We study how sparse connectivity changes generalization below this threshold in two-layer ReLU networks. Prior results have shown that for fully-connected networks (FCNs), generalization guarantees in this regime degrade and become vacuous on high-dimensional spherical inputs. Our analysis reveals that sparse connectivity fundamentally alters this picture. Under sparse connectivity, the network processes a collection of low-dimensional patches, so the effective constraint imposed by the stability condition is governed by the geometry of the training patch collection. We prove that when the receptive fields are small relative to the ambient dimension, the effective constraint yields non-vacuous generalization bounds in precisely the spherical regime where FCNs provably fail. For independently sampled spherical patches, we prove a complementary lower bound by constructing bounded BEoS predictors with zero training error and constant population error. We corroborate this theory by analyzing the patch geometry of natural images, showing that standard convolutional designs produce patch multisets with low-dimensional structure that facilitates generalization. These results connect architecture, data geometry, and gradient descent to generalization.

1 Introduction

Convolutional neural networks (CNNs) [Fukushima, 1988, LeCun et al., 1998] are among the most successful neural architectures. The renaissance of CNNs in computer vision tasks [Krizhevsky et al., 2012, Ronneberger et al., 2015, He et al., 2016] effectively ignited the modern deep learning revolution. CNNs remain a dominant choice in image and video generative models today [see, e.g., Lai et al., 2025, and the references therein].

Modern CNN models are *overparameterized* and often have more parameters than the number of data points to fit them. It was shown that, despite the ability to fit pure noise perfectly, overparameterized CNNs trained with gradient descent tend to generalize well without explicit regularization, much more so than their densely connected counterparts [Zhang et al., 2017, Arpit et al., 2017]. It remains a mystery how the CNN architecture interacts with the implicit regularization of gradient descent that ends up finding generalizing solutions.

A tractable way to study the implicit regularization of gradient descent is to focus on the properties shared by solutions that GD can practically reach and maintain. Empirical work shows that gradient

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

descent on neural networks typically enters the Edge-of-Stability (EoS) regime: the largest eigenvalue of the training loss Hessian increases, and then hovers around a stability threshold controlled by the step size [Cohen et al., 2021]. Theoretical analyses of minima stability prove that solutions that GD can stably converge to are inherently constrained by this threshold [Wu et al., 2018, Nar and Sastry, 2018, Mulayoff et al., 2021, Nacson et al., 2023, Damian et al., 2023]. We refer to such solutions as *below the edge of stability* (BEoS), following Qiao et al. [2024], Liang et al. [2026], and take the BEoS condition as the analytical setup for studying the implicit regularization of GD.

Recent work has analyzed this problem for two-layer ReLU fully connected networks (FCNs), establishing that generalization below the edge of stability depends on the geometry of ambient input vectors [Mulayoff et al., 2021, Nacson et al., 2023, Qiao et al., 2024, Liang et al., 2025, 2026]. In particular, Liang et al. [2025, 2026] show that as the input distribution concentrates closer to a sphere, the resulting generalization guarantees eventually become vacuous.

Standard normalization preprocessing often maps images near a hypersphere, yet CNNs trained on such data generalize well in this regime. The discrepancy points to the architectural inductive bias. A convolutional layer departs from a fully-connected one in that each neuron only receives input from a small subset of coordinates, a property we term *sparse connectivity*. On the forward side, the network no longer operates on full input vectors but on a collection of lower-dimensional *patches* induced by the sparse connectivity, and thus on the backward side, the gradient dynamics couple to the geometry of the patch collection rather than the ambient point cloud. Our central question is *how sparse connectivity shapes the BEoS class and its generalization.*

To isolate this effect, we study a minimal two-layer ReLU network where each hidden neuron receives input from a prescribed set of coordinate projections, and weights are shared across patches. This architecture captures sparse connectivity while remaining analytically tractable.

Contributions.

1. We derive a step-size-dependent bound on the weighted path norm of sparsely connected two-layer networks under the BEoS condition (Theorem 4.1). The weights encode the geometry of the induced *patch multiset*, linking stability to architecture.
2. We establish generalization upper and lower bounds that reveal the role of patch geometry. For uniformly distributed ambient spherical inputs, small receptive fields yield a generalization bound scaling as $n^{-1/6+O(m/d)}$, where m is the patch size and d the ambient dimension (Theorem 4.2). For independent spherical patches, bounded BEoS predictors attain zero training error and constant excess risk at every sample size (Theorem 4.3).
3. We show empirically that convolutional patches from CIFAR-10 concentrate in a low-dimensional subspace and are resistant to isolation by hyperplanes (Section 5). These properties link natural-image geometry to stability-induced complexity control.

Technical novelty. Our analysis expresses each neuron's contribution to the loss Hessian through its activated patches, connecting the curvature constraint to the geometry of the full patch multiset. This extends the stability-based analysis of FCNs [Mulayoff et al., 2021, Nacson et al., 2023, Qiao et al., 2024, Liang et al., 2025, 2026] to a regularizer whose weights encode the network's receptive fields.

Scope of analysis. We analyze the sharpness constraint that gradient descent with a finite step size must satisfy to remain stable. To make this constraint analytically tractable, we work with two-layer ReLU networks. In particular, we focus on the design of sparse connectivity, which is a generic structure appearing in nearly every block of modern vision backbones (e.g. CNNs and ViTs). To expose its role without confounding extra architectural assumptions, we pair it with idealized data distributions that create a sharp contrast while keeping the analysis feasible. Extending this analysis to deep networks requires understanding how stability constraints propagate through successive layers. In that setting, each layer's output forms the input geometry for the next, and the geometry that sparse connectivity processes is no longer static but an evolving representation, shaped by attention, normalization, and residual connections during training. How the gradient dynamics couples to this evolving representational hierarchy requires substantial hard work, which we leave as a future work.

2 Related Work

Patch-based representation learning. Early unsupervised image representation methods decomposed images into local patches and learned sparse dictionaries or manifolds from these primitives Aharon et al. [2006], Peyré [2009]. Patch based representations have also shown success in unsupervised learning Paulin et al. [2017] and as preprocessing for image classification Coates et al. [2011], Thiry et al. [2021], Brutzkus et al. [2022]. These results highlight that patch space is highly structured, but it remains open how neural network training exploits such structure. Our work helps bridge this gap by analyzing generalization below the edge of stability.

Geometric awareness in deep learning. Recent theoretical work has explored how architectural design incorporates geometric awareness to improve neural network training and approximation. Approximation theory research has demonstrated that CNN architectures overcome the curse of dimensionality by exploiting compositional structure in natural images Mhaskar et al. [2017], Zhou [2020], Poggio [2022], Poggio and Fraser [2024], with Mao et al. [2021] formally proving the superiority of CNNs over FCNs for learning certain composite functions. More recently, Vision Transformers Dosovitskiy et al. [2021] have been analyzed through the same lens: Shi et al. [2025] extend these approximation guarantees to ViTs, showing they outperform FCNs for compositional functions, and Trockman and Kolter [2022] identify patch extraction, rather than attention, as the critical component, further reinforcing the strength of sparsely connected architectures. These analyses exploit compositional labels or explicit regularization. We study how data geometry shapes implicit regularization in overparameterized networks with general labels.

Theoretical analysis of CNNs and separation from FCNs. The advantages of CNNs over FCNs have been established from both approximation-theoretic and statistical perspectives. Approximation theory shows that CNNs with appropriate sparse weights can achieve near-minimax optimal sample complexity Oono and Suzuki [2019] and adapt to intrinsic dimension when data lie on a low-dimensional manifold Liu et al. [2021a], with Zhang et al. [2024] extending these results to overparameterized regimes under weight decay. However, these analyses rely on reducing CNNs to fully connected networks Yarotsky [2017] and therefore do not reveal the architectural insights specific to CNNs that we describe in this paper. Specifically, we do not require the data to be supported on a low-dimensional manifold to avoid the curse of dimensionality, nor do we require explicit regularization such as weight decay or a sparsity constraint. On the statistical side, a sample complexity separating CNNs, locally connected networks without weight sharing, and FCNs has been established [Li et al., 2021, Wang and Wu, 2023, Lahoti et al., 2024]. Our results are for the same model family, but differ in that we consider an overparameterized regime without explicit regularization, which allows us to prove a stronger separation between FCNs and CNNs. Our analysis identifies input geometry as a driver of this separation.

Table 1: **From complexity conditions to generalization.** Galanti et al. prescribe a filter-norm budget, while Muthukumar–Sulam use measured activation sparsity and other predictor statistics. In the lower block, GD stability (BEoS) implies a weighted path-norm constraint. Our analysis derives its weights from the architecture-induced patch geometry (Theorem 4.1), then bounds generalization over the BEoS class. The FCN lower bounds expose a curse of dimensionality on a ball and remain constant on a sphere. The final row shows how sparse connectivity yields a decaying upper bound on the sphere, with exponent approaching $1/6$ for fixed patch size m and growing ambient dimension d. See Appendix H for further comparisons.

Work	Architecture	Starting point	Guarantee
[Galanti et al., 2023]	Deep sparse nets / CNNs	Prescribed product of local filter norms	Bound in terms of filter norms, connectivity, and input energy
[Muthukumar and Sulam, 2023]	Deep ReLU FCNs	Measured inactive sets, margin, PAC-Bayes KL	Sparsity-aware bound for a trained predictor
[Liang et al., 2025]	FCN with bias	BEoS; uniform ball	$\tilde{O}(n^{-1/(2d+4)})$ upper, $\Omega(n^{-2/(d+1)})$ lower
[Liang et al., 2026]	FCN with bias	BEoS; uniform sphere	$\Omega(1)$ lower bound
Ours	**SCN with bias**	**BEoS; uniform sphere**	$\tilde{O}\left(n^{-\frac{(d-m)(d+3)}{2(3d^2-md+3d-3m)}}\right)$

Implicit bias of gradient descent. Many existing work on the implicit bias of gradient descent relies on strong assumptions on the data distributions (e.g., linearly separable data), simplified model architectures (e.g., linear activation) to make a gradient dynamics analysis tractable [Soudry et al., 2018, Gunasekar et al., 2018a,b], or weight initialization schemes that keep the model in the kernel regimes [Jacot et al., 2018, Arora et al., 2019]. While CNNs were studied [Gunasekar et al., 2018b, Arora et al., 2019], the nature of the results are different from ours.

Sparsity, stability, and generalization. Sparse-network bounds quantify generalization through filter norms [Galanti et al., 2023] or activation sparsity [Muthukumar and Sulam, 2023]. Stability analyses derive complexity control from loss curvature [Ding et al., 2024, Mulayoff et al., 2021, Nacson et al., 2023, Wu and Su, 2023, Qiao et al., 2024, Liang et al., 2025, 2026]. Table 1 places our result in this line: sparse connectivity yields a vanishing bound on the same spherical inputs for which FCNs admit a constant lower bound.

3 Preliminaries and Notations

Throughout the paper, we use $O(\cdot)$ and $\Omega(\cdot)$ to absorb constants, while $\tilde{O}(\cdot)$ absorbs logarithmic factors. $\mathbb{S}^{d-1}$ denotes the unit hypersphere in $\mathbb{R}^d$ and $\mathbb{B}_R^d$ denotes a d-dimensional ball of radius R. The activation function is the ReLU denoted by $\phi(z) = \max\{0, z\}$. We write $[n] = \{1, 2, \ldots, n\}$.

Architectures and sparse connectivity. Fix integers $d \geq 1$ and $1 \leq m \leq d$. A *sparse connectivity pattern* is specified by a collection of coordinate subsets (receptive fields) $\mathcal{S} = \{S_j\}_{j=1}^J$, where each $S_j \subset [d]$ has size m. Each subset induces a coordinate projection $\pi_j : \mathbb{R}^d \to \mathbb{R}^m$, and we call $\boldsymbol{x}^{(S_j)} := \pi_j(\boldsymbol{x})$ a *patch*. This formalizes the property that each hidden neuron only receives input from a restricted subset of the ambient coordinates.

We analyze two-layer sparsely connected ReLU networks with weight sharing and Global Average Pooling (GAP)[1]. Given receptive fields $\mathcal{S}$ and width $K \in \mathbb{N}$, we consider

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \left(\frac{1}{J} \sum_{j=1}^{J} \phi\big(\boldsymbol{w}_k^\mathsf{T} \pi_j(\boldsymbol{x}) - b_k\big) \right) + \beta. \tag{1}$$

We refer to this architecture as a *sparsely connected network* (SCN) or $\mathcal{S}$CN if the sparse connectivity pattern is prescribed. A unit $\phi(\boldsymbol{w}_k^\mathsf{T} \cdot - b_k)$ is called a *filter* or *neuron*. It is said to be *activated* on a patch $\boldsymbol{x}^{(S_j)}$ if $\boldsymbol{w}_k^\mathsf{T} \boldsymbol{x}^{(S_j)} > b_k$. Let $\boldsymbol{\Theta}^{\mathcal{S}}$ be the parameter set of all such $\mathcal{S}$CNs of *any finite* width.

Data and loss function. A dataset is $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \in \mathbb{B}_R^d$ and $y_i \in [-D, D]$. We use the squared loss and define the training objective $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2n} \sum_{i=1}^n \big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\big)^2$.

"Edge of Stability" regime. Empirical and theoretical work [Cohen et al., 2021, Damian et al., 2023] has shown that gradient descent on neural networks typically enters the Edge of Stability (EoS) regime: the largest Hessian eigenvalue $\lambda_{\max}(\nabla^2 \mathcal{L}(\boldsymbol{\theta}_t))$ steadily increases ("progressive sharpening") until it hovers around the value $2/\eta$, where $\eta > 0$ is the step size. Throughout this paper, GD refers to vanilla gradient descent with learning rate η, and we assume $\eta < 2$.

Definition 3.1 (Below Edge of Stability [Qiao et al., 2024, Definition 2.3])**.** For a dataset $\mathcal{D}$ and step size η, a parameter $\boldsymbol{\theta}$ at which $\mathcal{L}$ is twice differentiable is *Below the Edge of Stability (BEoS)* if $\lambda_{\max}(\nabla^2 \mathcal{L}(\boldsymbol{\theta})) \leq 2/\eta$.

A GD iterate satisfying this condition belongs to the BEoS parameter class. Our results characterize this class, defined by

$$\boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta, \mathcal{D}) := \left\{ \boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}} \,\middle|\, \mathcal{L} \text{ is twice differentiable at } \boldsymbol{\theta},\ \lambda_{\max}\big(\nabla^2 \mathcal{L}(\boldsymbol{\theta})\big) \leq \frac{2}{\eta} \right\}. \tag{2}$$

Statistical learning framework. We assume the data points $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are i.i.d. samples from a distribution $\mathcal{P}$ on $\mathbb{B}_R^d \times [-D, D]$. The *population risk* of a predictor f is $\mathcal{R}(f) := \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}}[(f(\boldsymbol{x}) - y)^2]$. The *empirical risk* on dataset $\mathcal{D}$ is $\widehat{\mathcal{R}}_{\mathcal{D}}(f) = \frac{1}{n} \sum_{i=1}^n (f(\boldsymbol{x}_i) - y_i)^2$. The *generalization gap* is defined as $\mathrm{GenGap}(f, \mathcal{D}) := |\mathcal{R}(f) - \widehat{\mathcal{R}}_{\mathcal{D}}(f)|$.

[1]Weight sharing and GAP are non-essential and chosen only for cleaner exposition. Appendix F proves an analogous result for SCNs without weight sharing, where the global patch-multiset weight $g_{\mathcal{D},\mathcal{S}}$ is replaced by location wise weights.

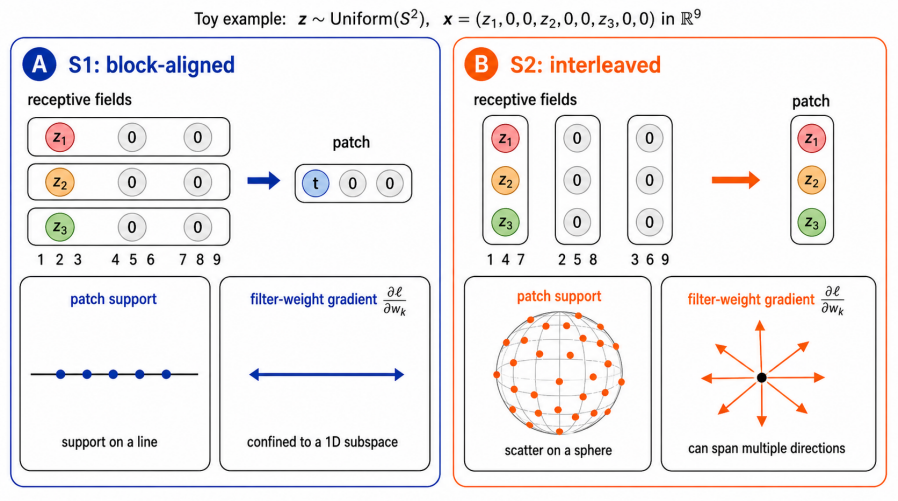


Figure 1: **Data $\times$ architecture determines the geometry that gradient descent interacts with.**

4 Main Results

To start with, we use a toy example to preview how sparse connectivity in the network shapes the gradient dynamics itself. Consider an input $z \sim \text{Uniform}(\mathbb{S}^2)$ embedded into $\mathbb{R}^9$ by padding each coordinate with two zeros. The ambient data is spherical, so a fully connected network sees a high-dimensional unstructured geometry. Now equip a two-layer SCN with two different receptive field systems: **(A)** $\mathcal{S}_1$ (block-aligned): each receptive field groups 3 consecutive coordinates. Every patch has the form $(t, 0, 0)$, so the patch geometry is one-dimensional; **(B)** $\mathcal{S}_2$ (interleaved): each receptive field groups coordinates modulo the stride. A patch recovers the full vector (z_1, z_2, z_3), so the patch geometry is spherical. The effect of these two designs becomes visible in the gradient. For the model (1), the gradient of the training loss with respect to a filter $\boldsymbol{w}_k$ is

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{w}_k} = \frac{v_k}{nJ} \sum_{i=1}^{n} \sum_{j=1}^{J} \underbrace{(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i) \mathbb{1}\{\boldsymbol{w}_k^\top \pi_j(\boldsymbol{x}_i) > b_k\}}_{\text{residue and activation scalars}} \underbrace{\pi_j(\boldsymbol{x}_i)}_{\text{patch vectors}} . \tag{3}$$

The update to $\boldsymbol{w}_k$ is a linear combination of the patches $\pi_j(\boldsymbol{x}_i)$ that activate it. Under $\mathcal{S}_1$, every activated patch lies in a one-dimensional subspace, so each filter gradient is confined to that subspace. Under $\mathcal{S}_2$, activated patches span the entire 3-dimensional patch space, and filter gradients can explore all directions. The two receptive field systems thus produce fundamentally different gradient dynamics, and the sets of solutions that GD can stably reach differ dramatically.

The rest of this section develops a framework that makes this intuition precise via the lens of dynamical stability analysis in gradient descent (the BEoS regime). Theorem 4.1 shows that the BEoS condition bounds a weighted path norm whose weight is governed by the patch geometry. Theorems 4.2 and 4.3 then establish a vanishing generalization upper bound for projected spherical patches and a constant population-error lower bound for spherical patches.

4.1 From the BEoS condition to a patch-geometry-weighted path norm

Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and receptive fields $\mathcal{S} = \{S_j\}_{j=1}^J$, let $\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}}$ be a random vector drawn uniformly from the patch multiset $\{\pi_j(\boldsymbol{x}_i)\}_{(i,j)\in[n]\times[J]} \subset \mathbb{R}^m$. Define the weight function $g_{\mathcal{D},\mathcal{S}} \colon \mathbb{S}^{m-1} \times \mathbb{R} \to \mathbb{R}$ by

$$g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) := \mathbb{E}\left[\phi\left(\boldsymbol{u}^\top \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} - t\right)\right] \sqrt{\mathbb{P}\left(\boldsymbol{u}^\top \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t\right)^2 + \left\| \mathbb{E}\left[\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} \mathbb{1}\left\{\boldsymbol{u}^\top \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t\right\}\right] \right\|_2^2}. \tag{4}$$

Theorem 4.1. *Fix $\mathcal{D}$ and $\mathcal{S}$. For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}}$,*

$$\sum_{k=1}^{K} |v_k| \, \|\boldsymbol{w}_k\| \, g_{\mathcal{D},\mathcal{S}}\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\right) \le \frac{1}{2}\left(\lambda_{\max}\left(\nabla^2 \mathcal{L}(\boldsymbol{\theta})\right) + 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})} - 1\right). \tag{5}$$

In particular, for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$,

$$\sum_{k=1}^{K} |v_k| \, \|\boldsymbol{w}_k\| \, g_{\mathcal{D},\mathcal{S}}\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\right) \leq \frac{1}{\eta} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{6}$$

The proof is in Appendix C. In particular, the connection between the Hessian and $g_{\mathcal{D},\mathcal{S}}$ is detailed in Proposition C.10. The weight $g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t)$ combines the mean activation with the size of the mean tangent features associated with the bias and filter weights. In (4), these tangent features are the activation frequency and the mean activated patch vector. Together, they measure how strongly a filter's path norm is penalized by the curvature constraint. Rare, small activations give a small weight, allowing a larger path norm under the same BEoS threshold. Changing the receptive fields changes these patch statistics and hence the penalty. Returning to our example, under $\mathcal{S}_1$ the patch multiset is essentially one-dimensional and highly concentrated; a direct calculation shows that $g_{\mathcal{D},\mathcal{S}}$ is bounded away from zero for any neuron that activates on a non-negligible fraction of patches, so the BEoS condition enforces a meaningful path-norm constraint. Under $\mathcal{S}_2$, the patch multiset is a 3-dimensional sphere; a neuron can isolate an individual patch, driving $g_{\mathcal{D},\mathcal{S}}$ arbitrarily close to zero and rendering the BEoS constraint vacuous. The next two subsections develop the quantitative consequences of this dichotomy.

4.2 When patch geometry induces strong regularization

The next result formalizes the $\mathcal{S}_1$ regime: small receptive fields on spherical ambient data.

Theorem 4.2. *Let* $\mathcal{P}$ *have marginal* $\mathrm{Uniform}(\mathbb{S}^{d-1})$*, and let* $\mathcal{D}$ *be a dataset of* n *i.i.d samples. Assume* $d > 3$ *and* $1 \leq m < \frac{d(d-3)}{d+3}$*. For any* $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$ *with* $\|f_{\boldsymbol{\theta}}\|_\infty \leq M$*, with probability* $\geq 1 - 2\delta$*,*

$$\mathrm{GenGap}(f_{\boldsymbol{\theta}}, \mathcal{D}) \lessapprox_d \mathrm{poly}\left(\frac{1}{\eta}, J, M\right) n^{-\frac{(d-m)(d+3)}{6d^2-2md+6d-6m}}, \tag{7}$$

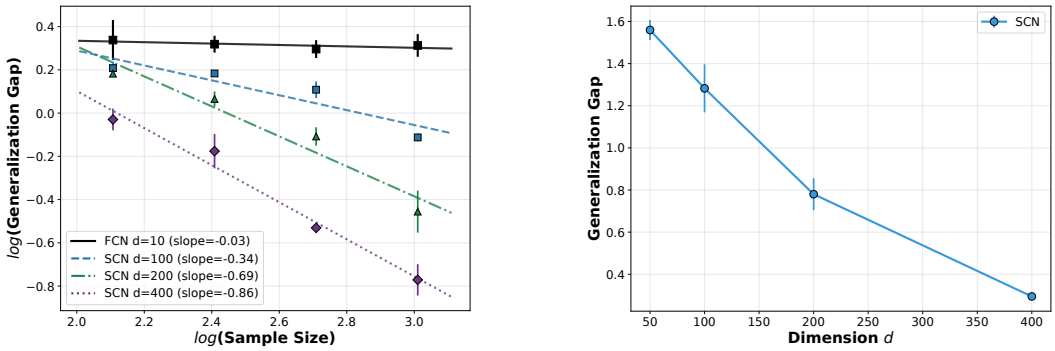
where $\lessapprox_d$ *hides constants depending on* d *and logarithmic factors in* n *and* (J/δ)*.*

For fixed m and large d, the rate approaches $n^{-1/6}$, while FCNs on the same spherical inputs admit a constant lower bound [Liang et al., 2026]. Small projections concentrate near the origin of $\mathbb{R}^m$, where the patch-weighted constraint effectively controls complexity. Figure 2 illustrates the resulting decay of the generalization gap.

The proof separates an interior, where every patch has norm at most $1-\varepsilon$, from the boundary region. Filters whose activation thresholds cross the interior have population weights of order at least ε^d. Uniform concentration transfers this control to the empirical weights, so Theorem 4.1 bounds the ordinary variation norm on the interior by a multiple of ε^{-d}. The resulting complexity term scales as $\varepsilon^{-d^2/(d+3)} n^{-1/2}$. The boundary contributes through its probability mass, of order $\varepsilon^{(d-m)/2}$, since the predictor is bounded. Here we suppress factors depending on η, J, M, d and logarithmic terms. Balancing these two contributions gives the rate in Theorem 4.2. Thus, the exponent near $1/6$ reflects the tradeoff between interior complexity and the mass near weakly penalized activation boundaries. Appendix D gives the full proof.

4.3 When patch geometry fails to constrain

Theorem 4.2 shows that small projections of an ambient spherical input yield a vanishing generalization gap. When the patches themselves lie on spheres, the BEoS class contains bounded predictors with constant population error. The following theorem gives a lower bound for noiseless regression.



(a) Log–log scaling with n.

(b) GenGap versus d (fixed n).

Figure 2: **Generalization-gap scaling in synthetic experiments. (Left)** $\mathrm{GenGap}(f_{\boldsymbol{\theta}}, \mathcal{D})$ versus the sample size n on a log–log scale. The fitted slope summarizes the empirical rate: if $\mathrm{GenGap} \lesssim n^{-c}$, then $\log(\mathrm{GenGap}) \leq -c \log n + b$, so a more negative slope indicates faster decay (better generalization). In our experiments, the FCN slope is nearly flat (slope $= -0.03$ at $d = 10$), whereas SCN exhibits increasingly negative slopes as d grows (slope $= -0.34$ at $d = 100$, -0.69 at $d = 200$, and -0.86 at $d = 400$), indicating faster decay with n. **(Right)** $\mathrm{GenGap}(f_{\boldsymbol{\theta}}, \mathcal{D})$ versus the ambient dimension d with n fixed, illustrating that for SCN (with patch size $m \ll d$) the generalization gap remains stable and can even decrease as d increases. More details can be found in Appendix G.

Theorem 4.3 (A generalization-gap lower bound under spherical patch geometry)**.** *Let $m \geq 2$, $J \geq 1$, and $d = mJ$. Take the receptive fields $\mathcal{S} = \{S_j\}_{j=1}^{J}$ with $S_j = \{(j-1)m + 1, \ldots, jm\}$. Let $\mathcal{P}$ be the distribution of $(\boldsymbol{X}, Y)$ with $\boldsymbol{X} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_J)$, independent $\boldsymbol{Z}_j \sim \mathrm{Uniform}(\mathbb{S}^{m-1})$, and $Y = 0$. Its input support is $(\mathbb{S}^{m-1})^J \subset \mathbb{B}^d_{\sqrt{J}}$. There exists $c_m > 0$, depending only on m, such that for every $M > 0$, $0 < \eta < 2$, and every dataset $\mathcal{D}$ of $n \geq 1$ samples from $\mathcal{P}$, there is a parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$ of width at most nJ satisfying*

$$\sup_{\boldsymbol{x} \in (\mathbb{S}^{m-1})^J} |f_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq M, \qquad \widehat{\mathcal{R}}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) = 0, \qquad \lambda_{\max}(\nabla^2 \mathcal{L}(\boldsymbol{\theta})) = 1, \tag{8}$$

and

$$\mathcal{R}(f_{\boldsymbol{\theta}}) = \mathrm{GenGap}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathbb{E}[f_{\boldsymbol{\theta}}(\boldsymbol{X})^2] \geq c_m M^2. \tag{9}$$

Since $Y = 0$, this population risk is also the excess risk.

The construction places filters in empty spherical caps, where $g_{\mathcal{D},\mathcal{S}} = 0$ because every training activation vanishes. Their supports jointly retain constant population mass. This is the counterpart of the boundary term in the upper-bound proof: projected spherical patches place vanishing mass near the boundary, while spherical patches support a constant population effect in directions with zero empirical weight. The proof appears in Appendix E.

Figure 3 examines training on independent spherical patches with noisy labels. The SCN interpolates while its sharpness hovers near $2/\eta$, and many filters develop large path norms with low activation rates. These rare activations receive weak weighted path-norm penalties.

Together, Theorem 4.2 and Theorem 4.3 identify two population regimes. Projecting an ambient spherical input into small patches yields a generalization bound that vanishes with n. When the patches themselves lie on spheres, empty patch regions support bounded BEoS predictors with constant population error. The contrast is governed by the population mass carried by directions that receive weak stability-induced penalties.

5 Patch Geometry of Natural Images

We now examine which side of the dichotomy established in Section 4 real image data falls on. We extract 3×3 convolutional patches (stride 1, no padding) from the CIFAR-10 training set, sample 10^7 patches, and apply standard torchvision normalization. Figure 4 compares the patch cloud with the cloud of full images. The patch cloud is strongly concentrated: 90% of its variance lives in

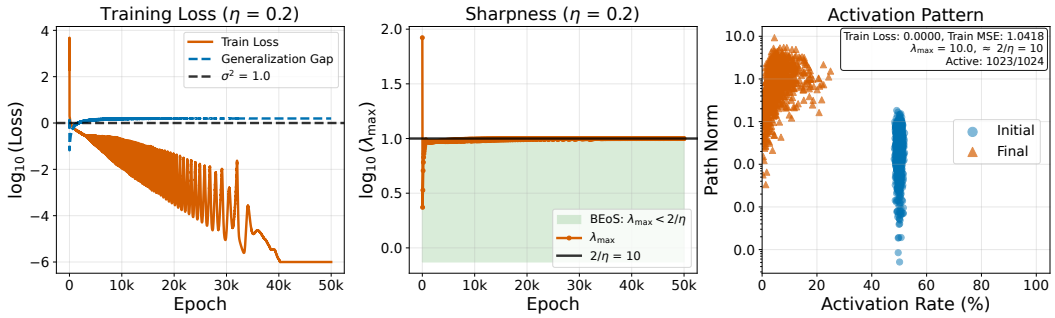


Figure 3: **Interpolation below the edge of stability. Left.** Training loss and generalization gap, showing that the SCN interpolates noisy labels in the random spherical-patch setting. **Middle.** The largest Hessian eigenvalue hovers near $2/\eta \approx 10$, indicating compatibility with the EoS regime. **Right.** Scatter plot of per-neuron path norm $|v_k| \|\boldsymbol{w}_k\|$ against activation rate $\frac{1}{nJ}\sum_{i,j} \mathbb{1}\{\boldsymbol{w}_k^\top \pi_j(\boldsymbol{x}_i) > b_k\}$. Many neurons have large path norms while activating on only a small fraction of the patch multiset, illustrating how rare activations receive weak stability-induced penalties.

just 3 directions of $\mathbb{R}^{27}$, while the full images in $\mathbb{R}^{3072}$ need over 100 directions. Recalling the gradient view (3), where filter updates are linear combinations of activated patch vectors, such a low-dimensional patch cloud implies that the gradients themselves are constrained to a low-rank subspace—much like the one-dimensional $\mathcal{S}_1$ example in Section 4, where filter updates were confined to a single direction. Following Liang et al. [2026], we further assess shatterability via half-space depth. The right panel reports the concentration curve $\Psi(T)$, the fraction of patches whose half-space depth exceeds T. A larger area under Ψ indicates that typical patches lie deeper in the distribution, making them harder to isolate with a single hyperplane. The patch cloud exhibits substantially larger area than the image cloud.

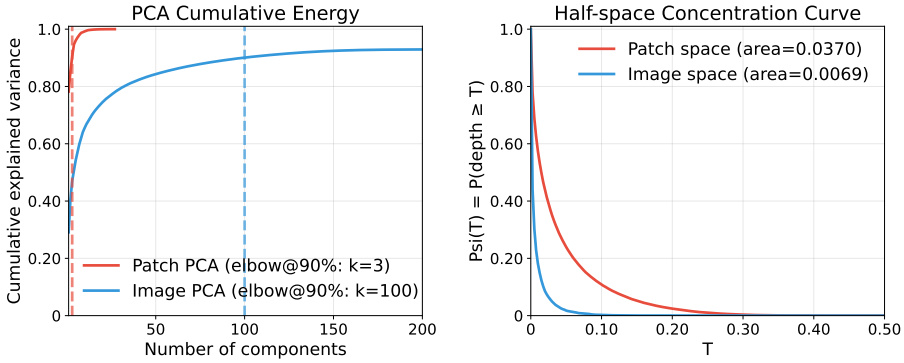


Figure 4: **Patch geometry vs. image geometry on CIFAR-10. Left:** PCA explained-variance ratio for the patch cloud ($\mathbb{R}^{27}$, solid) and the ambient image cloud ($\mathbb{R}^{3072}$, dashed). Patches concentrate 90% variance in 3 principal directions; images need over 100. **Right:** Half-space concentration curves $\Psi(T) = \mathbb{P}(\text{depth}(\mathbf{X}) \geq T)$ for $T \in [0, 0.5]$. Larger area means deeper points and fewer opportunities for near-isolating hyperplanes. The patch cloud shows markedly larger concentration.

These geometric properties place convolutional patches on CIFAR-10 on the structured side of the dichotomy. We verify the predicted generalization behavior with a nonparametric regression experiment on CIFAR-10 with noisy labels, comparing a two-layer SCN (3×3 convolution, stride 1, no padding) and a fully connected network of the same width. Figure 5 reports the results. The SCN resists overfitting while the FCN interpolates noise (left panel). Both networks enter the BEoS regime after roughly 3000 epochs, with the largest Hessian eigenvalue hovering around $2/\eta$ (middle panel). The right panel tracks excess risk throughout training: after roughly 2k epochs, the SCN maintains a substantially lower excess risk than the FCN. The contrast follows from the geometric measurements: the structured patch cloud makes the BEoS constraint effective for the SCN, whereas the unstructured image cloud provides no such constraint for the FCN.

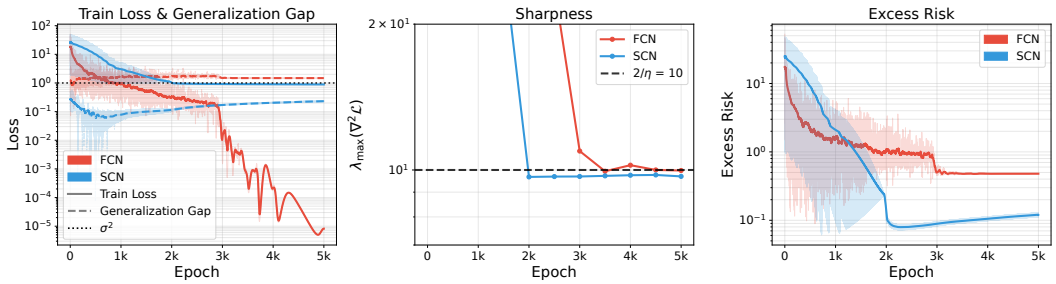


Figure 5: **Generalization on CIFAR-10 with noisy labels.** Two-layer SCN (3×3 convolution, stride 1, no padding) vs. FCN of the same width, both trained with full-batch GD ($\eta = 0.2$). **Left:** Training loss (solid) and generalization gap (dashed) over epochs. The SCN stabilizes near the noise floor; the FCN interpolates. **Middle:** Sharpness $\lambda_{\max}(\nabla^2 \mathcal{L})$. Both networks hover around $2/\eta \approx 10$ after $\sim$3k epochs (before that the loss of FCN has not started to converge), confirming the BEoS regime. **Right:** After 2k epochs, SCN maintains substantially lower excess risk than FCN.

Table 2: The design trend of sparse connectivity in modern vision backbones.

Representative backbone	**Core sparse connectivity pattern**	**Dimension of the processed unit**
ResNet-50 [He et al., 2016]	3×3 bottleneck convolution	$3 \times 3 \times C_{\text{in}}$ (up to $C_{\text{in}} = 512$)
ResNeXt-50 [Xie et al., 2017]	3×3 grouped convolution	$3 \times 3 \times C_{\text{in}}/g$ (up to $g = 32$)
ConvNeXt-T [Liu et al., 2022]	7×7 depthwise convolution	$7 \times 7 = 49$
ViT-B/16 [Dosovitskiy et al., 2021]	uniform Transformer	768 for all blocks
Swin-T [Liu et al., 2021b]	hierarchical Transformer	$96 \to 192 \to 384 \to 768$

6 Discussion

We set out to understand how sparse connectivity shapes the way gradient descent trained two-layer ReLU networks generalize, viewed through the lens of stability analysis. We proved that the stability of gradient descent translates into an explicit constraint on the network whose strength is governed by the geometry of the patches that sparse connectivity produces. We then proved that this constraint yields non-vacuous generalization bounds on high-dimensional spherical data, where the induced patch geometry is concentrated and low-dimensional, and where fully connected networks provably fail. On independent spherical patches, we proved a complementary constant excess-risk lower bound by constructing bounded BEoS predictors supported on empty patch regions. Our CIFAR-10 analysis shows that convolutional receptive fields produce concentrated patches that resist isolation, connecting natural-image geometry to this generalization mechanism.

This geometric perspective suggests a broader lens for thinking about architecture design. Sparse connectivity, the structure at the center of our analysis, is equally fundamental in deep networks: it appears in nearly every building block of modern CNNs and Vision Transformers. In our two-layer setting, sparse connectivity processes patches of the raw input, and we showed that the effective dimension of these patches controls the strength of implicit regularization. In a deep network, a local operator at an intermediate layer processes not raw pixels but a patch of the representation produced by earlier layers. The geometry on which the stability constraint acts is therefore a dynamically evolving learned representation rather than a fixed data distribution.

This shift recasts sparse connectivity as more than an efficiency tool. Through its receptive-field configuration, it also shapes the implicit regularization that gradient descent provides, with the vector dimension seen by each local operator as the relevant quantity. Table 2 collects this perspective across architectures. From ResNet [He et al., 2016] to ResNeXt [Xie et al., 2017] to ConvNeXt [Liu et al., 2022], the per-operator dimension steadily shrinks through bottlenecks, grouping, and depthwise convolutions. ConvNeXt's kernel-size ablation finds that increasing the depthwise kernel from 3×3 to 7×7 improves performance, with gains saturating at larger sizes. RepLKNet [Ding et al., 2022] extends this direction further by showing that very large spatial kernels can be effective when

the input to the spatial operator is restricted to k^2 rather than k^2C through depthwise design. Separating spatial and channel mixing may also produce more structured patches, a direction for future study.

Vision Transformers exhibit the same pattern. From ViT [Dosovitskiy et al., 2021] to Swin [Liu et al., 2021b], a uniform high-dimensional token space gives way to a hierarchy whose dimensions grow gradually, controlling the effective local operator dimension while recovering expressive power through later-stage mixing. Recent systematic patch scaling studies show that ViT performance on classification and dense prediction consistently improves as patch size decreases, even to 1×1 pixel tokenization [Wang et al., 2025b]. In addition, self-supervised ViTs such as DINO also highlight the importance of small patches [Caron et al., 2021]. In our framework, reducing patch size changes the geometry of the patch matrix itself, potentially exposing the optimizer to more structured local vectors and thus strengthening the implicit regularization. Appendix A develops these connections.

Our analysis identifies sparse connectivity as an active part of the gradient dynamics governing representation learning and implicit regularization. This geometric perspective motivates joint design of architectures and training algorithms through the representations they expose to gradient descent.

Acknowledgments and Disclosure of Funding

YW acknowledges support from NSF DMS 2134214. AC acknowledges support from NSF CISE 2403452, NSF DMS 2608292, and a fellowship from Simons Foundation. RP acknowledges support from a Hellman Fellowship from the University of California.

We thank anonymous reviewers and AC at NeurIPS'26 for helpful feedback.

References

NIST digital library of mathematical functions. http://dlmf.nist.gov/. See §5.6(i), Eq. 5.6.E4 for Gautschi's inequality (Gamma ratio bounds).

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11): 4311–4322, 2006.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 233–242. PMLR, 2017.

Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14496–14506, 2023.

Alon Brutzkus, Amir Globerson, Eran Malach, Alon Regev Netser, and Shai Shalev-Schwartz. Efficient learning of cnns using patch based features. In *International Conference on Machine Learning*, pages 2336–2356. PMLR, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *International Conference on Learning Representations*, 2023.

Lijun Ding, Dmitriy Drusvyatskiy, Maryam Fazel, and Zaid Harchaoui. Flat minima generalize for low-rank matrix recovery. *Information and Inference: A Journal of the IMA*, 13(2):iaae009, 2024.

Xiaohan Ding, Xiangyu Zhang, Yizhuang Zhou, Jungong Han, Guiguang Ding, and Jian Sun. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11963–11975, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

Tomer Galanti, Mengjia Xu, Liane Galanti, and Tomaso Poggio. Norm-based generalization bounds for sparse neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/hash/8493e190ff1bbe3837eca821190b61ff-Abstract-Conference.html`.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018a.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018b.

David Haussler. Decision-theoretic generalizations of the pac model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

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

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Aakash Lahoti, Stefani Karp, Ezra Winston, Aarti Singh, and Yuanzhi Li. Role of locality and weight sharing in image-based tasks: A sample complexity separation between cnns, lcns, and fcns. In *The Twelfth International Conference on Learning Representations*, 2024.

Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

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

Zhiyuan Li, Yi Zhang, and Sanjeev Arora. Why are convolutional nets more sample-efficient than fully-connected nets? In *International Conference on Learning Representations*, 2021.

Tongtong Liang, Dan Qiao, Yu-Xiang Wang, and Rahul Parhi. Stable minima of ReLU neural networks suffer from the curse of dimensionality: The neural shattering phenomenon. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Tongtong Liang, Alexander Cloninger, Rahul Parhi, and Yu-Xiang Wang. Generalization below the edge of stability: The role of data geometry. In *International Conference on Learning Representations (ICLR)*, 2026.

Hao Liu, Minshuo Chen, Tuo Zhao, and Wenjing Liao. Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. In *International Conference on Machine Learning*, pages 6770–6780. PMLR, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021b.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022.

Tong Mao, Zhongjie Shi, and Ding-Xuan Zhou. Theory of deep convolutional neural networks iii: Approximating radial functions. *Neural Networks*, 144:778–790, 2021.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. When and why are deep networks better than shallow ones? In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

Rotem Mulayoff, Tomer Michaeli, and Daniel Soudry. The implicit bias of minima stability: A view from function space. *Advances in Neural Information Processing Systems*, 34:17749–17761, 2021.

Ramchandran Muthukumar and Jeremias Sulam. Sparsity-aware generalization theory for deep neural networks. In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 5311–5342. PMLR, 2023. URL `https://proceedings.mlr.press/v195/muthukumar23a.html`.

Mor Shpigel Nacson, Rotem Mulayoff, Greg Ongie, Tomer Michaeli, and Daniel Soudry. The implicit bias of minima stability in multivariate shallow ReLU networks. In *International Conference on Learning Representations*, 2023.

Kamil Nar and Shankar Sastry. Step size matters in deep learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Duy-Kien Nguyen, Mahmoud Assran, Unnat Jain, Martin R. Oswald, Cees G. M. Snoek, and Xinlei Chen. An image is worth more than 16x16 patches: Exploring transformers on individual pixels. *arXiv preprint arXiv:2406.09415*, 2024.

Kenta Oono and Taiji Suzuki. Approximation and non-parametric estimation of resnet-type convolutional neural networks. In *International conference on machine learning*, pages 4922–4931. PMLR, 2019.

Rahul Parhi and Robert D. Nowak. Near-minimax optimal estimation with shallow ReLU neural networks. *IEEE Transactions on Information Theory*, 69(2):1125–1139, 2023.

Mattis Paulin, Julien Mairal, Matthijs Douze, Zaid Harchaoui, Florent Perronnin, and Cordelia Schmid. Convolutional patch representations for image retrieval: an unsupervised approach. *International Journal of Computer Vision*, 121(1):149–168, 2017.

Gabriel Peyré. Manifold models for signals and images. *Computer vision and image understanding*, 113(2):249–260, 2009.

Tomaso Poggio. Foundations of deep learning: Compositional sparsity of computable functions. Technical report, CBMM memo 138, 2022.

Tomaso Poggio and Maia Fraser. Compositional sparsity of learnable functions. *Bulletin of the American Mathematical Society*, 61(3):438–456, 2024.

Dan Qiao, Kaiqi Zhang, Esha Singh, Daniel Soudry, and Yu-Xiang Wang. Stable minima cannot overfit in univariate ReLU networks: Generalization by large step sizes. In *Advances in Neural Information Processing Systems*, volume 37, pages 94163–94208, 2024.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Zhongjie Shi, Zhiying Fang, and Yuan Cao. Approximation and estimation capability of vision transformers for hierarchical compositional models. *Applied and Computational Harmonic Analysis*, page 101849, 2025.

Jonathan W. Siegel and Jinchao Xu. Characterization of the variation spaces corresponding to shallow neural networks. *Constructive Approximation*, pages 1–24, 2023.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70): 1–57, 2018.

Louis Thiry, Michael Arbel, Eugene Belilovsky, and Edouard Oyallon. The unreasonable effectiveness of patches in deep convolutional kernels methods. In *International Conference on Learning Representations*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 2021.

Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Feng Wang, Timing Yang, Yaodong Yu, Sucheng Ren, Guoyizhe Wei, Angtian Wang, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Adventurer: Optimizing vision mamba architecture designs for efficiency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30157–30166, June 2025a.

Feng Wang, Yaodong Yu, Guoyizhe Wei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Scaling laws in patchification: An image is worth 50,176 tokens and more. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 65278–65290. PMLR, 2025b.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

Zihao Wang and Lei Wu. Theoretical analysis of the inductive biases in deep convolutional networks. *Advances in Neural Information Processing Systems*, 36:74289–74338, 2023.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22–31, 2021.

Lei Wu and Weijie J Su. The implicit regularization of dynamical stability in stochastic gradient descent. In *International Conference on Machine Learning*, pages 37656–37684. PMLR, 2023.

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

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *Advances in Neural Information Processing Systems*, volume 34, pages 30392–30400, 2021.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, 2020.

Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural networks*, 94: 103–114, 2017.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis E. H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 558–567, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. URL `https://openreview.net/forum?id=Sy8gdB9xx`.

Zixuan Zhang, Kaiqi Zhang, Minshuo Chen, Yuma Takeda, Mengdi Wang, Tuo Zhao, and Yu-Xiang Wang. Nonparametric classification on low dimensional manifolds using overparameterized convolutional residual networks. *Advances in Neural Information Processing Systems*, 37:65738–65764, 2024.

Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48(2):787–794, 2020.

A Broader (Informal) Discussion on Network Architecture Design: Bridging our Theoretical Insights to Empirical Observations

This appendix expands on the architectural implications discussed in Section 6. The purpose is not to claim that the stability analysis in the main text provides a complete theory of modern deep vision architectures. Rather, the point is that stability gives a tractable lens through which one can see a more general object: the geometry of the local vectors exposed to gradient descent. This object appears explicitly in the shallow model analyzed in this paper, and it has close analogues in the patchification and local-operator design choices used in modern CNNs and Vision Transformers.

The patch matrix as the geometry seen by backpropagation. In the setting of this paper, consider the patch matrix

$$\mathbf{X} \in \mathbb{R}^{(nJ)\times m},$$

whose row indexed by (i, j) equals $\pi_j(\boldsymbol{x}_i)^\mathsf{T}$. Let $r_i := f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i$, and define the lifted residual vector $\bar{\mathbf{r}} \in \mathbb{R}^{nJ}$ by

$$(\bar{\mathbf{r}})_{(i,j)} := r_i.$$

Let $\mathbf{S}(\boldsymbol{\theta}) \in \{0, 1\}^{(nJ)\times K}$ be the gating matrix

$$\mathbf{S}(\boldsymbol{\theta})_{(i,j),k} \;:=\; \mathbb{1}\big\{\boldsymbol{w}_k^\mathsf{T}\pi_j(\boldsymbol{x}_i) > b_k\big\},$$

and let $\boldsymbol{v} := (v_1, \ldots, v_K)^\mathsf{T}$. Stack first-layer weights as

$$\mathbf{W} := [\boldsymbol{w}_1^\mathsf{T}; \ldots; \boldsymbol{w}_K^\mathsf{T}] \in \mathbb{R}^{K\times m}.$$

A direct calculation gives a backprop-aligned factorization of the gradient. Define the patch-level backprop signal

$$\mathbf{R}(\boldsymbol{\theta}) := \frac{1}{nJ}\,\bar{\mathbf{r}}\,\boldsymbol{v}^\mathsf{T} \in \mathbb{R}^{(nJ)\times K}, \quad \mathbf{G}(\boldsymbol{\theta}) := \mathbf{R}(\boldsymbol{\theta}) \odot \mathbf{S}(\boldsymbol{\theta}),$$

where $\odot$ denotes entrywise multiplication. Then

$$\nabla_{\mathbf{W}}\mathcal{L}(\boldsymbol{\theta}) \;=\; \mathbf{G}(\boldsymbol{\theta})^\mathsf{T}\mathbf{X}. \tag{10}$$

Equivalently, writing $\mathbf{V} := \mathrm{diag}(v_1, \ldots, v_K)$, one can express the same signal as

$$\mathbf{G}(\boldsymbol{\theta}) = \frac{1}{nJ}\,\mathrm{diag}(\bar{\mathbf{r}})\,\mathbf{S}(\boldsymbol{\theta})\,\mathbf{V}.$$

The factorization (10) separates two roles. The matrix $\mathbf{G}(\boldsymbol{\theta})$ aggregates the learning signal generated by residuals, readout weights, and gates. The patch matrix $\mathbf{X}$ then maps this signal into parameter updates. Thus the architecture does not only restrict the number of parameters, it determines the geometry through which backpropagated signals are converted into motion in parameter space. In this sense, $\mathbf{X}$ acts as a signal rectifier in the sense that data priors and receptive-field structure jointly shape the directions along which gradient descent can effectively move. For example, Figure 4 shows that, for CIFAR-10, 90% of the energy of the convolutional patch matrix is concentrated in only a few principal directions, suggesting a strong geometric constraint on the filter dynamics.

This viewpoint also clarifies why the stability condition in the main text is informative. The terms appearing in the definition of $g_{\mathcal{D},\mathcal{S}}$ are drawn from the same ingredients as (10): gate statistics induced by $\mathbf{S}(\boldsymbol{\theta})$ and geometric moments of the patch cloud encoded by $\mathbf{X}$. Theorem 4.2 can therefore be viewed as a static stability proxy for the backpropagation geometry in (10). Theorem 4.3 shows how empty regions of patch space support BEoS predictors with zero training error and constant population error. These regions carry positive population mass while their filters receive zero empirical weight in the stability constraint. If the patch cloud is concentrated and resistant to such isolation, the same stability lens yields an effective regularity constraint.

Patch size as local-operator dimension in Vision Transformers. Vision Transformers make the role of patch geometry especially explicit. The original ViT converts an image into a sequence of non-overlapping patches and applies a shared patch embedding before global token mixing [Dosovitskiy et al., 2021]. This patchification step is often introduced as an efficiency device, since larger patches shorten the token sequence and reduce the cost of attention. The factorization (10) suggests a second interpretation. Patchification also chooses the first local geometry seen by training. A

Table 3: Selected patchification-scaling results adapted from [Wang et al., 2025b]. Patch size denotes the side length of the square image patch used to form one token. Panel A shows that reducing patch size improves ImageNet accuracy. Panel B separates true patch-size reduction from only increasing the sequence length by interpolating coarse tokens. Here we only list their results on DeiT-B [Touvron et al., 2021] and Adventurer-B [Wang et al., 2025a].

Panel A: ImageNet classification top-1 accuracy

Model and input	16×16	8×8	4×4	2×2	1×1
DeiT-B, 64×64 input	68.2	76.9	80.1	80.8	81.3
DeiT-B, 128×128 input	78.1	81.0	82.3	82.9	–
Adventurer-B, 224×224 input	82.6	83.9	84.3	84.5	84.6

Panel B: Sequence-length ablation on ImageNet top-1 accuracy

Model and input	**Seq. length**	**Interpolated coarse tokens**	**True patch scaling**
DeiT-B, 128×128 input	256	78.2 (+0.1)	81.0 (+2.9)
DeiT-B, 128×128 input	1,024	78.2 (+0.1)	82.3 (+4.2)
Adventurer-B, 224×224 input	784	82.7 (+0.1)	83.9 (+1.3)
Adventurer-B, 224×224 input	3,136	82.8 (+0.2)	84.3 (+1.7)
Adventurer-B, 224×224 input	12,544	82.8 (+0.2)	84.5 (+1.9)

larger patch presents each shared local operator with a higher-dimensional and more compressed local vector. A smaller patch reduces this local dimension and preserves finer spatial information, while increasing the number of tokens that later layers must mix.

Several empirical lines of work support the idea that patch size is a genuine training and generalization variable, not only an implementation detail. FlexiViT trains a single ViT over a range of patch sizes and demonstrates that patch size controls an accuracy–compute tradeoff at deployment time [Beyer et al., 2023]. Self-supervised ViTs also reveal the importance of this axis: DINO identifies small patches as one ingredient behind the strong emergent properties of self-supervised ViTs [Caron et al., 2021]. Early-convolution studies make the same point from the optimization side. The standard ViT patchify stem is a large-kernel, large-stride convolution, and replacing it by a short stack of small-stride convolutions significantly changes optimization stability and improves performance under common training recipes [Xiao et al., 2021]. These results are consistent with the idea that the first patch matrix is not an innocuous preprocessing choice: it shapes the geometry of the signals that subsequent layers and backpropagation operate on.

Recent patchification scaling studies push this observation further. Wang et al. [Wang et al., 2025b] systematically reduce the patch size from the standard coarse-token regime toward pixel-level tokenization and report smooth gains across classification, semantic segmentation, object detection, and instance segmentation. Table 3 summarizes the results most relevant to our discussion. The most informative part is their sequence-length ablation. When coarse patch tokens are merely interpolated into a longer sequence, the gains remain small. When genuinely smaller patches are extracted from the image, the gains are much larger. This suggests that patch-size scaling changes the local visual primitives exposed to the model, not only the number of tokens. In the notation of this paper, reducing patch size changes the initial patch matrix itself: it lowers spatial compression in each local vector and changes the multiset of local rows through which the network forms its updates. This aligns with the gradient factorization $\nabla_{\mathbf{W}}\mathcal{L} = \mathbf{G}^{\top}\mathbf{X}$: the relevant object is the geometry of the rows in the patch or token matrix, not only the nominal sequence length. Related pixel-token studies push the same question to the extreme by asking how far visual models can go when patchification is removed almost entirely [Nguyen et al., 2024].

Why small patches require hierarchy and mixing. Smaller patches improve the local geometry exposed to the model, but they also increase the burden of mixing information across locations. This tradeoff helps explain a major trend in vision-transformer architecture design. Hierarchical Transformers such as Swin do not keep a fixed high-dimensional token space throughout the network; they begin with finer local tokens, restrict early mixing to local windows, and then gradually merge tokens while increasing representation dimension [Liu et al., 2021b]. PVTv2 follows a related logic through overlapping patch embeddings, linear-complexity attention, and convolutional

feed-forward layers [Wang et al., 2022]. CvT introduces convolutional token embeddings and convolutional projections inside Transformer blocks [Wu et al., 2021]. Tokens-to-Token ViT replaces one-shot patchification by a progressive tokenization module that recursively aggregates local neighboring tokens [Yuan et al., 2021]. These designs are not only remedies for a defective large-patch ViT. They can be read as different ways of managing the same geometric tradeoff: expose early layers to controlled local vectors, then recover expressivity through hierarchy, overlap, convolutional projection, or stage-wise mixing.

In the notation of (10), a deep network replaces the fixed raw patch matrix $\mathbf{X}$ by a sequence of representation matrices. At layer ℓ, one should imagine a matrix

$$\mathbf{X}_\ell = \left[\boldsymbol{z}_{\ell,j}(\boldsymbol{x}_i)^\mathsf{T}\right]_{(i,j)}$$

whose rows are local tokens or local feature vectors produced by the preceding layers. Backpropagation through a local operator at layer ℓ again has the schematic form

$$\nabla_{\mathbf{W}_\ell}\mathcal{L} \approx \mathbf{G}_\ell(\boldsymbol{\theta})^\mathsf{T}\mathbf{X}_\ell,$$

up to the additional Jacobian factors introduced by normalization, residual connections, attention, and nonlinear mixing. Thus the relevant geometry is no longer fixed by the raw data distribution. It evolves during training. Patch size, windowing, overlap, token merging, and convolutional stems can all be viewed as mechanisms for shaping the sequence

$$\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_L$$

of optimization-facing geometries.

Patch geometry in CNNs: from spatial-channel mixing to depthwise convolutional kernel. CNN architecture design provides a sequence of examples in which the vector seen by each local operator is progressively controlled. In a residual bottleneck block, the 3×3 spatial convolution acts after a channel-reducing 1×1 projection, so the spatial operator does not see the full channel dimension of the block [He et al., 2016]. ResNeXt makes this control explicit through grouped convolutions: each group applies a 3×3 operator to only a fraction of the channels, reducing the local vector from $3 \times 3 \times C_{\text{in}}$ to $3 \times 3 \times C_{\text{in}}/g$ [Xie et al., 2017]. ConvNeXt pushes the same principle further by using depthwise spatial convolutions inside an inverted-bottleneck block, with expressive channel mixing supplied by pointwise layers [Liu et al., 2022]. Thus, from ResNet to ResNeXt to ConvNeXt, the spatial operator is increasingly restricted to a structured local geometry, while expressivity is recovered through channel mixing and depth.

The kernel-size ablation in ConvNeXt gives a useful empirical instance of the trade-off suggested by our theory. Increasing the depthwise kernel from 3×3 to 7×7 improves ImageNet accuracy, while increasing it further to 9×9 or 11×11 yields little additional gain [Liu et al., 2022]. In the language of this paper, enlarging the spatial kernel increases the context available to each local operator, reducing the bias induced by overly local processing. At the same time, a larger local vector exposes gradient descent to a richer patch geometry, which can weaken the regularizing effect of locality. The saturation around 7×7 is consistent with a bias–variance trade-off for the geometry seen by local operators: enough spatial support improves representation, while excessive support brings less benefit once the local computation is already sufficiently expressive.

RepLKNet develops this direction further by treating very large kernels as a separate architectural scaling dimension [Ding et al., 2022]. It shows that replacing ConvNeXt's 7×7 kernels with much larger stage-wise kernels can further improve performance, especially on dense prediction tasks. The key point for our purposes is not only the larger spatial support, but the training-time parameterization used to make such kernels effective. RepLKNet does not train a bare large kernel. It introduces *small-kernel structural re-parameterization*: during training, a large kernel branch is paired with a small kernel branch, such as a 3×3 or 5×5 branch. After training, the small kernel is padded to the center of the large kernel and algebraically merged, so inference uses a single large kernel.

This mechanism has a direct interpretation in our gradient-geometry framework. Let $\boldsymbol{x}_L \in \mathbb{R}^{K^2}$ denote the large local patch and let $\boldsymbol{x}_s = \boldsymbol{C}\boldsymbol{x}_L \in \mathbb{R}^{s^2}$ be the centered small patch selected from it. Ignoring normalization for clarity, the training-time operator has the form

$$h(\boldsymbol{x}) = \boldsymbol{w}_L^\mathsf{T}\boldsymbol{x}_L + \boldsymbol{w}_s^\mathsf{T}\boldsymbol{x}_s = (\boldsymbol{w}_L + \boldsymbol{C}^\mathsf{T}\boldsymbol{w}_s)^\mathsf{T}\boldsymbol{x}_L, \qquad \boldsymbol{w}_{\text{eff}} = \boldsymbol{w}_L + \boldsymbol{C}^\mathsf{T}\boldsymbol{w}_s.$$

At inference time this is just a large kernel with weight $\boldsymbol{w}_{\text{eff}}$. During training, however, the two branches expose different local geometries to gradient descent. If r denotes the backpropagated scalar signal, then

$$\nabla_{\boldsymbol{w}_L}\mathcal{L} = r\boldsymbol{x}_L, \qquad \nabla_{\boldsymbol{w}_s}\mathcal{L} = r\boldsymbol{C}\boldsymbol{x}_L, \qquad \Delta\boldsymbol{w}_{\text{eff}} = -\eta r(\boldsymbol{I} + \boldsymbol{C}^{\mathsf{T}}\boldsymbol{C})\boldsymbol{x}_L,$$

up to the branch-dependent scaling introduced by normalization. The small branch therefore gives the centered local subspace an additional optimization path. It biases training toward small-scale local structure while the large branch captures broad spatial context.

This example sharpens the message of the patch-matrix factorization $\nabla_{\mathbf{W}}\mathcal{L} = \mathbf{G}^{\mathsf{T}}\mathbf{X}$. The geometry relevant to learning is not determined solely by the inference-time operator. RepLKNet uses an inference-time large kernel, but its training-time architecture exposes gradient descent to both a large-patch geometry and a centered small-patch geometry. Thus structural re-parameterization is a practical mechanism for modifying the optimization-facing patch geometry without changing the final operator class. Together with the ConvNeXt ablation, this supports the broader view that CNN design balances spatial context against implicit regularization induced by controlled local geometry.

B Functional Analysis of Shallow ReLU Networks

B.1 Path-norm and Variation Semi-norm of ReLU Networks

This subsection collects several basic facts from [Parhi and Nowak, 2023] and [Siegel and Xu, 2023] that we will use later.

Definition B.1. Let $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k\,\phi(\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x} - b_k) + \beta$ be a fully-connected two-layer neural network. The (unweighted) path-norm of $f_{\boldsymbol{\theta}}$ is defined to be

$$\|f_{\boldsymbol{\theta}}\|_{\text{path}} := \sum_{k=1}^{K} |v_k|\,\|\boldsymbol{w}_k\|_2\,. \tag{11}$$

Dictionary representation of ReLU networks. Using the positive 1-homogeneity of ReLU, one may rescale each hidden unit while leaving the realized function unchanged:

$$v_k\,\phi(\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x} - b_k) = a_k\,\phi(\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{x} - t_k), \quad \boldsymbol{u}_k := \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|_2} \in \mathbb{S}^{d-1},\ t_k := \frac{b_k}{\|\boldsymbol{w}_k\|_2},\ a_k := v_k\,\|\boldsymbol{w}_k\|_2\,.$$

Consequently, f_θ can be written in the normalized finite-sum form

$$f(\boldsymbol{x}) = \sum_{k=1}^{K'} a_k\,\phi(\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{x} - t_k) + \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + c_0. \tag{12}$$

Define the (ReLU) ridge dictionary as $\mathscr{D}_\phi := \big\{\phi(\boldsymbol{u}^{\mathsf{T}}\cdot - t) :\ \boldsymbol{u} \in \mathbb{S}^{d-1},\ t \in \mathbb{R}\big\}$. We focus on the *overparameterized, width-agnostic* collection obtained by taking the *union over all finite widths*

$$\mathcal{F}_{\text{fin}} := \bigcup_{K\geq 1} \left\{ \sum_{k=1}^{K} a_k\,\phi(\boldsymbol{u}_k^{\mathsf{T}}\cdot - t_k) + \boldsymbol{c}^{\mathsf{T}}(\cdot) + c_0 \right\}, \tag{13}$$

and quantify complexity via the smallest path-norm among all realizations of f:

$$\|f\|_{\text{path,min}} := \inf\left\{\|f_{\boldsymbol{\theta}}\|_{\text{path}} :\ f_{\boldsymbol{\theta}} \equiv f \text{ of the form (12)}\right\}.$$

From finite sums to a *width-agnostic* integral representation. Rather than fixing a particular width K, it is convenient to work with a convex, measure-theoretic formulation that *captures the closure/convex hull of* (13). Concretely, let ν be a finite signed Radon measure on $\mathbb{S}^{d-1} \times [-R, R]$ and consider

$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}\times[-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x} - t)\ \mathrm{d}\nu(\boldsymbol{u}, t) + c^{\mathsf{T}}\boldsymbol{x} + c_0. \tag{14}$$

Every finite network of the form (12) corresponds to an *atomic* (hence sparse) measure $\nu = \sum_{k=1}^{K} a_k\,\delta_{(\boldsymbol{u}_k, t_k)}$, and conversely any such atomic ν yields a finite network. Therefore, (14) should be viewed as a *width-agnostic relaxation* aligned with (11), rather than as an assumption of an infinite-width limit.

Definition B.2. The (unweighted) variation (semi)norm

$$|f|_{\mathrm{V}} := \inf \left\{ \|\nu\|_{\mathcal{M}} : \ f \text{ admits (12) for some } (\nu, c, c_0) \right\}, \tag{15}$$

where $\|\nu\|_{\mathcal{M}}$ is the total variation of ν.

For the compact region $\Omega = \mathbb{B}_R^d$, we define the bounded variation function class as

$$\mathrm{V}_C(\Omega) := \left\{ f \colon \Omega \to \mathbb{R} \mid f = \int_{\mathbb{S}^{d-1} \times [-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}} \boldsymbol{x} - t) \, \mathrm{d}\nu(\boldsymbol{u}, t) + \boldsymbol{c}^{\mathsf{T}} \boldsymbol{x} + b, \, |f|_{\mathrm{V}} \leq C \right\}. \tag{16}$$

In particular, identifying (12) with the atomic measure $\nu = \sum_k a_k \delta_{(\boldsymbol{u}_k, t_k)}$ yields

$$|f|_{\mathrm{V}} \leq \sum_k |a_k| = \|f_{\boldsymbol{\theta}}\|_{\mathrm{path}}, \quad \text{hence} \quad |f|_{\mathrm{V}} \leq \|f\|_{\mathrm{path,min}}.$$

Moreover, the minimal total variation required to represent f coincides with the minimal path-norm over all finite decompositions:

$$\|f\|_{\mathrm{path,min}} = |f|_{\mathrm{V}}. \tag{17}$$

Thus, (15) provides a *nonparametric* analogue of the path-norm: it encodes the same complexity notion while *not committing to a fixed width* K.

Remark B.3 ("Arbitrary width" $\neq$ "infinite width")**.** All statements here are about $\mathcal{F}_{\mathrm{fin}}$ in (13), namely networks of *finite* (but unconstrained) width. The integral representation (14) is introduced as a convenient convexification/closure of this union for analysis and regularization; it *does not* posit an infinite-width limit. In particular, when training in the variational form with a total-variation penalty on ν, first-order optimality implies that optimal measures are sparse (i.e., have finite support), which corresponds exactly to *finite-width* networks. Therefore, our results hold for *arbitrary (yet finite) width*; the continuum measure serves only as a tool to characterize and control $\|f\|_{\mathrm{path,min}}$.

B.2 The Metric Entropy of Variation Spaces

Metric entropy is a standard way to describe how "compact" a subset A is inside a metric space (X, ρ_X). We recall the notions of covering numbers and metric entropy.

Definition B.4 (Covering Number and Entropy)**.** Let A be a compact subset of a metric space (X, ρ_X). For $t > 0$, the *covering number* $N(A, t, \rho_X)$ is the smallest number of closed balls of radius t whose union contains A:

$$N(t, A, \rho_X) := \min \left\{ N \in \mathbb{N} : \exists \, x_1, \ldots, x_N \in X \text{ s.t. } A \subset \bigcup_{i=1}^{N} \mathbb{B}(x_i, t) \right\}, \tag{18}$$

where $\mathbb{B}(x_i, t) = \{y \in X : \rho_X(y, x_i) \leq t\}$. The *metric entropy* of A at scale t is then

$$H_t(A)_X := \log N(t, A, \rho_X). \tag{19}$$

Covering/entropy bounds for bounded-variation-type classes have been established in the literature. In what follows, we will use the estimate stated below.

Proposition B.5 (Parhi and Nowak 2023, Appendix D)**.** *The metric entropy of* $\mathrm{V}_C(\mathbb{B}_R^d)$ *(see Definition B.2) with respect to the* $L^\infty(\mathbb{B}_R^d)$*-distance* $\|\cdot\|_\infty$ *satisfies*

$$\log N(t, \mathrm{V}_C(\mathbb{B}_R^d), \|\cdot\|_\infty) \lessapprox_d \left(\frac{C}{t} \right)^{\frac{2d}{d+3}}. \tag{20}$$

where $\lessapprox_d$ *hides constants (which could depend on* d*) and logarithmic factors.*

B.3 Generalization Gap of Unweighted Variation Function Class

As a middle step towards bounding the generalization gap of the weighted variation function class, we bound the generalization gap of the unweighted variation function class using chaining and Gaussian complexity, together with the L^∞ metric entropy bound in Proposition B.5.

Proposition B.6 (Wainwright 2019, Chapter 13)**.** *Fix design points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ and denote the empirical norm*

$$\|f\|_n^2 := \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)^2.$$

Let $\mathcal{F}$ be a class of real-valued functions on $\{\boldsymbol{x}_i\}_{i=1}^n$, and define

$$\widehat{\mathcal{G}}_n(\mathcal{F}) := \sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i f(\boldsymbol{x}_i), \qquad \varepsilon_1, \dots, \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \quad \mathcal{G}_n(\mathcal{F}) := \mathbb{E}\,\widehat{\mathcal{G}}_n(\mathcal{F}).$$

Then

$$\mathcal{G}_n(\mathcal{F}) \;\lesssim\; \frac{16}{\sqrt{n}} \int_0^{\operatorname{diam}(\mathcal{F},\|\cdot\|_n)} \sqrt{\log N\big(t,\ \mathcal{F},\ \|\cdot\|_n\big)}\ \mathrm{d}t, \tag{21}$$

where $\operatorname{diam}(\mathcal{F}, \|\cdot\|_n) := \sup_{f_1,f_2\in\mathcal{F}} \|f_1 - f_2\|_n$*. Moreover, with probability at least $1-\delta$,*

$$\widehat{\mathcal{G}}_n(\mathcal{F}) \;\le\; \mathcal{G}_n(\mathcal{F}) + \operatorname{diam}(\mathcal{F}, \|\cdot\|_n)\,\frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} \qquad (\delta > 0). \tag{22}$$

Lemma B.7. *Let $\mathcal{F}_{M,C} = \{f \in \mathrm{V}_C(\mathbb{B}_R^d) \mid \|f\|_\infty \le M\}$ with $M \ge D$, and let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \sim \mathcal{P}^{\otimes n}$ where $|Y| \le D$ a.s. Then with probability at least $1-\delta$,*

$$\sup_{f\in\mathcal{F}_{M,C}} \big|R(f) - \widehat{R}_{\mathcal{D}}(f)\big| \;\lesssim_d\; C^{\frac{d}{d+3}}\, M^{\frac{d+6}{d+3}}\, n^{-\frac{1}{2}} \;+\; M^2\left(\frac{\log(1/\delta)}{n}\right)^{\frac{1}{2}}. \tag{23}$$

Proof. Let $\ell_f(\boldsymbol{x}, y) := (y - f(\boldsymbol{x}))^2$ and $\mathcal{L}_{M,C} := \{\ell_f : f \in \mathcal{F}_{M,C}\}$. Since $|y| \le D \le M$ and $\|f\|_\infty \le M$, for any $f, g \in \mathcal{F}_{M,C}$ and any $(\boldsymbol{x}, y)$,

$$\big|\ell_f(\boldsymbol{x},y) - \ell_g(\boldsymbol{x},y)\big| = |f(\boldsymbol{x}) - g(\boldsymbol{x})|\,|f(\boldsymbol{x}) + g(\boldsymbol{x}) - 2y| \le 4M\,|f(\boldsymbol{x}) - g(\boldsymbol{x})|.$$

Therefore, for the empirical norm on $\{\boldsymbol{x}_i\}_{i=1}^n$,

$$\|\ell_f - \ell_g\|_n \le 4M\,\|f - g\|_n \le 4M\,\|f - g\|_\infty, \tag{24}$$

and $\operatorname{diam}(\mathcal{L}_{M,C}, \|\cdot\|_n) \le 8M^2$.

A standard Gaussian symmetrization/concentration argument (see, e.g., Wainwright 2019, Chapter 5) yields that with probability at least $1-\delta$,

$$\sup_{f\in\mathcal{F}_{M,C}} \big|R(f) - \widehat{R}_{\mathcal{D}}(f)\big| \;\lesssim\; \widehat{\mathcal{G}}_n(\mathcal{L}_{M,C}) \;+\; M^2\left(\frac{\log(1/\delta)}{n}\right)^{\frac{1}{2}}. \tag{25}$$

It remains to bound $\widehat{\mathcal{G}}_n(\mathcal{L}_{M,C})$ by chaining. By (24), any t-cover of $\mathcal{F}_{M,C}$ in $\|\cdot\|_n$ induces a $(4Mt)$-cover of $\mathcal{L}_{M,C}$ in $\|\cdot\|_n$, hence

$$\log N\big(t, \mathcal{L}_{M,C}, \|\cdot\|_n\big) \le \log N\Big(\frac{t}{4M}, \mathcal{F}_{M,C}, \|\cdot\|_n\Big) \le \log N\Big(\frac{t}{4M}, \mathrm{V}_C(\mathbb{B}_R^d), \|\cdot\|_\infty\Big),$$

where we used $\|h\|_n \le \|h\|_\infty$. Proposition B.5 then gives, up to logarithmic factors,

$$\log N\big(t, \mathcal{L}_{M,C}, \|\cdot\|_n\big) \;\lessapprox_d\; \left(\frac{MC}{t}\right)^{\frac{2d}{d+3}}.$$

Applying Proposition B.6 with $\mathcal{G} = \mathcal{L}_{M,C}$ and $\operatorname{diam}(\mathcal{L}_{M,C}, \|\cdot\|_n) \le 8M^2$ yields

$$\begin{aligned}\widehat{\mathcal{G}}_n(\mathcal{L}_{M,C}) \;&\lesssim_d\; \frac{1}{\sqrt{n}}\int_0^{8M^2} \left(\frac{MC}{t}\right)^{\frac{d}{d+3}} \mathrm{d}t \;+\; M^2\left(\frac{\log(1/\delta)}{n}\right)^{\frac{1}{2}} \\ &\asymp_d\; C^{\frac{d}{d+3}}\, M^{\frac{d+6}{d+3}}\, n^{-\frac{1}{2}} \;+\; M^2\left(\frac{\log(1/\delta)}{n}\right)^{\frac{1}{2}}.\end{aligned}$$

Combining with (25) proves (23). □

C Proof of Theorem 4.1

Let $\iota_j : \mathbb{R}^m \to \mathbb{R}^d$ be the dual embedding such that $\pi_j \circ \iota_j = \mathrm{id}_{\mathbb{R}^m}$. Then (1) is equivalent to a fully connected neural network in a form of

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) \;=\; \sum_{k=1}^{K} \frac{1}{J} \sum_{j=1}^{J} v_k \, \phi\big(\iota_j(\boldsymbol{w}_k)^{\mathsf{T}} \boldsymbol{x} - b_k\big) + \beta. \tag{26}$$

Note that for any fixed k, $\|\iota_j(\boldsymbol{w}_k)\|_2 = \|\boldsymbol{w}_k\|_2$ for all $j = 1, \cdots, J$. Therefore, the notion of path norm and variation norm (together with their weighted version) still make sense for the CNN model

$$\|f_{\boldsymbol{\theta}}\|_{\mathrm{path}} = \frac{1}{J} \sum_{k=1}^{K} \sum_{j=1}^{J} |v_k| \, \|\iota_j(\boldsymbol{w}_k)\|_2 = \sum_{k=1}^{K} |v_k| \, \|\boldsymbol{w}_k\|_2 \,. \tag{27}$$

By direct computation, the Hessian matrix of the loss function is expressed as

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L} = \underbrace{\frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i) \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)^{\mathsf{T}}}_{\boldsymbol{T}_{\mathcal{D}}} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i) \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)}_{\boldsymbol{R}_{\mathcal{D}}} . \tag{28}$$

Definition C.1. Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and a set of local receptive fields $\mathcal{S} = \{S_j\}_{j=1}^J$, we define a random vector $\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}}$ uniformly drawn from $\{\pi_j(\boldsymbol{x}_i)\}_{(i,j)}^{n \times J} \subset \mathbb{R}^m$. For any $\boldsymbol{u} \in \mathbb{S}^{m-1}, t \in \mathbb{R}$, we define the weight function by

$$g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) := \mathbb{E}\left[\phi\left(\boldsymbol{u}^{\mathsf{T}} \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} - t\right)\right] \sqrt{\mathbb{P}\left(\boldsymbol{u}^{\mathsf{T}} \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t\right)^2 + \left\| \mathbb{E}\left[\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} \mathbb{1}\left\{\boldsymbol{u}^{\mathsf{T}} \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t\right\}\right] \right\|_2^2}. \tag{29}$$

Proposition C.2. *Given a data set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and a network model* (1) *with local receptive fields $\mathcal{S}$. Then we have*

$$\lambda_{\max}\left(\boldsymbol{T}_{\mathcal{D}}\right) \geq 1 + 2 \sum_{k=1}^{K} |v_k| \, \|\boldsymbol{w}_k\| \; g_{\mathcal{D},\mathcal{S}}\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\right). \tag{30}$$

Proof. We write $\boldsymbol{T}_{\mathcal{D}}$ in terms of tangent features

$$\boldsymbol{T}_{\mathcal{D}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i) \, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)^{\mathsf{T}} \;=\; \frac{1}{n} \Phi \Phi^{\mathsf{T}}.$$

Consequently,

$$\lambda_{\max}\big(\boldsymbol{T}_{\mathcal{D}}\big) \;=\; \max_{u \in \mathbb{S}^{n-1}} \frac{1}{n} \|\Phi u\|^2 \;\geq\; \frac{1}{n^2} \|\Phi \mathbf{1}\|^2. \tag{31}$$

For any point $\boldsymbol{x}$ and abbreviate its patch extraction on S_j by $\boldsymbol{x}^{(S_j)} := \pi_j(\boldsymbol{x})$. Define the ReLU gate

$$m_k^{(S_j)}(\boldsymbol{x}) := \mathbb{1}\left\{\boldsymbol{w}_k^{\mathsf{T}} \boldsymbol{x}^{(S_j)} > b_k\right\}.$$

For any sample $\boldsymbol{x}_i$, denote $m_{k,i}^{(S_j)} := m_k^{(S_j)}(\boldsymbol{x}_i)$. Then the partial derivatives are

$$\frac{\partial f(\boldsymbol{x}_i)}{\partial v_k} = \frac{1}{J} \sum_{j=1}^{J} m_{k,i}^{(S_j)} \cdot \big(\boldsymbol{w}_k^{\mathsf{T}} \boldsymbol{x}_i^{(S_j)} - b_k\big), \quad \frac{\partial f(\boldsymbol{x}_i)}{\partial \boldsymbol{w}_k} = \frac{1}{J} \sum_{j=1}^{J} m_{k,i}^{(S_j)} \cdot v_k \cdot \boldsymbol{x}_i^{(S_j)},$$

$$\frac{\partial f(\boldsymbol{x}_i)}{\partial b_k} = -\frac{1}{J} \sum_{j=1}^{J} m_{k,i}^{(S_j)} \cdot v_k, \quad \frac{\partial f(\boldsymbol{x}_i)}{\partial \beta} = 1.$$

Stacking these over i and plugging $u = \mathbf{1}/\sqrt{n}$ in (31), we get

$$\begin{aligned}
\frac{1}{n^2}\|\Phi\mathbf{1}\|^2 &= 1 + \frac{1}{n^2}\sum_{k=1}^{K}\Bigg[(v_k)^2\left(\left\|\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\boldsymbol{x}_i^{(S_j)}\right\|^2 + \left(\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\right)^2\right) \\
&\qquad + \left(\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J}\phi\left(\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x}_i^{(S_j)} - b_k\right)\right)^2\Bigg] \\
&= 1 + \frac{1}{n^2}\sum_{k=1}^{K}\Bigg[(v_k)^2\left(\left\|\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\boldsymbol{x}_i^{(S_j)}\right\|^2 + \left(\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\right)^2\right) \\
&\qquad + \|\boldsymbol{w}_k\|_2^2\left(\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J}\phi\left(\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{x}_i^{(S_j)} - t_k\right)\right)^2\Bigg] \qquad (32) \\
&\geq 1 + \frac{2}{n^2}\sum_{k=1}^{K}|v_k|\,\|\boldsymbol{w}_k\|_2\left(\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J}\phi\left(\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{x}_i^{(S_j)} - t_k\right)\right) \\
&\qquad \cdot\sqrt{\left\|\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\boldsymbol{x}_i^{(S_j)}\right\|^2 + \left(\sum_{i=1}^{n}\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\right)^2} \quad (\text{since } a^2+b^2 \geq 2ab) \\
&= 1 + 2\sum_{k=1}^{K}|v_k|\,\|\boldsymbol{w}_k\|_2\left(\frac{1}{nJ}\sum_{i=1}^{n}\sum_{j=1}^{J}\phi\left(\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{x}_i^{(S_j)} - t_k\right)\right) \\
&\qquad \cdot\sqrt{\left\|\frac{1}{nJ}\sum_{i=1}^{n}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\boldsymbol{x}_i^{(S_j)}\right\|^2 + \left(\frac{1}{nJ}\sum_{i=1}^{n}\sum_{j=1}^{J} m_{k,i}^{(S_j)}\right)^2} \\
&= 1 + 2\sum_{k=1}^{K}|v_k|\,\|\boldsymbol{w}_k\|_2\,\mathbb{E}\left[\phi\left(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} - t\right)\right] \\
&\qquad \cdot\sqrt{\mathbb{P}\left(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t\right)^2 + \left\|\mathbb{E}\left[\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}}\,\mathbb{1}\left\{\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t\right\}\right]\right\|_2^2} \\
&= 1 + 2\sum_{k=1}^{K}|v_k|\,\|\boldsymbol{w}_k\|_2\,g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}_k, t_k), \qquad \boldsymbol{u}_k = \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|_2}, \quad t_k = \frac{b_k}{\|\boldsymbol{w}_k\|_2}.
\end{aligned}$$

Then combining (31) and the above inequality yields the claim. □

Lemma C.3. *Consider the model* (1)

$$f(\boldsymbol{x}) \;=\; \sum_{k=1}^{K}\frac{v_k}{J}\sum_{j=1}^{J}\phi\left(\boldsymbol{w}_k^{\mathsf{T}}\pi_j(\boldsymbol{x}) - b_k\right) + \beta.$$

where the input satisfies $\|\boldsymbol{x}\|_2 \leq R$, each patch extractor $\pi_j : \mathbb{R}^d \to \mathbb{R}^m$ is a coordinate projection, and $\phi(t) = \max\{0, t\}$. Let $\boldsymbol{\theta} = \left(\boldsymbol{w}_1^{\mathsf{T}}, \ldots, \boldsymbol{w}_K^{\mathsf{T}}, b_1, \ldots, b_K, v_1, \ldots, v_K, \beta\right)^{\mathsf{T}}$ collect all parameters. Assume $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ is twice differentiable with respect to $\boldsymbol{\theta}$ at $\boldsymbol{x}$, i.e., for all k and $S_j \in \mathcal{S}$ we have $\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x}^{(S_j)} \neq b_k$. Then for any perturbation vector $\boldsymbol{\omega}$ with $\|\boldsymbol{\omega}\|_2 = 1$, it holds that

$$\left|\boldsymbol{\omega}^{\mathsf{T}}\,\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\,\boldsymbol{\omega}\right| \;\leq\; 2\,(R+1).$$

Proof. Write $\boldsymbol{\theta} = \left(\boldsymbol{w}_1^{\mathsf{T}}, \ldots, \boldsymbol{w}_K^{\mathsf{T}}, b_1, \ldots, b_K, v_1, \ldots, v_K, \beta\right)^{\mathsf{T}}$. The total number of parameters is $N = K \cdot m + K + K + 1 = K(m+2) + 1$.

Let the corresponding perturbation vector be

$$\boldsymbol{\omega} = \big(\boldsymbol{\alpha}_1^\mathsf{T}, \ldots, \boldsymbol{\alpha}_K^\mathsf{T},\, \delta_1, \ldots, \delta_K,\, \gamma_1, \ldots, \gamma_K,\, \iota\big)^\mathsf{T} \in \mathbb{R}^N,$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^m$ corresponds to $\boldsymbol{w}_k$, $\delta_k \in \mathbb{R}$ to b_k, $\gamma_k \in \mathbb{R}$ to v_k, and $\iota \in \mathbb{R}$ to β. The normalization constraint is

$$\|\boldsymbol{\omega}\|_2^2 = \sum_{k=1}^K \|\boldsymbol{\alpha}_k\|_2^2 + \sum_{k=1}^K \delta_k^2 + \sum_{k=1}^K \gamma_k^2 + \iota^2 = 1.$$

For the fixed input $\boldsymbol{x}$, set $\boldsymbol{x}^{(S_j)} := \pi_j(\boldsymbol{x}) \in \mathbb{R}^m$ and define the ReLU gate

$$m_k^{(S_j)} \;:=\; \mathbb{1}\big\{\boldsymbol{w}_k^\mathsf{T} \boldsymbol{x}^{(S_j)} > b_k\big\} \in \{0, 1\}.$$

By the twice-differentiability assumption, all gates are constant in a neighborhood of $\boldsymbol{\theta}$.

Within this gate-fixed region, $f_{\boldsymbol{\theta}}$ is affine in $(\boldsymbol{w}_k, b_k)$ once v_k is held fixed, and affine in v_k once $(\boldsymbol{w}_k, b_k)$ are held fixed. Therefore the only nonzero second partial derivatives inside the k-th neuron block are the mixed ones with v_k

$$\frac{\partial^2 f_{\boldsymbol{\theta}}}{\partial \boldsymbol{w}_k\, \partial v_k} = \frac{\partial}{\partial v_k}\left(\frac{1}{J}\sum_{S_j \in \mathcal{S}} v_k\, m_k^{(S_j)}\, \boldsymbol{x}^{(S_j)}\right) = \frac{1}{J}\sum_{j=1}^J m_k^{(S_j)}\, \boldsymbol{x}^{(S_j)} =: \boldsymbol{s}_k \in \mathbb{R}^m, \tag{33}$$

$$\frac{\partial^2 f_{\boldsymbol{\theta}}}{\partial b_k\, \partial v_k} = \frac{\partial}{\partial v_k}\left(\frac{1}{J}\sum_{j=1}^J v_k\, m_k^{(S_j)}\right) = -\frac{1}{J}\sum_{j=1}^J m_k^{(S_j)} =: t_k \in \mathbb{R}. \tag{34}$$

All other second derivatives inside the block vanish (as do any cross-neuron second derivatives and all those involving β). Hence, with the block variable $\theta_k := (\boldsymbol{w}_k^\mathsf{T},\, b_k,\, v_k)^\mathsf{T}$, we have

$$\nabla^2_{(\theta_k)} f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \begin{pmatrix} \mathbf{0}_{m\times m} & \mathbf{0}_m & \boldsymbol{s}_k \\ \mathbf{0}_m^\mathsf{T} & 0 & t_k \\ \boldsymbol{s}_k^\mathsf{T} & t_k & 0 \end{pmatrix}. \tag{35}$$

The full quadratic form splits over neuron blocks:

$$\begin{aligned} \boldsymbol{\omega}^\mathsf{T}\, \nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\, \boldsymbol{\omega} &= \sum_{k=1}^K \begin{pmatrix} \boldsymbol{\alpha}_k^\mathsf{T} & \delta_k & \gamma_k \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \boldsymbol{s}_k \\ \mathbf{0}^\mathsf{T} & 0 & t_k \\ \boldsymbol{s}_k^\mathsf{T} & t_k & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_k \\ \delta_k \\ \gamma_k \end{pmatrix} \\ &= \sum_{k=1}^K 2\,\gamma_k\big(\boldsymbol{\alpha}_k^\mathsf{T} \boldsymbol{s}_k + \delta_k\, t_k\big). \end{aligned} \tag{36}$$

Recall the definition of the notations (33) and (34),

$$|t_k| = \frac{1}{J}\sum_{j=1}^J m_k^{(S_j)} \le 1, \tag{37}$$

$$\|\boldsymbol{s}_k\|_2 = \left\| \frac{1}{J}\sum_{j=1}^J m_k^{(S_j)}\, \boldsymbol{x}^{(S_j)} \right\|_2 \le \frac{1}{J}\sum_{j=1}^J m_k^{(S_j)}\, \|\boldsymbol{x}^{(S_j)}\|_2 \le \frac{1}{J}\sum_{j=1}^J m_k^{(S_j)}\, R = R|t_k|, \tag{38}$$

because $\boldsymbol{x}^{(S_j)} = \pi_j(\boldsymbol{x})$ is a coordinate projection of $\boldsymbol{x}$, $\|\boldsymbol{x}^{(S_j)}\|_2 \le \|\boldsymbol{x}\|_2 \le R$.

Then for each k,

$$\begin{aligned} \big|2\,\gamma_k(\boldsymbol{\alpha}_k^\mathsf{T}\boldsymbol{s}_k + \delta_k\, t_k)\big| &\le 2\,|\gamma_k|\Big(\|\boldsymbol{\alpha}_k\|_2\, \|\boldsymbol{s}_k\|_2 + |\delta_k|\, t_k\Big) \quad \text{(Cauchy-Schwarz)} \\ (38) \Longrightarrow \quad &\le 2\,|\gamma_k|\Big(R\, t_k\, \|\boldsymbol{\alpha}_k\|_2 + |\delta_k|\, t_k\Big) \\ (37) \Longrightarrow \quad &\le 2\,|\gamma_k|\Big(R\, \|\boldsymbol{\alpha}_k\|_2 + |\delta_k|\Big), \end{aligned} \tag{39}$$

Summing over k for (39) and plug in (36), we have

$$\begin{aligned}\left|\boldsymbol{\omega}^{\mathsf{T}}\,\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\,\boldsymbol{\omega}\right| &\le 2\Big(R\sum_k |\gamma_k|\,\|\boldsymbol{\alpha}_k\|_2 + \sum_k |\gamma_k|\,|\delta_k|\Big)\\ \text{(Cauchy-Schwarz)} &\le 2\Big(R\sqrt{\textstyle\sum_k \gamma_k^2}\sqrt{\textstyle\sum_k \|\boldsymbol{\alpha}_k\|_2^2} + \sqrt{\textstyle\sum_k \gamma_k^2}\sqrt{\textstyle\sum_k \delta_k^2}\Big). \end{aligned} \tag{40}$$

Using the normalization $\sum_k \|\boldsymbol{\alpha}_k\|_2^2 + \sum_k \delta_k^2 + \sum_k \gamma_k^2 + \iota^2 = 1$, we have $\sqrt{\sum_k \gamma_k^2} \le 1$, $\sqrt{\sum_k \|\boldsymbol{\alpha}_k\|_2^2} \le 1$, and $\sqrt{\sum_k \delta_k^2} \le 1$. Thus $\left|\boldsymbol{\omega}^{\mathsf{T}}\,\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x})\,\boldsymbol{\omega}\right| \le 2\,(R+1)$. □

Theorem C.4 (Restate Theorem 4.1)**.** *Given a data set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and a network model* (1) *with the set of local receptive fields $\mathcal{S}$, we have*

$$\sum_{k=1}^K |v_k|\,\|\boldsymbol{w}_k\|\,g_{\mathcal{D},\mathcal{S}}\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\right) \;\le\; \frac{1}{2}\left(\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})) + 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})} - 1\right).$$

Proof. It suffices to prove the first assertion. Recall that by direct computation, the Hessian matrix of the loss function is expressed as

$$\nabla^2_{\boldsymbol{\theta}}\mathcal{L} = \underbrace{\frac{1}{n}\sum_{i=1}^n \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)^{\mathsf{T}}}_{\boldsymbol{T}_{\mathcal{D}}} + \underbrace{\frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - y_i)\nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)}_{\boldsymbol{R}_{\mathcal{D}}}. \tag{41}$$

Let $\boldsymbol{\omega}$ be the unit eigenvector (i.e., $\|\boldsymbol{\omega}\|_2 = 1$) corresponding to the largest eigenvalue of the matrix $\boldsymbol{T}_{\mathcal{D}}$, the maximum eigenvalue of the Hessian matrix of the loss can be lower-bounded as follows:

$$\begin{aligned}\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})) \;&\ge\; \boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\boldsymbol{\omega}\\ &\ge\; \underbrace{\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}})}_{\text{(Term A)}} + \underbrace{\frac{1}{n}\sum_{i=1}^n (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)\boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\boldsymbol{\omega}}_{\text{(Term B)}}. \end{aligned} \tag{42}$$

According to Proposition C.2, Term A is lower bounded by

$$\lambda_{\max}\left(\boldsymbol{T}_{\mathcal{D}}\right) \ge 1 + 2\sum_{k=1}^K |v_k|\,\|\boldsymbol{w}_k\|\,g_{\mathcal{D},\mathcal{S}}\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\right), \tag{43}$$

For (Term B), an upper bound can be established using the training loss $\mathcal{L}(\boldsymbol{\theta})$ via the Cauchy-Schwarz inequality. This also employs a notable uniform upper bound for $|\boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\boldsymbol{\omega}|$, as detailed in Lemma C.3:

$$|\text{(Term B)}| \le \sqrt{\frac{1}{n}\sum_{i=1}^n \left(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\right)^2}\cdot\sqrt{\frac{1}{n}\sum_{i=1}^n \left(\boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\boldsymbol{\omega}\right)^2} \le 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{44}$$

Thus, we have

$$\lambda_{\max}\left(\boldsymbol{T}_{\mathcal{D}}\right) \le \lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})) + |\text{(Term B)}| \le \lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})) + 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{45}$$

Finally, we plug (43) into (45)

$$\sum_{k=1}^K |v_k|\,\|\boldsymbol{w}_k\|\,g_{\mathcal{D},\mathcal{S}}\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\right) \;\le\; \frac{1}{2}\left(\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})) + 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})} - 1\right).$$

□

C.1 Empirical Process for the Weight Function

We study uniform deviations of the empirical weight function $g_{\mathcal{D},\mathcal{S}}$ from its population counterpart $g_{\mathcal{P},\mathcal{S}}$ under i.i.d. sampling $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim \mathcal{P}_{\boldsymbol{X}}$. Although the collection $\{\pi_j(\boldsymbol{x}_i)\}_{i,j}$ is not i.i.d. across (i, j) in general, for each fixed j the patches $\{\pi_j(\boldsymbol{x}_i)\}_{i=1}^n$ are i.i.d. in $\mathbb{R}^m$.

Definition C.5. For any $S_j \in \mathcal{S}$, let $\boldsymbol{X}_{\mathcal{D}}^{(S_j)}$ be a random vector drawn uniformly at random from the training examples $\{\pi_j(\boldsymbol{x}_i)\}_{i=1}^n$ with patch extraction to the local receptive field S_j. For any $\boldsymbol{u} \in \mathbb{S}^{m-1}, t \in \mathbb{R}$, define the aggregated gate probability, ReLU margin, and gated first moment

$$\begin{aligned}
\bar{p}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) &:= \frac{1}{J} \sum_{j=1}^{J} \mathbb{P}\left(\boldsymbol{u}^\mathsf{T} \boldsymbol{X}_{\mathcal{D}}^{(S_j)} > t\right), \\
\bar{r}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) &:= \frac{1}{J} \sum_{j=1}^{J} \mathbb{E}\left[\phi\left(\boldsymbol{u}^\mathsf{T} \boldsymbol{X}_{\mathcal{D}}^{(S_j)} - t\right)\right], \\
\bar{\boldsymbol{a}}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) &:= \frac{1}{J} \sum_{j=1}^{J} \mathbb{E}\left[\boldsymbol{X}_{\mathcal{D}}^{(S_j)} \mathbb{1}\left\{\boldsymbol{u}^\mathsf{T} \boldsymbol{X}_{\mathcal{D}}^{(S_j)} > t\right\}\right] \in \mathbb{R}^m.
\end{aligned} \tag{46}$$

Based on these quantities, define the empirical weight function

$$g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) := \bar{r}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) \sqrt{\bar{p}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t)^2 + \left\|\bar{\boldsymbol{a}}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t)\right\|_2^2}. \tag{47}$$

For each $j \in [J]$ and $(\boldsymbol{u}, t) \in \mathbb{S}^{m-1} \times [-1, 1]$, define the population components

$$\begin{aligned}
p_j(\boldsymbol{u}, t) &:= \mathbb{P}\big(\boldsymbol{u}^\mathsf{T} \pi_j(\boldsymbol{X}) > t\big), \\
r_j(\boldsymbol{u}, t) &:= \mathbb{E}\big[\phi(\boldsymbol{u}^\mathsf{T} \pi_j(\boldsymbol{X}) - t)\big], \\
\boldsymbol{a}_j(\boldsymbol{u}, t) &:= \mathbb{E}\big[\pi_j(\boldsymbol{X})\, \mathbb{1}\{\boldsymbol{u}^\mathsf{T} \pi_j(\boldsymbol{X}) > t\}\big].
\end{aligned}$$

Let $\bar{p}_{\mathcal{P},\mathcal{S}}$, $\bar{r}_{\mathcal{P},\mathcal{S}}$, and $\bar{\boldsymbol{a}}_{\mathcal{P},\mathcal{S}}$ denote their averages over $j \in [J]$. Define

$$g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u}, t) := \bar{r}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u}, t) \sqrt{\bar{p}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u}, t)^2 + \|\bar{\boldsymbol{a}}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u}, t)\|_2^2}.$$

Lemma C.6 (Complexity of the component classes)**.** *Under the constraint $\|\boldsymbol{z}\|_2 \le 1$, the following classes are uniformly bounded VC-subgraph classes with VC-subgraph dimension $O(m)$:*

$$\mathcal{H} := \left\{\boldsymbol{z} \mapsto \mathbb{1}\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} > t\} : \boldsymbol{u} \in \mathbb{S}^{m-1},\, t \in [-1, 1]\right\},$$

$$\mathcal{F}_{\mathrm{ReLU}} := \left\{\boldsymbol{z} \mapsto \phi(\boldsymbol{u}^\mathsf{T} \boldsymbol{z} - t) : \boldsymbol{u} \in \mathbb{S}^{m-1},\, t \in [-1, 1]\right\},$$

and

$$\mathcal{A} := \left\{\boldsymbol{z} \mapsto \boldsymbol{v}^\mathsf{T} \boldsymbol{z}\, \mathbb{1}\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} > t\} : \boldsymbol{u}, \boldsymbol{v} \in \mathbb{S}^{m-1},\, t \in [-1, 1]\right\}.$$

Moreover, $\mathcal{H}$ and $\mathcal{A}$ are bounded by 1, and $\mathcal{F}_{\mathrm{ReLU}}$ is bounded by 2.

Proof. The class $\mathcal{H}$ is the class of halfspace indicators in $\mathbb{R}^m$, hence has VC-dimension at most $m + 1$.

For $\mathcal{F}_{\mathrm{ReLU}}$, the subgraph of $\boldsymbol{z} \mapsto \phi(\boldsymbol{u}^\mathsf{T} \boldsymbol{z} - t)$ is

$$\{(\boldsymbol{z}, s) : s < \phi(\boldsymbol{u}^\mathsf{T} \boldsymbol{z} - t)\} = \big(\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} > t\} \cap \{s < \boldsymbol{u}^\mathsf{T} \boldsymbol{z} - t\}\big) \cup \big(\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} \le t\} \cap \{s < 0\}\big).$$

For $\mathcal{A}$, the subgraph of $\boldsymbol{z} \mapsto \boldsymbol{v}^\mathsf{T} \boldsymbol{z}\, \mathbb{1}\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} > t\}$ is

$$\{(\boldsymbol{z}, s) : s < \boldsymbol{v}^\mathsf{T} \boldsymbol{z}\, \mathbb{1}\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} > t\}\} = \big(\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} > t\} \cap \{s < \boldsymbol{v}^\mathsf{T} \boldsymbol{z}\}\big) \cup \big(\{\boldsymbol{u}^\mathsf{T} \boldsymbol{z} \le t\} \cap \{s < 0\}\big).$$

Both subgraph classes are constant-size Boolean combinations of halfspaces in $\mathbb{R}^{m+1}$. By the standard growth-function closure argument for finite Boolean combinations of VC classes, their VC dimensions are $O(m)$. Equivalently, their underlying real-valued function classes have pseudo-dimension $O(m)$, see [Anthony and Bartlett, 1999, Ch. 3 and Definition 11.2]. The boundedness claims follow from $\|\boldsymbol{z}\|_2 \le 1$, $t \in [-1, 1]$, and $\|\boldsymbol{u}\|_2 = \|\boldsymbol{v}\|_2 = 1$. □

Theorem C.7 (Uniform deviation for $g_{\mathcal{D},\mathcal{S}}$)**.** *Assume $\|\pi_j(\boldsymbol{X})\|_2 \le 1$ almost surely for all $j \in [J]$. There exists a universal constant $C_{\mathrm{ep}} > 0$ such that for every $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\sup_{\boldsymbol{u}\in\mathbb{S}^{m-1},\, t\in[-1,1]} \left|g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t) - g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\right| \le C_{\mathrm{ep}}\sqrt{\frac{m+\log(2J/\delta)}{n}}.$$

Proof. For each fixed $j \in [J]$, let P_j denote the law of $\boldsymbol{Z}_j := \pi_j(\boldsymbol{X}) \in \mathbb{R}^m$, and let $P_{n,j} := n^{-1}\sum_{i=1}^n \delta_{\pi_j(\boldsymbol{x}_i)}$. Define

$$\begin{aligned}
\hat{p}_j(\boldsymbol{u},t) &:= P_{n,j}\mathbb{1}\{\boldsymbol{u}^\mathsf{T}\boldsymbol{z} > t\},\\
\hat{r}_j(\boldsymbol{u},t) &:= P_{n,j}\phi(\boldsymbol{u}^\mathsf{T}\boldsymbol{z} - t),\\
\hat{\boldsymbol{a}}_j(\boldsymbol{u},t) &:= P_{n,j}\left[\boldsymbol{z}\,\mathbb{1}\{\boldsymbol{u}^\mathsf{T}\boldsymbol{z} > t\}\right].
\end{aligned}$$

By the VC-subgraph entropy bound and maximal inequality [van der Vaart and Wellner, 1996, Theorems 2.6.7 and 2.14.1], together with McDiarmid's inequality, any class $\mathcal{F}$ uniformly bounded by B with VC-subgraph dimension V satisfies

$$\sup_{f\in\mathcal{F}} |(P_n - P)f| \lesssim B\sqrt{\frac{V+\log(1/\eta)}{n}}$$

with probability at least $1-\eta$.

Applying this bound to $\mathcal{H}$ and $\mathcal{F}_{\mathrm{ReLU}}$ from Lemma C.6, for every fixed j and $\eta \in (0,1)$, with probability at least $1-\eta$,

$$\begin{aligned}
\sup_{\boldsymbol{u},t} |\hat{p}_j(\boldsymbol{u},t) - p_j(\boldsymbol{u},t)| &\lesssim \sqrt{\frac{m+\log(1/\eta)}{n}},\\
\sup_{\boldsymbol{u},t} |\hat{r}_j(\boldsymbol{u},t) - r_j(\boldsymbol{u},t)| &\lesssim \sqrt{\frac{m+\log(1/\eta)}{n}},
\end{aligned}$$

where all suprema are over $\boldsymbol{u} \in \mathbb{S}^{m-1}$ and $t \in [-1,1]$.

For the vector moment, use duality:

$$\begin{aligned}
\sup_{\boldsymbol{u},t} \|\hat{\boldsymbol{a}}_j(\boldsymbol{u},t) - \boldsymbol{a}_j(\boldsymbol{u},t)\|_2 &= \sup_{\boldsymbol{u},t}\sup_{\boldsymbol{v}\in\mathbb{S}^{m-1}} \left|\boldsymbol{v}^\mathsf{T}\big(\hat{\boldsymbol{a}}_j(\boldsymbol{u},t) - \boldsymbol{a}_j(\boldsymbol{u},t)\big)\right|\\
&= \sup_{\boldsymbol{u},\boldsymbol{v},t} \left|(P_{n,j} - P_j)\left[\boldsymbol{v}^\mathsf{T}\boldsymbol{z}\,\mathbb{1}\{\boldsymbol{u}^\mathsf{T}\boldsymbol{z} > t\}\right]\right|.
\end{aligned}$$

The last supremum is over the class $\mathcal{A}$ from Lemma C.6. Hence, with probability at least $1-\eta$,

$$\sup_{\boldsymbol{u},t} \|\hat{\boldsymbol{a}}_j(\boldsymbol{u},t) - \boldsymbol{a}_j(\boldsymbol{u},t)\|_2 \lesssim \sqrt{\frac{m+\log(1/\eta)}{n}}.$$

Taking $\eta = \delta/(3J)$ and applying a union bound over the three component bounds and over $j \in [J]$, we obtain that, with probability at least $1-\delta$, simultaneously for all $j \in [J]$,

$$\sup_{\boldsymbol{u},t} |\hat{p}_j(\boldsymbol{u},t) - p_j(\boldsymbol{u},t)| \le \varepsilon, \qquad \sup_{\boldsymbol{u},t} |\hat{r}_j(\boldsymbol{u},t) - r_j(\boldsymbol{u},t)| \le \varepsilon, \qquad \sup_{\boldsymbol{u},t} \|\hat{\boldsymbol{a}}_j(\boldsymbol{u},t) - \boldsymbol{a}_j(\boldsymbol{u},t)\|_2 \le \varepsilon,$$

where all suprema are over $\boldsymbol{u} \in \mathbb{S}^{m-1}$ and $t \in [-1,1]$, and $\varepsilon := C_0\sqrt{(m+\log(3J/\delta))/n}$ for a universal constant $C_0 > 0$. Averaging over j does not increase the deviations. Therefore, uniformly over $(\boldsymbol{u},t) \in \mathbb{S}^{m-1} \times [-1,1]$,

$$\begin{aligned}
&|\bar{p}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t) - \bar{p}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)| \le \varepsilon,\\
&|\bar{r}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t) - \bar{r}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)| \le \varepsilon,\\
&\|\bar{\boldsymbol{a}}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t) - \bar{\boldsymbol{a}}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\|_2 \le \varepsilon.
\end{aligned}$$

It remains to pass from the components to g. For both empirical and population quantities,

$$0 \le \bar{p} \le 1, \qquad 0 \le \bar{r} \le 2, \qquad \|\bar{\boldsymbol{a}}\|_2 \le \bar{p} \le 1.$$

Define $F(r,p,\boldsymbol{a}) := r\sqrt{p^2+\|\boldsymbol{a}\|_2^2}$. On this domain,

$$|F(r,p,\boldsymbol{a}) - F(r',p',\boldsymbol{a}')| \le \sqrt{2}\,|r-r'| + 2\sqrt{(p-p')^2+\|\boldsymbol{a}-\boldsymbol{a}'\|_2^2}.$$

Thus, uniformly over $(\boldsymbol{u},t) \in \mathbb{S}^{m-1}\times[-1,1]$,

$$\left|g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t) - g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\right| \lesssim \varepsilon \lesssim \sqrt{\frac{m+\log(2J/\delta)}{n}},$$

after adjusting the universal constant. This proves the theorem. □

C.2 The Computation of the Population Weight Functions for Uniform Distributions on Spheres

Assume that $\mathcal{P}_{\boldsymbol{X}}$ is a uniform distribution on $\mathbb{S}^{d-1} \subset \mathbb{R}^d$. Here we compute the population version of the components in the weighted function $g_{\mathcal{D},\mathcal{S}}$ defined in Definition C.5.

Let $d \ge 2$, $m < d$. Draw $\boldsymbol{X} \sim \mathrm{Uniform}(\mathbb{S}^{d-1}) \subset \mathbb{R}^d$, set $\boldsymbol{Z} = \mathrm{proj}(\boldsymbol{X}) \in \mathbb{R}^m$ by projecting the first m coordinates, and fix any unit $\boldsymbol{u} \in \mathbb{R}^m$. Define

$$Z := \boldsymbol{u}^{\mathsf{T}}\boldsymbol{Z} \in [-1,1]. \tag{48}$$

By rotational invariance in $\mathbb{R}^d$, Z has the onecoordinate marginal of $\mathrm{Uniform}(\mathbb{S}^{d-1})$, hence its density is

$$f_d(z) = c_d\,(1-z^2)^{\frac{d-3}{2}}\mathbb{1}\{-1 \le z \le 1\}, \qquad c_d := \frac{\Gamma\left(\frac{d}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\frac{d-1}{2}\right)}. \tag{49}$$

All bounds below are independent of m and $\boldsymbol{u}$, depending only on d.

For shorthand, write $\alpha := \frac{d-3}{2} \ge -\frac{1}{2}$ and observe

$$(1-z^2)^{\alpha} = \big[(1-z)(1+z)\big]^{\alpha}, \qquad z \in [-1,1], \tag{50}$$

together with the elementary bounds $1 \le 1+z \le 2$ for $z \in [0,1]$.

Lemma C.8 ((Tail of Z))**.** *For all $t \in [0,1)$,*

$$\frac{c_d}{\alpha+1}\,(1-t)^{\alpha+1} \;\le\; \mathbb{P}(Z>t) \;\le\; \frac{c_d\,2^{\alpha}}{\alpha+1}\,(1-t)^{\alpha+1}. \tag{51}$$

Consequently $\mathbb{P}(Z>t) \asymp (1-t)^{\frac{d-1}{2}}$ as $t \to 1^-$.

Proof. Using (49) and (50),

$$\mathbb{P}(Z>t) = \int_t^1 c_d\,(1-z^2)^{\alpha}\,dz = c_d\int_t^1 \big[(1-z)(1+z)\big]^{\alpha}\,dz. \tag{52}$$

Since $1 \le 1+z \le 2$ for $z \in [t,1]$, we have

$$c_d\int_t^1 (1-z)^{\alpha}\,dz \;\le\; \mathbb{P}(Z>t) \;\le\; c_d\,2^{\alpha}\int_t^1 (1-z)^{\alpha}\,dz. \tag{53}$$

Evaluating $\int_t^1 (1-z)^{\alpha}\,dz = \frac{(1-t)^{\alpha+1}}{\alpha+1}$ (valid for $\alpha > -1$; here $\alpha \ge -\frac{1}{2}$), we obtain (51). □

Lemma C.9. *For all $t \in [0,1)$,*

$$\frac{c_d}{(\alpha+1)(\alpha+2)}\,(1-t)^{\alpha+2} \;\le\; \mathbb{E}\big[\phi(Z-t)\big] \;\le\; \frac{c_d\,2^{\alpha}}{(\alpha+1)(\alpha+2)}\,(1-t)^{\alpha+2}. \tag{54}$$

Consequently $\mathbb{E}[\phi(Z-t)] \asymp (1-t)^{\frac{d+1}{2}}$ as $t \uparrow 1$.

Proof. By definition and (49),

$$\mathbb{E}[\phi(Z-t)] = \int_t^1 (z-t) f_d(z)\, dz = c_d \int_t^1 (z-t)\big[(1-z)(1+z)\big]^\alpha\, dz. \tag{55}$$

Bounding $1 \le 1+z \le 2$ for $z \in [t,1]$ yields

$$c_d \int_t^1 (z-t)(1-z)^\alpha\, dz \;\le\; \mathbb{E}[\phi(Z-t)] \;\le\; c_d\, 2^\alpha \int_t^1 (z-t)(1-z)^\alpha\, dz. \tag{56}$$

Substitute $y = 1-z$ to compute the shared integral:

$$\int_t^1 (z-t)(1-z)^\alpha\, dz = \int_0^{1-t} \big[(1-t)-y\big]\, y^\alpha\, dy = \frac{(1-t)^{\alpha+2}}{(\alpha+1)(\alpha+2)}, \tag{57}$$

valid for $\alpha > -1$. This gives (54). □

Proposition C.10. *There exist absolute constants (depending only on d)*

$$c_L(d) := \frac{c_d^2}{\big(\alpha+1\big)^2\big(\alpha+2\big)}, \qquad c_U(d) := \frac{c_d^2\, 2^{\,2\alpha}}{\big(\alpha+1\big)^2\big(\alpha+2\big)}, \tag{58}$$

such that for all $t \in [0,1)$,

$$c_L(d)\,(1-t)^{2\alpha+3} \;\le\; \mathbb{P}(Z>t)\cdot\mathbb{E}\big[\phi(Z-t)\big] \;\le\; c_U(d)\,(1-t)^{2\alpha+3}. \tag{59}$$

Equivalently, since $\alpha = \frac{d-3}{2}$,

$$\mathbb{P}(Z>t)\cdot\mathbb{E}\big[\phi(Z-t)\big] \asymp (1-t)^d \qquad \text{as } t \uparrow 1, \tag{60}$$

with explicit

$$c_L(d) = \frac{c_d^2}{\big(\frac{d-1}{2}\big)^2\big(\frac{d+1}{2}\big)}, \qquad c_U(d) = \frac{c_d^2\, 2^{\,d-3}}{\big(\frac{d-1}{2}\big)^2\big(\frac{d+1}{2}\big)}, \qquad c_d = \frac{\Gamma\big(\frac{d}{2}\big)}{\sqrt{\pi}\,\Gamma\big(\frac{d-1}{2}\big)}. \tag{61}$$

Proof. The inequality can be deduced by combining (51) and (54). □

Proposition C.11 (Boundary tail for projected radius)**.** *Fix integers $d \ge 2$ and $1 \le m < d$. Let $\boldsymbol{X} \sim \mathrm{Uniform}(\mathbb{S}^{d-1}) \subset \mathbb{R}^d$ and $\boldsymbol{Z} = \mathrm{proj}(\boldsymbol{X}) \in \mathbb{R}^m$. Then there exist universal constants*

$$c_L(d,m),\, c_U(d,m) \in (0,\infty)$$

depending only on d, m (and independent of the choice of projection and of t) such that for all $t \in (0, \frac{1}{4}]$,

$$c_L(d,m)\, t^{\frac{d-m}{2}} \;\le\; \mathbb{P}\big(\|\boldsymbol{Z}\| > 1-t\big) \;\le\; c_U(d,m)\, t^{\frac{d-m}{2}}. \tag{62}$$

In particular,

$$\mathbb{P}\big(\|\boldsymbol{Z}\| > 1-t\big) \asymp t^{\frac{d-m}{2}} \qquad \text{as } t \downarrow 0, \tag{63}$$

with one admissible choice

$$c_L(d,m) = \frac{2^{-|\frac{m}{2}-1|}}{\frac{d-m}{2}\, B\big(\frac{m}{2}, \frac{d-m}{2}\big)}, \qquad c_U(d,m) = \frac{2^{\,|\frac{m}{2}-1|+\frac{d-m}{2}}}{\frac{d-m}{2}\, B\big(\frac{m}{2}, \frac{d-m}{2}\big)}, \tag{64}$$

where $B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ denotes the Beta function.

Proof. Write $R := \|\boldsymbol{Z}\| \in [0,1]$ and $S := R^2$. A standard Gaussian-ratio representation gives

$$S \sim \mathrm{Beta}(a,b)\,, \qquad a := \frac{m}{2}, \quad b := \frac{d-m}{2}, \tag{65}$$

with density

$$f_S(x) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)}\, \mathbf{1}_{\{0<x<1\}}. \tag{66}$$

For $t \in (0,1)$,

$$\mathbb{P}(R > 1-t) = \mathbb{P}\big(S > (1-t)^2\big) = \frac{1}{B(a,b)} \int_{(1-t)^2}^{1} x^{a-1}(1-x)^{b-1}\, dx. \tag{67}$$

Set $y := 1-(1-t)^2 = 2t - t^2$. Then $y \in (0, 2t)$ and for $t \in (0, \frac{1}{4}]$ we have $y \le \frac{1}{2}$ and $(1-y) \ge \frac{1}{2}$. Since $x \in [1-y, 1]$, the elementary bound

$$2^{-|a-1|} \;\le\; x^{a-1} \;\le\; 2^{|a-1|} \qquad \text{for all } x \in \big[\tfrac{1}{2}, 1\big] \tag{68}$$

holds deterministically (because $\ln x \in [-\ln 2, 0]$ and $x^{a-1} = e^{(a-1)\ln x}$). Using (68) in (67) and integrating the $(1-x)^{b-1}$ part exactly gives

$$\frac{2^{-|a-1|}}{B(a,b)} \int_{1-y}^{1} (1-x)^{b-1}\, dx \;\le\; \mathbb{P}(R > 1-t) \;\le\; \frac{2^{|a-1|}}{B(a,b)} \int_{1-y}^{1} (1-x)^{b-1}\, dx. \tag{69}$$

Since $\int_{1-y}^{1} (1-x)^{b-1}\, dx = \frac{y^b}{b}$, we obtain

$$\frac{2^{-|a-1|}}{b\, B(a,b)}\, y^b \;\le\; \mathbb{P}(R > 1-t) \;\le\; \frac{2^{|a-1|}}{b\, B(a,b)}\, y^b. \tag{70}$$

Finally, because $t \le y \le 2t$, we have $t^b \le y^b \le (2t)^b$, and (70) yields

$$\frac{2^{-|a-1|}}{b\, B(a,b)}\, t^b \;\le\; \mathbb{P}(R > 1-t) \;\le\; \frac{2^{|a-1|+b}}{b\, B(a,b)}\, t^b, \tag{71}$$

which is exactly (62) with the explicit constants in (64). This proves $\mathbb{P}(\|\boldsymbol{Z}\| > 1-t) \asymp t^b = t^{(d-m)/2}$ as $t \downarrow 0$. □

D Proof of Theorem 4.2

Theorem D.1 (Detailed version of Theorem 4.2)**.** *Suppose $\mathcal{P}$ is a joint distribution of $(\boldsymbol{x}, y)$. Assume that the marginal distribution of $\boldsymbol{x}$ is $\mathrm{Uniform}(\mathbb{S}^{d-1})$ and the marginal distribution of y is supported on $[-D, D]$ for some $D > 0$. Fix a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where each data point is drawn i.i.d. from $\mathcal{P}$. Let $M \ge D$. If $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$ and $\|f_{\boldsymbol{\theta}}\|_\infty \le M$, then, with probability $\ge 1 - 2\delta$, we have that for the plug-in risk estimator $\widehat{\mathcal{R}}_{\mathcal{D}}(f) := \frac{1}{n}\sum_{i=1}^n \big(f(\boldsymbol{x}_i) - y_i\big)^2$,*

$$\begin{aligned} &\left| \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{P}} \Big[\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y\big)^2\Big] - \widehat{\mathcal{R}}(f_{\boldsymbol{\theta}}) \right| \\ &\lessapprox_d J M^2\, \varepsilon^{\frac{d-m}{2}} + \Big(A\, \varepsilon^{-d}\Big)^{\frac{d}{d+3}}\, M^{\frac{d+6}{d+3}}\, n^{-\frac{1}{2}}, \end{aligned} \tag{72}$$

for any $\varepsilon \in (0,1)$ such that $\varepsilon^d \gtrsim d^2 \sqrt{\frac{m + \log(J/\delta)}{n}}$[2]. Here $A = \frac{1}{\eta} - \frac{1}{2} + 4M$, and $\lessapprox_d$ hides constants depending only on d and logarithmic factors in n and (J/δ).

Proof. For $\varepsilon \in (0,1)$, we define

$$\mathsf{I}_\varepsilon^{\mathrm{all}} := \Big\{ \boldsymbol{x} \in \mathbb{S}^{d-1} :\; \max_{S_j \in \mathcal{S}} \|\boldsymbol{x}^{(S_j)}\| \le 1 - \varepsilon \Big\}, \tag{73}$$

$$\mathsf{O}_\varepsilon^{\mathrm{any}} := \Big\{ \boldsymbol{x} \in \mathbb{S}^{d-1} :\; \exists\, S_j \in \mathcal{S},\; \|\boldsymbol{x}^{(S_j)}\| > 1 - \varepsilon \Big\}. \tag{74}$$

Then $\mathbb{S}^{d-1} = \mathsf{I}_\varepsilon^{\mathrm{all}} \cup \mathsf{O}_\varepsilon^{\mathrm{any}}$ (disjoint). According to the law of total expectation, the population risk is decomposed into

$$\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{P}} \Big[\big(f(\boldsymbol{x}) - y\big)^2\Big] = \mathbb{P}(\boldsymbol{x} \in \mathsf{O}_\varepsilon^{\mathrm{any}}) \cdot \mathbb{E}_{\mathsf{O}} \Big[\big(f(\boldsymbol{x}) - y\big)^2\Big] + \mathbb{P}(\boldsymbol{x} \in \mathsf{I}_\varepsilon^{\mathrm{all}}) \cdot \mathbb{E}_{\mathsf{I}} \Big[\big(f(\boldsymbol{x}) - y\big)^2\Big], \tag{75}$$

[2]We only need $\mathrm{poly}(d)$ samples to make the feasible choice of ε non-vacuous. Here we hide the universal constant.

where $\mathbb{E}_{\mathsf{O}}$ means that $\{\boldsymbol{x}, y\}$ is a new sample from the data distribution conditioned on $\boldsymbol{x} \in \mathsf{O}_{\varepsilon}^{\text{any}}$ and $\mathbb{E}_{\mathsf{I}}$ means that $(\boldsymbol{x}, y)$ is a new sample from the data distribution conditioned on $\boldsymbol{x} \in \mathsf{I}_{\varepsilon}^{\text{all}}$.

Similarly, we also have this decomposition for empirical risk

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i)-y_i)^2 &= \frac{1}{n}\left(\sum_{i\in I}(f(\boldsymbol{x}_i)-y_i)^2+\sum_{i\in O}(f(\boldsymbol{x}_i)-y_i)^2\right)\\ &= \frac{n_I}{n}\frac{1}{n_I}\sum_{i\in I}(f(\boldsymbol{x}_i)-y_i)^2+\frac{n_O}{n}\frac{1}{n_O}\sum_{i\in O}(f(\boldsymbol{x}_i)-y_i)^2,\end{aligned} \tag{76}$$

where I is the set of data points with $\boldsymbol{x}_i \in \mathsf{I}^{\text{all}}$ and O is the set of data points with $\boldsymbol{x}_i \in \mathsf{O}_{\varepsilon}^{\text{any}}$. Then the generalization gap can be decomposed into

$$|R(f)-\hat{R}_{\mathcal{D}}(f)| \le \mathbb{P}(\boldsymbol{x}\in\mathsf{O}_{\varepsilon}^{\text{any}})\cdot\mathbb{E}_{\mathsf{O}}\left[(f_{\boldsymbol{\theta}}(\boldsymbol{x})-y)^2\right]+\frac{n_O}{n}\frac{1}{n_O}\sum_{i\in O}(f(\boldsymbol{x}_i)-y_i)^2 \tag{77}$$

$$+\left|\mathbb{P}(\boldsymbol{x}\in\mathsf{I}_{\varepsilon}^{\text{all}})-\frac{n_I}{n}\right|\frac{1}{n_I}\sum_{i\in I}(f(\boldsymbol{x}_i)-y_i)^2 \tag{78}$$

$$+\mathbb{P}(\boldsymbol{x}\in\mathsf{I}_{\varepsilon}^{\text{all}})\cdot\left|\mathbb{E}_{\mathsf{I}}\left[(f(\boldsymbol{x})-y)^2\right]-\frac{1}{n_I}\sum_{i\in I}(f(\boldsymbol{x}_i)-y_i)^2\right|. \tag{79}$$

Using the property that the marginal distribution of $\boldsymbol{x}$ is $\text{Uniform}(\mathbb{S}^{d-1})$ and its concentration property (Proposition C.11 + union bound),

$$\mathbb{P}\big(\mathsf{O}_{\varepsilon}^{\text{any}}\big)=\mathbb{P}\Big(\bigcup_{S_j\in\mathcal{S}}\{\|\boldsymbol{x}^{(S_j)}\|>1-\varepsilon\}\Big)\lesssim_d J\cdot\varepsilon^{\frac{d-m}{2}}. \tag{80}$$

The Hoeffding inequality guarantees that, with probability at least $1-\delta/3$,

$$\frac{n_O}{n}\le J\cdot\varepsilon^{\frac{d-m}{2}}+\sqrt{\frac{\log(6/\delta)}{n}} \tag{81}$$

Therefore, we may conclude that

$$(77)\lessapprox_d JM^2\varepsilon^{\frac{d-m}{2}}, \tag{82}$$

where $\lessapprox_d$ hides the constants that could depend on d and logarithmic factors of $1/\delta$.

For the term (78), with probability $1-\delta/3$

$$\begin{cases}\left|\mathbb{P}(\boldsymbol{x}\in\mathsf{I}_{\varepsilon}^{\text{all}})-\frac{n_I}{n}\right| & \lesssim\sqrt{\frac{\varepsilon^{(d-m)/2}\log(6J/\delta)}{n}},\\ \frac{1}{n_I}\sum_{i\in I}(f(\boldsymbol{x}_i)-y_i)^2 & \lesssim M^2\end{cases} \tag{83}$$

so we may also conclude that

$$(78)\lesssim M^2\sqrt{\frac{\varepsilon^{(d-m)/2}\log(6J/\delta)}{n}}\le M^2\sqrt{\frac{\log(6J/\delta)}{n}} \tag{84}$$

For the part of the interior (79), the scalar $\mathbb{P}(\boldsymbol{x}\in\mathbb{I}_{\varepsilon}^{d})$ is less than 1 with high-probability. Therefore, we just need to deal with the term

$$\mathbb{E}_{\mathbb{I}}\left[(f(\boldsymbol{x})-y)^2\right]-\frac{1}{n_I}\sum_{i\in I}(f(\boldsymbol{x}_i)-y_i)^2. \tag{85}$$

Define the projected core index set $\mathcal{C}_{\varepsilon}:=\{(\boldsymbol{u},t)\in\mathbb{S}^{m-1}\times[-1,1]:|t|\le 1-\varepsilon\}$. For any $S_j\in\mathcal{S}$, $\boldsymbol{X}\sim\text{Uniform}(\mathbb{S}^{d-1})$ and $\boldsymbol{X}^{(S_j)}=\pi_j(\boldsymbol{X})$,

- (51) shows that $\mathbb{P}(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}^{(S_j)}>t)\asymp(1-t)^{\frac{d-1}{2}}$, and
- (54) shows that $\mathbb{E}[\phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}^{(S_j)}-t)]\asymp(1-t)^{\frac{d+1}{2}}$.

Therefore we have

$$\left(\frac{1}{J}\sum_{j=1}^{J}\mathbb{P}(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}^{(S_j)}>t)\right)\left(\frac{1}{J}\sum_{j=1}^{J}\mathbb{E}[\phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}^{(S_j)}-t)]\right)\asymp(1-t)^d,\qquad t\uparrow 1. \tag{86}$$

According to (47), we have the pointwise lower bound

$$g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\ \geq\ \min\Big\{\bar{r}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\,\bar{p}_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t),\ \bar{r}_{\mathcal{P},\mathcal{S}}(-\boldsymbol{u},-t)\,\bar{p}_{\mathcal{P},\mathcal{S}}(-\boldsymbol{u},-t)\Big\},$$

since $\sqrt{\bar{p}^2+\|\bar{\boldsymbol{a}}\|_2^2}\geq\bar{p}$.

Under $\boldsymbol{X}\sim\mathrm{Uniform}(\mathbb{S}^{d-1})$, by rotational invariance, for each fixed j and unit $\boldsymbol{u}$ the quantities $\mathbb{P}(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}^{(S_j)}>t)$ and $\mathbb{E}[\phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}^{(S_j)}-t)]$ depend on t only through the scalar marginal Z in (49). Moreover both are non-increasing in t. Hence the product

$$h(t):=\mathbb{P}(Z>t)\cdot\mathbb{E}[\phi(Z-t)]$$

is non-increasing for $t\in[0,1)$, and for any $t\in[-1,1]$ we have

$$g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\ \gtrsim_d\ h(|t|).$$

Therefore, for the core $\mathcal{C}_\varepsilon=\{(\boldsymbol{u},t):|t|\leq 1-\varepsilon\}$,

$$g_{\mathcal{P},\mathcal{S},\min}(\varepsilon):=\inf_{(\boldsymbol{u},t)\in\mathcal{C}_\varepsilon}g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\ \gtrsim_d\ h(1-\varepsilon).$$

By Proposition C.10, $h(1-\varepsilon)\asymp\varepsilon^d$, hence there exists $c_g(d)>0$ such that

$$g_{\mathcal{P},\mathcal{S},\min}(\varepsilon)\ \geq\ c_g(d)\,\varepsilon^d. \tag{87}$$

On the simultaneous core $\mathsf{I}_\varepsilon^{\mathrm{all}}$, for *every* local receptive field S_j and any unit $\boldsymbol{u}\in\mathbb{R}^m$,

$$\|\boldsymbol{x}^{(S_j)}\|\leq 1-\varepsilon\ \Rightarrow\ \begin{cases}t\geq 1-\varepsilon\ \Rightarrow\ \phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x}^{(S_j)}-t)=0,\\ t\leq -1+\varepsilon\ \Rightarrow\ \phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x}^{(S_j)}-t)=\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x}^{(S_j)}-t\ \text{(affine)}.\end{cases} \tag{88}$$

Therefore all *large-offset* units ($|t|\geq 1-\varepsilon$) are affine on $\mathsf{I}_\varepsilon^{\mathrm{all}}$ *simultaneously across all views* and can be absorbed into a global affine term. The remaining *core-offset* units with $|t|\leq 1-\varepsilon$ are controlled by the $g_{\mathcal{S}}$-weighted variation. Using $|f_\theta|_{\mathrm{V}_g}\leq A$ and (87),

$$\Big|f_\theta\Big|_{\mathrm{V}(\mathsf{I}_\varepsilon^{\mathrm{all}})}\leq\ \frac{|f_\theta|_{\mathrm{V}_g}}{g_{\mathcal{P},\mathcal{S},\min}(\varepsilon)}\ \lesssim\ A\,\varepsilon^{-d}. \tag{89}$$

Therefore, we may leverage the generalization bounds for the unweighted path-norm constraint (see Lemma B.7) to deduce that with probability at least $1-\delta/3$,

$$(79)\ \lessapprox_d\ (A\,\varepsilon^{-d})^{\frac{d}{d+3}}\,M^{\frac{d+6}{d+3}}\,n^{-\frac{1}{2}}, \tag{90}$$

Combining (82), (84) and (90), we obtain

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_{\mathrm{BEoS}}(\eta,\mathcal{D})}\mathrm{GenGap}(f_{\boldsymbol{\theta}};\mathcal{D})\ \lessapprox_d\ JM^2\varepsilon^{\frac{d-m}{2}}+(A\varepsilon^{-d})^{\frac{d}{d+3}}\,M^{\frac{d+6}{d+3}}\,n^{-\frac{1}{2}}. \tag{91}$$

According to standard empirical process theory (see Theorem C.7), for any $\delta\in(0,1)$, with probability at least $1-\delta$,

$$\sup_{\substack{\boldsymbol{u}\in\mathbb{S}^{m-1}\\ t\in[-1,1]}}\big|g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t)-g_{\mathcal{P},\mathcal{S}}(\boldsymbol{u},t)\big|\leq C_{\mathrm{ep}}\sqrt{\frac{m+\log(2J/\delta)}{n}}=:\zeta_n. \tag{92}$$

Consequently, for any $\varepsilon\in(0,1)$,

$$g_{\mathcal{D},\mathcal{S},\min}(\varepsilon)\ :=\ \inf_{\substack{\boldsymbol{u}\in\mathbb{S}^{m-1}\\ |t|\leq 1-\varepsilon}}g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u},t)\ \geq\ g_{\mathcal{P},\mathcal{S},\min}(\varepsilon)-\zeta_n, \tag{93}$$

where $g_{\mathcal{P},\mathcal{S},\min}(\varepsilon)$ is the corresponding population quantity.

For $\mathcal{P}_{\boldsymbol{X}} = \mathrm{Uniform}(\mathbb{S}^{d-1})$, Proposition C.10 (see (62)) gives the explicit lower bound

$$g_{\mathcal{P},\mathcal{S},\min}(\varepsilon) \;\geq\; c_L(d)\,\varepsilon^d. \tag{94}$$

Therefore, a sufficient **validity condition** ensuring $g_{\mathcal{D},\mathcal{S},\min}(\varepsilon) \geq 2\zeta_n$ is

$$\varepsilon^d \;\geq\; \frac{2C_{\mathrm{ep}}}{c_L(d)}\sqrt{\frac{m+\log(2J/\delta)}{n}}. \tag{95}$$

The constant $C_{\mathrm{ep}} > 0$ is universal (it does not depend on d, m, n, J), and can be taken explicitly from any standard VC-/pseudo-dimension uniform convergence inequality; see, e.g., Mohri et al. [2018, Chapter 3] or Haussler [1992]. Moreover, the constant $c_L(d)$ defined in Proposition C.10 satisfies

$$c_L(d) \;\gtrsim\; \frac{1}{d^2} \qquad (d \geq 3), \tag{96}$$

which follows from standard bounds on Gamma-function ratios (e.g. Gautschi's inequality; see dlm, §5.6(i)). Hence the admissible range for ε is nonempty (e.g. $\varepsilon \in [\varepsilon_{\min}, 1)$) as soon as $n \gtrsim \mathrm{poly}(d)\,(m+\log(2J/\delta))$. □

Corollary D.2. *Under the same conditions as Theorem 4.2, assume $d > 3$ and $1 \leq m < \frac{d(d-3)}{d+3}$. Let*

$$Q \coloneqq 3d^2 + 3d - md - 3m \qquad \textit{and} \qquad \varepsilon_{\min} \coloneqq \left(c_d\, d^2\sqrt{\frac{m+\log(J/\delta)}{n}}\right)^{1/d}.$$

Define the optimal choice

$$\varepsilon^\star \;\asymp\; A^{\frac{2d}{Q}}\, J^{-\frac{2(d+3)}{Q}}\, M^{-\frac{2d}{Q}}\, n^{-\frac{d+3}{Q}}, \tag{97}$$

and choose the feasible/truncated value

$$\varepsilon^\dagger \coloneqq \max\{\varepsilon_{\min},\, \varepsilon^\star\}.$$

Then, with probability at least $1-\delta$,

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta,\mathcal{D})} \left|\mathbb{E}\big[(f_{\boldsymbol{\theta}}(\boldsymbol{x})-y)^2\big] - \widehat{\mathcal{R}}_{\mathcal{D}}(f_{\boldsymbol{\theta}})\right| \;\lesssim_d\; JM^2(\varepsilon^\dagger)^{\frac{d-m}{2}} + \Big(A(\varepsilon^\dagger)^{-d}\Big)^{\frac{d}{d+3}} M^{\frac{d+6}{d+3}} n^{-1/2}. \tag{98}$$

Moreover, in the regime where $\varepsilon^\star \geq \varepsilon_{\min}$ (e.g., for n sufficiently large, treating A, J, M as constants), plugging $\varepsilon = \varepsilon^\star$ into (7) yields the optimized rate

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta,\mathcal{D})} \left|\mathbb{E}\big[(f_{\boldsymbol{\theta}}(\boldsymbol{x})-y)^2\big] - \widehat{\mathcal{R}}_{\mathcal{D}}(f_{\boldsymbol{\theta}})\right| \;\lesssim_d\; A^{\alpha_A}\, J^{\alpha_J}\, M^{\alpha_M}\, n^{-\alpha_n}, \tag{99}$$

where

$$\begin{aligned}
\alpha_A &= \frac{d(d-m)}{Q},\\
\alpha_J &= \frac{2d^2}{Q},\\
\alpha_M &= \frac{4d^2}{Q} + \frac{(d+6)(d-m)(d+3)}{(d+3)Q},\\
\alpha_n &= \frac{(d-m)(d+3)}{2Q}.
\end{aligned}$$

In particular, if m is fixed and $d \to \infty$, then

$$\alpha_n \to \frac{1}{6}, \qquad \alpha_A \to \frac{1}{3}, \qquad \alpha_J \to \frac{2}{3}, \qquad \alpha_M \to \frac{5}{3}.$$

Proof. The optimal choice of ε minimizing the RHS of (91) is

$$\varepsilon^* \asymp A^{\frac{2d}{Q}} J^{-\frac{2(d+3)}{Q}} M^{-\frac{2d}{Q}} n^{-\frac{d+3}{Q}}, \tag{100}$$

where

$$Q = (d-m)(d+3) + 2d^2 = 3d^2 + 3d - md - 3m.$$

Plugging ε^* into (91) yields

$$\begin{aligned} \sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}_{\mathrm{BEoS}}(\eta, \mathcal{D})} \mathrm{GenGap}(f_{\boldsymbol{\theta}}; \mathcal{D}) &\lessapprox_d A^{\alpha_A} J^{\alpha_J} M^{\alpha_M} n^{-\alpha_n} \\ &\lessapprox_{A,M,J,d} O\Big(n^{-\frac{(d-m)(d+3)}{2\left((d-m)(d+3)+2d^2\right)}}\Big). \end{aligned} \tag{101}$$

In particular, when m is fixed and $d \to \infty$,

$$\frac{(d-m)(d+3)}{2\big((d-m)(d+3)+2d^2\big)} \longrightarrow \frac{1}{6}.$$

Therefore, this rate does not suffer from the curse of dimensionality when m is fixed.

Finally, we verify the validity of the plug-in choice ε^*. The validity condition (95) requires $\varepsilon^d \gtrsim_d n^{-1/2}$ (up to logarithmic factors). Since $(\varepsilon^*)^d \asymp n^{-d(d+3)/Q}$ (treating A, J, M as constants), plugging (100) into (95) gives

$$n^{-\frac{d(d+3)}{Q}} \gtrsim_d n^{-1/2}.$$

This requires

$$\frac{1}{2} > \frac{d(d+3)}{Q} = \frac{d(d+3)}{(d-m)(d+3)+2d^2},$$

which is equivalent to

$$(d-m)(d+3) + 2d^2 > 2d(d+3) \iff d^2 - 3d > m(d+3) \iff m < \frac{d(d-3)}{d+3}.$$

□

E Proof of Theorem 4.3

We use the model (1) and the loss $\mathcal{L} = \widehat{\mathcal{R}}_{\mathcal{D}}/2$ from Section 3, with $0 < \eta < 2$. Population expectations are evaluated on a fresh sample independent of $\mathcal{D}$.

The proof places one filter in each of nJ disjoint spherical caps containing no training patches. Each filter has height one and is active on a region of probability proportional to $1/(nJ)$. Together, these filters retain a constant population effect while their training activations vanish.

Fix $m \geq 2$ and $J \geq 1$, let $d = mJ$, and take the disjoint receptive fields

$$S_j = \{(j-1)m + 1, \ldots, jm\}, \qquad j \in [J].$$

Consider the input distribution

$$\boldsymbol{X} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_J), \qquad \boldsymbol{Z}_j \overset{\text{i.i.d.}}{\sim} \mathrm{Uniform}(\mathbb{S}^{m-1}), \qquad \mathcal{X} = (\mathbb{S}^{m-1})^J. \tag{102}$$

Each patch lies on the unit sphere, and $\|\boldsymbol{X}\|_2 = \sqrt{J}$. This is the spherical-patch model used in Figure 3.

Proof. Write $N = nJ$ for the number of training patches.

For a uniform patch $\boldsymbol{Z} \in \mathbb{S}^{m-1}$ and any center $\boldsymbol{u} \in \mathbb{S}^{m-1}$, the cap of Euclidean radius $r \leq 1$ has probability

$$\mathbb{P}(\|\boldsymbol{Z} - \boldsymbol{u}\|_2 < r) \asymp_m r^{m-1}. \tag{103}$$

To see this, its angular radius is $\alpha = 2\arcsin(r/2) \asymp r$, and its area is proportional to $\int_0^\alpha \sin^{m-2}(t)\,\mathrm{d}t \asymp_m r^{m-1}$. Choose $\varepsilon = aN^{-1/(m-1)}$, with $a > 0$ depending only on m and

small enough that $4\varepsilon \le 1$ and every cap of radius 4ε has probability at most $1/(2N)$. A maximal collection of centers separated by at least 4ε covers the sphere with caps of radius 4ε: any uncovered point could be added to the collection. Since the covering caps each have probability at most $1/(2N)$, there are at least $2N$ centers. The caps of radius 2ε around these centers are disjoint, so the N training patches occupy at most N of them. Select N empty caps and label their centers $\boldsymbol{u}_1, \dots, \boldsymbol{u}_N$.

We choose each filter to have height one at its center and an activation region strictly inside the empty cap. For unit vectors, $\|\boldsymbol{z} - \boldsymbol{u}_k\|_2^2 = 2(1 - \boldsymbol{u}_k^\mathsf{T}\boldsymbol{z})$, so the threshold $1 - \varepsilon^2$ gives an activation radius of $\sqrt{2}\varepsilon < 2\varepsilon$. Dividing by ε^2 sets the height at the center to one. Thus we define

$$\psi_k(\boldsymbol{z}) = \varepsilon^{-2}\phi\big(\boldsymbol{u}_k^\mathsf{T}\boldsymbol{z} - (1-\varepsilon^2)\big), \qquad h(\boldsymbol{x}) = \frac{M}{J}\sum_{j=1}^{J}\sum_{k=1}^{N}\psi_k(\pi_j(\boldsymbol{x})). \tag{104}$$

On the unit sphere, $0 \le \psi_k \le 1$. The filter supports lie in disjoint empty caps, giving $0 \le \sum_k \psi_k \le 1$ and hence $0 \le h \le M$ on $\mathcal{X}$. This is the model (1) with N filters, $\boldsymbol{w}_k = \boldsymbol{u}_k$, $b_k = 1 - \varepsilon^2$, $v_k = M\varepsilon^{-2}$, and $\beta = 0$.

Every training patch is at distance at least 2ε from each selected center. Therefore,

$$\boldsymbol{u}_k^\mathsf{T}\pi_j(\boldsymbol{x}_i) - (1-\varepsilon^2) \le -\varepsilon^2 \qquad \text{for all } i, j, k. \tag{105}$$

All filters remain inactive on the training sample in a neighborhood of this parameter. In that neighborhood the network's training predictions equal its output bias β. Since all labels are zero, the loss is locally $\beta^2/2$. Thus the empirical risk is zero and the Hessian has one eigenvalue equal to 1, with all others zero. This proves membership in $\boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$ and (8).

Inside the smaller cap $\|\boldsymbol{z} - \boldsymbol{u}_k\|_2 < \varepsilon$, the filter satisfies $\psi_k(\boldsymbol{z}) \ge 1/2$. Equation (103) therefore gives

$$\mathbb{E}\left[\sum_{k=1}^{N}\psi_k(\boldsymbol{Z})\right] \ge \frac{1}{2}\sum_{k=1}^{N}\mathbb{P}(\|\boldsymbol{Z} - \boldsymbol{u}_k\|_2 < \varepsilon) \gtrsim_m N\varepsilon^{m-1} = a^{m-1}.$$

Each patch has the same expected response, so averaging over the J patches preserves this lower bound. Multiplying by M gives $\mathbb{E}[h(\boldsymbol{X})] \gtrsim_m M$. Jensen's inequality yields

$$\mathcal{R}(h) = \mathbb{E}[h(\boldsymbol{X})^2] \ge \big(\mathbb{E}[h(\boldsymbol{X})]\big)^2 \ge c_m M^2$$

for a constant $c_m > 0$ depending only on m. The empirical risk is zero, so this is also the generalization gap, proving (9). □

F Extending the Framework beyond Weight Sharing and Global Average Pooling

The main text analyzes the minimal convolutional model (1) with weight sharing and global average pooling (GAP). These two design choices simplify the exposition but are not prerequisites for the stability-induced regularization mechanism. To demonstrate that *sparse connectivity alone* is the essential ingredient, this appendix analyzes a variant that retains the sparse receptive-field pattern of the main-text SCN but removes both weight sharing and global pooling.

F.1 Sparsely connected network without weight sharing

Setup and notation. Fix the collection of local receptive fields $\mathcal{S} = \{S_j\}_{j=1}^{J}$ (each $S_j \subset [d]$ with $|S_j| = m$) and the corresponding coordinate projections $\pi_j : \mathbb{R}^d \to \mathbb{R}^m$.

Definition F.1 (SCN without weight sharing)**.** A *sparsely connected network without weight sharing*, with receptive fields $\mathcal{S}$ and width K, is a function of the form

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) \;=\; \sum_{k=1}^{K}\sum_{j=1}^{J} v_{k,j}\,\phi\big(\boldsymbol{w}_{k,j}^\mathsf{T}\pi_j(\boldsymbol{x}) - b_{k,j}\big) \;+\; \beta, \tag{106}$$

where $\boldsymbol{w}_{k,j} \in \mathbb{R}^m$, $b_{k,j} \in \mathbb{R}$, $v_{k,j} \in \mathbb{R}$ are *independent* parameters for each neuron index $k \in [K]$ and location index $j \in [J]$, and $\beta \in \mathbb{R}$ is a scalar bias. We write $\boldsymbol{\theta} = \{(\boldsymbol{w}_{k,j}, b_{k,j}, v_{k,j})\}_{k\in[K],j\in[J]} \cup \{\beta\}$ for the full parameter vector.

The key structural property shared with the main-text SCN (1) is *sparse connectivity*: neuron (k, j) receives input only from the m-dimensional patch $\pi_j(\boldsymbol{x})$. The architecture (106) differs from (1) in two respects: (i) filter weights $\boldsymbol{w}_{k,j}$ and biases $b_{k,j}$ are *not* shared across locations j, and (ii) output weights $v_{k,j}$ may vary freely with j, so there is no global average pooling.

Data and loss. As in the main text, the dataset is $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \in B_R^d$ and $y_i \in [-D, D]$, and the training objective is $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2n}\sum_{i=1}^n (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2$.

Location-specific weight function. Because neurons at different locations j now operate with independent parameters on different patch distributions, the relevant weight function is defined *per location*. For each $j \in [J]$, let $\boldsymbol{X}_{\mathcal{D}}^{(j)}$ be a random vector drawn uniformly from the training patches at position j, i.e. from $\{\pi_j(\boldsymbol{x}_i)\}_{i=1}^n \subset \mathbb{R}^m$. Define

$$g_{\mathcal{D},\mathcal{S}}^{(j)}(\boldsymbol{u}, t) \;:=\; \mathbb{E}\Big[\phi\big(\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{\mathcal{D}}^{(j)} - t\big)\Big]\,\sqrt{\mathbb{P}\big(\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{\mathcal{D}}^{(j)} > t\big)^2 + \Big\|\mathbb{E}\Big[\boldsymbol{X}_{\mathcal{D}}^{(j)}\,\mathbb{1}\Big\{\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{\mathcal{D}}^{(j)} > t\Big\}\Big]\Big\|_2^2}. \tag{107}$$

This is the direct analogue of the global weight function $g_{\mathcal{D},\mathcal{S}}$ in Definition C.1, specialized to a single location.

F.2 From the BEoS condition to a location-wise weighted path norm

We establish the counterpart of Theorem 4.1 for the unshared architecture (106). The proof follows the same strategy—lower-bounding $\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}})$ via the all-ones direction—but now each neuron (k, j) contributes independently, which allows the bound to decompose over locations.

Lemma F.2 (Hessian residual bound for the unshared SCN)**.** *For the architecture* (106) *and any* $\boldsymbol{x}$ *with* $\|\boldsymbol{x}\|_2 \le R$*,*

$$\big\|\nabla_{\boldsymbol{\theta}}^2 f_{\boldsymbol{\theta}}(\boldsymbol{x})\big\|_{\mathrm{op}} \;\le\; 2(R+1).$$

Proof. The proof of Lemma C.3 bounds $|\boldsymbol{\omega}^\mathsf{T}\nabla_{\boldsymbol{\theta}}^2 f(\boldsymbol{x})\,\boldsymbol{\omega}|$ for a unit perturbation $\boldsymbol{\omega}$ by examining each neuron independently: the only nonzero second derivatives within neuron (k, j) are the mixed partials between $(\boldsymbol{w}_{k,j}, b_{k,j})$ and $v_{k,j}$, exactly as in (33)–(34). Since different neurons occupy orthogonal parameter blocks, their contributions to $\boldsymbol{\omega}^\mathsf{T}\nabla_{\boldsymbol{\theta}}^2 f(\boldsymbol{x})\,\boldsymbol{\omega}$ sum without interaction. Each block contributes at most $2|\gamma_{k,j}|(R\,\|\boldsymbol{\alpha}_{k,j}\|_2 + |\delta_{k,j}|)$ (cf. (39)), where $(\boldsymbol{\alpha}_{k,j}, \delta_{k,j}, \gamma_{k,j})$ is the perturbation restricted to neuron (k, j). Summing over (k, j) and applying Cauchy–Schwarz with the normalization $\|\boldsymbol{\omega}\|_2 = 1$ yields the same bound $2(R+1)$. □

Proposition F.3 (Tangent-feature lower bound for the unshared SCN)**.** *Let* $\boldsymbol{T}_{\mathcal{D}} = \frac{1}{n}\sum_{i=1}^n \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\,\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)^\mathsf{T}$*. Then*

$$\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}}) \;\ge\; 1 \;+\; 2\sum_{k=1}^K\sum_{j=1}^J |v_{k,j}|\;\|\boldsymbol{w}_{k,j}\|\;g_{\mathcal{D},\mathcal{S}}^{(j)}\!\left(\frac{\boldsymbol{w}_{k,j}}{\|\boldsymbol{w}_{k,j}\|},\,\frac{b_{k,j}}{\|\boldsymbol{w}_{k,j}\|}\right). \tag{108}$$

Proof. Write the gate indicators $m_{k,j,i} := \mathbb{1}\left\{\boldsymbol{w}_{k,j}^\mathsf{T}\pi_j(\boldsymbol{x}_i) > b_{k,j}\right\}$, the normalized direction $\boldsymbol{u}_{k,j} := \boldsymbol{w}_{k,j}/\|\boldsymbol{w}_{k,j}\|_2$, and the normalized threshold $t_{k,j} := b_{k,j}/\|\boldsymbol{w}_{k,j}\|_2$. The partial derivatives of $f_{\boldsymbol{\theta}}$ at sample $\boldsymbol{x}_i$ are

$$\frac{\partial f}{\partial \boldsymbol{w}_{k,j}}(\boldsymbol{x}_i) = v_{k,j}\,m_{k,j,i}\,\pi_j(\boldsymbol{x}_i), \qquad \frac{\partial f}{\partial b_{k,j}}(\boldsymbol{x}_i) = -v_{k,j}\,m_{k,j,i}, \tag{109}$$

$$\frac{\partial f}{\partial v_{k,j}}(\boldsymbol{x}_i) = \phi\big(\boldsymbol{w}_{k,j}^\mathsf{T}\pi_j(\boldsymbol{x}_i) - b_{k,j}\big), \qquad \frac{\partial f}{\partial \beta}(\boldsymbol{x}_i) = 1. \tag{110}$$

Since each neuron (k, j) operates on the patch $\pi_j(\boldsymbol{x}_i)$ with *independent* parameters $(\boldsymbol{w}_{k,j}, b_{k,j}, v_{k,j})$, the gradient blocks are orthogonal across different (k, j) pairs.

Denoting the tangent-feature matrix $\Phi = [\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_1), \ldots, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_n)] \in \mathbb{R}^{p\times n}$, we use the standard lower bound (cf. (31))

$$\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}}) = \frac{1}{n}\max_{\|\boldsymbol{u}\|=1}\|\Phi\boldsymbol{u}\|^2 \;\ge\; \frac{1}{n^2}\,\|\Phi\,\mathbf{1}\|^2 = \frac{1}{n^2}\Big\|\sum_{i=1}^n \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)\Big\|^2. \tag{111}$$

Expanding the squared norm and using the orthogonality of neuron blocks gives

$$\frac{1}{n^2}\,\|\Phi\,\mathbf{1}\|^2 = 1 + \sum_{k=1}^{K}\sum_{j=1}^{J}\Bigg[\underbrace{v_{k,j}^2\Big(\Big\|\frac{1}{n}\sum_i m_{k,j,i}\,\pi_j(\boldsymbol{x}_i)\Big\|^2 + \Big(\frac{1}{n}\sum_i m_{k,j,i}\Big)^2\Big)}_{=:\;P_{k,j}} + \underbrace{\|\boldsymbol{w}_{k,j}\|_2^2\,\Big(\frac{1}{n}\sum_i \phi\big(\boldsymbol{u}_{k,j}^\mathsf{T}\pi_j(\boldsymbol{x}_i) - t_{k,j}\big)\Big)^2}_{=:\;Q_{k,j}}\Bigg]. \tag{112}$$

Here the "1" comes from $\big(\frac{1}{n}\sum_i \partial f/\partial\beta\big)^2 = 1$. The term $P_{k,j}$ collects the squared sums of the $\boldsymbol{w}_{k,j}$- and $b_{k,j}$-gradients (both proportional to $v_{k,j}$), while $Q_{k,j}$ comes from the $v_{k,j}$-gradient (proportional to $\|\boldsymbol{w}_{k,j}\|_2$). Crucially, the double sum decomposes over (k,j) because the parameters are not shared across locations.

We apply $a^2+b^2 \geq 2ab$ to each (k,j) independently (cf. the analogous step in the proof of Proposition C.2):

$$P_{k,j} + Q_{k,j} \;\geq\; 2\,|v_{k,j}|\;\|\boldsymbol{w}_{k,j}\|_2 \times \underbrace{\Big(\frac{1}{n}\sum_i \phi\big(\boldsymbol{u}_{k,j}^\mathsf{T}\pi_j(\boldsymbol{x}_i) - t_{k,j}\big)\Big)\sqrt{\Big(\frac{1}{n}\sum_i m_{k,j,i}\Big)^2 + \Big\|\frac{1}{n}\sum_i m_{k,j,i}\,\pi_j(\boldsymbol{x}_i)\Big\|^2}}_{=\;g^{(j)}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}_{k,j},\,t_{k,j})}. \tag{113}$$

To verify the identification with (107), note that the empirical averages over $i \in [n]$ are precisely the expectation, probability, and conditional first moment under the uniform draw $\boldsymbol{X}_{\mathcal{D}}^{(j)} \sim \mathrm{Uniform}(\{\pi_j(\boldsymbol{x}_i)\}_{i=1}^n)$:

$$\frac{1}{n}\sum_{i=1}^n \phi\big(\boldsymbol{u}^\mathsf{T}\pi_j(\boldsymbol{x}_i) - t\big) = \mathbb{E}\big[\phi(\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{\mathcal{D}}^{(j)} - t)\big], \qquad \frac{1}{n}\sum_{i=1}^n m_{k,j,i} = \mathbb{P}\big(\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{\mathcal{D}}^{(j)} > t\big),$$

$$\frac{1}{n}\sum_{i=1}^n m_{k,j,i}\,\pi_j(\boldsymbol{x}_i) = \mathbb{E}\big[\boldsymbol{X}_{\mathcal{D}}^{(j)}\,\mathbb{1}\left\{\boldsymbol{u}^\mathsf{T}\boldsymbol{X}_{\mathcal{D}}^{(j)} > t\right\}\big].$$

Summing (113) over (k,j) and combining with (111)–(112) yields (108). □

Remark F.4 (Why the per-neuron decomposition succeeds here)**.** In the GAP model (1), the output weight v_k is shared uniformly across all locations via the $1/J$ average. This forces $\nabla_{\boldsymbol{w}_k} f = \frac{v_k}{J}\sum_j m_{k,i}^{(S_j)}\,\boldsymbol{x}_i^{(S_j)}$, in which contributions from different locations are combined with a *common* coefficient v_k/J. The common coefficient allows v_k^2 to be factored out, and the resulting $\frac{1}{n^2}\,\|\Phi\mathbf{1}\|^2$ factors through the *global* patch distribution $\boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}}$, producing the global weight function $g_{\mathcal{D},\mathcal{S}}$ of Proposition C.2.

In the unshared architecture (106), each neuron (k,j) has its own output weight $v_{k,j}$, but since $\boldsymbol{w}_{k,j}$ is also independent across j, the gradient blocks are orthogonal and the squared norm decomposes cleanly into per-(k,j) terms (112). This per-neuron decomposition makes the AM–GM step (113) straightforward.

An intermediate architecture—shared filters $\boldsymbol{w}_k$ with location-dependent output weights $v_{k,j}$—would couple locations through $\nabla_{\boldsymbol{w}_k} f = \sum_j v_{k,j}\,m_{k,j,i}\,\pi_j(\boldsymbol{x}_i)$, in which contributions from different locations can cancel when $v_{k,j}$ have varying signs. This coupling prevents both the per-location decomposition used above and the common-coefficient factorization available under GAP. Analyzing such architectures would require different proof techniques.

Theorem F.5 (BEoS constraint for the unshared SCN)**.** *For the architecture* (106)*, if* $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}}_{\mathrm{BEoS}}(\eta,\mathcal{D})$*, i.e.,* $\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) \leq 2/\eta$*, then*

$$\sum_{k=1}^K\sum_{j=1}^J |v_{k,j}|\;\|\boldsymbol{w}_{k,j}\|\;g^{(j)}_{\mathcal{D},\mathcal{S}}\Big(\frac{\boldsymbol{w}_{k,j}}{\|\boldsymbol{w}_{k,j}\|},\,\frac{b_{k,j}}{\|\boldsymbol{w}_{k,j}\|}\Big) \;\leq\; \frac{1}{\eta} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{114}$$

Proof. Decompose the Hessian as $\nabla^2_{\boldsymbol{\theta}}\mathcal{L} = \boldsymbol{T}_{\mathcal{D}} + \boldsymbol{R}_{\mathcal{D}}$ (cf. (28)) with

$$\boldsymbol{R}_{\mathcal{D}} = \frac{1}{n}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)\,\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i).$$

Let $\boldsymbol{\omega}$ be the unit eigenvector of $\boldsymbol{T}_{\mathcal{D}}$ corresponding to $\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}})$. Then (cf. (42))

$$\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) \;\geq\; \boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}}\mathcal{L}\,\boldsymbol{\omega} \;=\; \lambda_{\max}(\boldsymbol{T}_{\mathcal{D}}) + \underbrace{\frac{1}{n}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)\,\boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\,\boldsymbol{\omega}}_{=:\;\Delta}.$$

By the Cauchy–Schwarz inequality and Lemma F.2,

$$|\Delta| \;\leq\; \sqrt{\frac{1}{n}\sum_{i}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2}\;\cdot\;\sqrt{\frac{1}{n}\sum_{i}\big(\boldsymbol{\omega}^{\mathsf{T}}\nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\,\boldsymbol{\omega}\big)^2} \;\leq\; 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}.$$

Therefore,

$$\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}}) \;\leq\; \lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) + |\Delta| \;\leq\; \frac{2}{\eta} + 2(R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}.$$

Inserting the lower bound from Proposition F.3 and rearranging yields (114). □

F.3 Generalization analysis

We now derive the generalization guarantee for the unshared architecture (106), following the same interior/exterior decomposition as in Appendix D.

Interior/exterior decomposition. For $\varepsilon \in (0,1)$, define

$$\mathsf{I}^{\mathrm{all}}_{\varepsilon} := \Big\{\boldsymbol{x} \in \mathbb{S}^{d-1} : \max_{j\in[J]} \|\pi_j(\boldsymbol{x})\| \leq 1-\varepsilon\Big\},$$
$$\mathsf{O}^{\mathrm{any}}_{\varepsilon} := \Big\{\boldsymbol{x} \in \mathbb{S}^{d-1} : \exists\, j \in [J],\ \|\pi_j(\boldsymbol{x})\| > 1-\varepsilon\Big\}.$$

By a union bound and Proposition C.11, $\mathbb{P}(\mathsf{O}^{\mathrm{any}}_{\varepsilon}) \lesssim J\,\varepsilon^{(d-m)/2}$.

Lower bound on $g^{(j)}_{\mathcal{D},\mathcal{S}}$ over the interior. On $\mathsf{I}^{\mathrm{all}}_{\varepsilon}$, every patch satisfies $\|\pi_j(\boldsymbol{x})\| \leq 1-\varepsilon$ for all j. When the marginal of $\boldsymbol{x}$ is $\mathrm{Uniform}(\mathbb{S}^{d-1})$, each projected patch $\pi_j(\boldsymbol{x}) \in \mathbb{R}^m$ inherits the same rotational-invariance structure analyzed in Appendix C. By the same computation as in Proposition C.10 (see (59)), the population weight function at each location satisfies

$$g^{(j)}_{\mathcal{P}}(\boldsymbol{u}, t) \;\geq\; c_g(d)\,\varepsilon^d \qquad \text{for all } \boldsymbol{u} \in \mathbb{S}^{m-1},\ |t| \leq 1-\varepsilon,$$

with a constant $c_g(d) > 0$ depending only on d. The uniform deviation bound of Theorem C.7, applied separately at each location j (the patches $\{\pi_j(\boldsymbol{x}_i)\}_{i=1}^n$ are i.i.d. for fixed j), gives

$$\sup_{\boldsymbol{u}\in\mathbb{S}^{m-1},\,|t|\leq 1} \big|g^{(j)}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) - g^{(j)}_{\mathcal{P}}(\boldsymbol{u}, t)\big| \;\leq\; C_{\mathrm{ep}}\sqrt{\frac{m + \log(2J/\delta)}{n}} \;=:\; \zeta_n,$$

with probability at least $1-\delta$, simultaneously for all $j \in [J]$ (via a union bound absorbing J into the logarithm). Hence, for any ε satisfying $\varepsilon^d \geq 2\zeta_n / c_g(d)$, we have

$$g^{(j)}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) \;\geq\; \tfrac{1}{2}\,c_g(d)\,\varepsilon^d \qquad \text{for all } j \in [J],\ \boldsymbol{u} \in \mathbb{S}^{m-1},\ |t| \leq 1-\varepsilon. \tag{115}$$

From the BEoS constraint to an unweighted path norm. On the interior $\mathsf{I}^{\mathrm{all}}_{\varepsilon}$, every neuron with $|t_{k,j}| \geq 1-\varepsilon$ is either identically zero or fully affine on all training patches (cf. (88)), and can be absorbed into the affine component. For the remaining neurons with $|t_{k,j}| \leq 1-\varepsilon$, inserting the lower bound (115) into the BEoS constraint (114) and setting $A := \frac{1}{\eta} - \frac{1}{2} + 4M$ yields

$$\sum_{\substack{k,j:\\ |t_{k,j}|\leq 1-\varepsilon}} |v_{k,j}|\;\|\boldsymbol{w}_{k,j}\| \;\leq\; \frac{A}{\frac{1}{2}c_g(d)\,\varepsilon^d} \;\asymp\; A\,\varepsilon^{-d}. \tag{116}$$

Controlling the generalization gap. The function class on the interior set $\mathsf{I}^{\mathrm{all}}_{\varepsilon}$ has variation norm bounded by $A\varepsilon^{-d}$ (equation (116)). The generalization gap decomposes as in (77)–(79):

- *Exterior contribution.* Since $\|f_{\boldsymbol{\theta}}\|_\infty \le M$ and $|y| \le D \le M$, the exterior contributes at most $O(JM^2\varepsilon^{(d-m)/2})$ (cf. (82)).
- *Interior contribution.* By Lemma B.7 (metric entropy of variation spaces), the interior generalization gap satisfies

$$\left|\mathbb{E}_{\mathsf{I}}\big[(f_{\boldsymbol{\theta}}-y)^2\big] - \frac{1}{n_I}\sum_{i\in I}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)-y_i)^2\right| \lessapprox_d \ (A\varepsilon^{-d})^{\frac{d}{d+3}}\, M^{\frac{d+6}{d+3}}\, n^{-1/2}$$

  (cf. (90)).
- *Concentration of the interior fraction.* By Hoeffding's inequality, $|\mathbb{P}(\mathsf{I}_\varepsilon^{\mathrm{all}}) - n_I/n| \lesssim \sqrt{\log(1/\delta)/n}$ with high probability, contributing a lower-order term $O(M^2\sqrt{\log(1/\delta)/n})$ (cf. (84)).

Combining these three terms gives the following result.

Theorem F.6 (Generalization for the unshared SCN)**.** *Assume $\boldsymbol{x} \sim \mathrm{Uniform}(\mathbb{S}^{d-1})$ and $|y| \le D$. Let $M \ge D$, $\|f_{\boldsymbol{\theta}}\|_\infty \le M$, and suppose $\boldsymbol{\theta}$ satisfies the BEoS condition $\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) \le 2/\eta$. Assume $d > 3$ and $1 \le m < \frac{d(d-3)}{d+3}$. Then, with probability at least $1-2\delta$,*

$$\mathrm{GenGap}(f_{\boldsymbol{\theta}};\mathcal{D}) \ \lessapprox_d \ JM^2\,\varepsilon^{\frac{d-m}{2}} + \big(A\,\varepsilon^{-d}\big)^{\frac{d}{d+3}} M^{\frac{d+6}{d+3}} n^{-1/2}, \tag{117}$$

for any $\varepsilon \in (0,1)$ satisfying $\varepsilon^d \gtrsim d^2\sqrt{(m+\log(J/\delta))/n}$, where $A = \frac{1}{\eta} - \frac{1}{2} + 4M$.

Proof. The proof follows the decomposition and estimates described above. The exterior bound uses Proposition C.11 and Hoeffding's inequality (see (82)). The interior bound uses the path-norm estimate (116) together with the metric entropy of variation spaces (Lemma B.7; see (90)). The concentration term $\big|\mathbb{P}(\mathsf{I}_\varepsilon^{\mathrm{all}}) - n_I/n\big|$ is controlled as in (84) and is of smaller order. □

Optimizing over ε as in Corollary D.2 yields the rate exponent

$$-\frac{(d-m)(d+3)}{2\big((d-m)(d+3)+2d^2\big)},$$

which tends to $-\frac{1}{6}$ for fixed m and $d \to \infty$, identical to the rate obtained for the weight-shared SCN in Theorem 4.2.

Conclusion. The only architectural prerequisite for the stability-induced implicit regularization is the *sparse connectivity pattern*: each hidden neuron receives input from a small subset of the ambient coordinates. Weight sharing and GAP change how the patch geometry is aggregated. In the GAP model, the shared filter and common output coefficient couple all locations and yield a single global patch-multiset weight $g_{\mathcal{D},\mathcal{S}}$. In the unshared model, the same mechanism appears as a sum of location-wise weights $g^{(j)}_{\mathcal{D},\mathcal{S}}$. Thus weight sharing and GAP are not necessary for stability-induced regularization. They determine whether the patch-geometry penalty is global and pooled or location-wise. This conclusion substantiates the design principle articulated in Section 6: the effective vector dimension seen by each local operator governs the strength of implicit regularization, regardless of whether parameters are shared across locations.

G Experimental Details

We adopt the random-design nonparametric regression setting.

The setting of nonparametric regression. The non-parametric regression with noisy labels on the random design is one typical setting in this program. Suppose $\{\boldsymbol{x}_i\}_{i=1}^n$ are i.i.d. sampled from a distribution $\mathcal{P}_{\boldsymbol{X}}$ supported on $\mathbb{B}_R^d$ and $y_i = f_{true}(\boldsymbol{x}_i) + \xi_i$ for $i \in [n]$, where $f_{true}\colon \mathbb{R}^d \to \mathbb{R}$ is the ground-true function and $\{\xi_i\}_{i=1}^n$ are i.i.d. Gaussian noises $\mathcal{N}(0,\sigma^2)$. In this setting, the goal of the regression task is to find a predictor f to minimize the mean squared error (MSE):

$$\mathrm{MSE}(f) = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_{true}(\boldsymbol{x}_i))^2. \tag{118}$$

The population level of (118) is known as *excess risk*,

$$\mathrm{Excess}(f) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X}}}[(f(\boldsymbol{x}) - f_{true}(\boldsymbol{x}))^2], \tag{119}$$

which is also called the *estimation error* under $L^2(\mathcal{P}_{\boldsymbol{X}})$.

In this setting, the population risk (under squared loss) of a predictor f decomposes as

$$\mathcal{R}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}}\big[(f(\boldsymbol{x}) - y)^2\big] = \mathrm{Excess}(f) + \sigma^2, \tag{120}$$

The additive term σ^2 is the irreducible error contributed by the label noise, and it is achieved by the Bayes predictor $f^*(\boldsymbol{x}) = \mathbb{E}[y \mid \boldsymbol{x}] = f_{true}(\boldsymbol{x})$, namely $R(f^*) = \sigma^2$. Consequently, $\mathcal{R}(f) - \mathcal{R}(f^*) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X}}}[(f(\boldsymbol{x}) - f_{true}(\boldsymbol{x}))^2]$, so controlling the excess risk is equivalent to controlling the population regret in squared-loss regression.

Moreover, the generalization gap in this random-design regression model admits the following equivalent form

$$\mathrm{GenGap}(f, \mathcal{D}) = \Big|\mathrm{Excess}(f) + \sigma^2 - \widehat{\mathcal{R}}_{\mathcal{D}}(f)\Big|. \tag{121}$$

This identity makes explicit how the generalization gap compares the *population* performance (excess risk plus irreducible noise) against the *training* performance measured by the empirical squared loss.

G.1 Experimental details for Figure 2a in Section 4.2

We use synthetic data to empirically validate our claim that SCN generalizes well on spherical data when $m \ll d$, and the rate does not deteriorate as the ambient dimension increases.

Data generation: Fix a collection of receptive fields $\mathcal{S} = \{S_j\}_{j=1}^J$ with patch size m. In all experiments we use disjoint coordinate patches $S_j = \{(j-1)m + 1, \ldots, jm\}$ so that $J = \lfloor d/m \rfloor$. We first sample a ground-truth predictor $f_{true} \in \boldsymbol{\Theta}^{\mathcal{S}}$ from the architecture in (1) with a moderate width $K_{true} = 20$ (fixed across all n), and generate training inputs $\{\boldsymbol{x}_i\}_{i=1}^n$ i.i.d. from $\mathrm{Uniform}(\mathbb{S}^{d-1})$. The labels are generated by $y_i = f_{true}(\boldsymbol{x}_i) + \xi_i$, where $\xi_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d Gaussian noise. We also generate an independent test set $\{\tilde{\boldsymbol{x}}_r\}_{r=1}^N \sim \mathrm{Uniform}(\mathbb{S}^{d-1})$ (with $N \gg n$) for Monte Carlo evaluation of generalization gap.

Architecture: We compare two overparameterized two-layer ReLU models trained by full-batch GD: (i) **SCN** (two-layer sparsely connected ReLU network in (1)), and (ii) **FCN** obtained by taking $m = d$ and $J = 1$ in (1). Unless otherwise stated, both models use width $K = 1024$, which places the training in an overparameterized regime for the sample sizes we consider. Both use Kaiming-normal initialization for hidden weights and zero initialization for all biases.

Metrics: For each trained predictor f, we report (i) **train loss** $\widehat{\mathcal{R}}_{\mathcal{D}}(f)$ This quantity measures how well the predictor fit the training set; (ii) **estimated generalization gap**, $\widehat{\mathrm{GenGap}}(f, \mathcal{D})$, which is the main object predicted by our theory (Theorem 4.2). (iii) **estimated excess risk** $\widehat{\mathrm{Excess}}(f) = \frac{1}{N}\sum_{r=1}^N \big(f(\tilde{\boldsymbol{x}}_r) - f_{\mathrm{true}}(\tilde{\boldsymbol{x}}_r)\big)^2$, the Monte Carlo estimator of $\mathrm{Excess}(f)$ defined in Eq (119); (iv) **Hessian Sharpness** $\lambda_{\max}(\nabla^2\mathcal{L}(\theta_t))$, computed every 500 epochs via PyHessian Yao et al. [2020] to verify the trajectory satisfies BEoS;

Optimization All models are trained with full-batch gradient descent (no momentum, no weight decay) on the squared loss $\mathcal{L}(\theta)$, using learning rate $\eta = 0.2$ for 30000 epochs; the corresponding BEoS threshold is $2/\eta = 10$.

Sweep and Scaling For SCN, we sweep (Figure 2) ambient dimension $d \in \{100, 200, 400\}$; for FCN, we fix $d = 10$ as a baseline where fully connected networks are expected to perform well. Sample sizes range over $n \in \{128, 256, 512, 1024\}$. Results averaged over 5 seeds. To estimate the sample-size scaling, we sweep $n \in \{128, 256, 512, 1024\}$ and repeat the full pipeline over multiple random seeds. We plot $\log \widehat{\mathrm{GenGap}}(f_{\hat{\boldsymbol{\theta}}}, \mathcal{D})$ versus $\log n$ and report the least-squares fitted slope (**SCN**: $d = 100, 200, 400$; **FCN**: $d = 10$).

G.2 Experimental details for Figure 3

Similar setup and metrics as the spherical-data setup above in G.1, with two changes. First, each patch is sampled independently from the unit sphere and concatenated: $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(J)}]$ with

$\boldsymbol{x}^{(j)} \sim \text{Uniform}(\mathbb{S}^{m-1})$, instead of the whole-vector sampling $\boldsymbol{x} \sim \text{Uniform}(\mathbb{S}^{d-1})$. Second, we fix $m = 10$, $J = 8$ (so $d = 80$), $K_{\text{true}} = 50$, $n = 512$, $\sigma = 1$, and train at width $K = 1024$ for 50,000 epochs at $\eta = 0.2$ with gradient clipping. Additional experiments with similar setup in Fig 6, 7 and Table 4.

G.3 Experimental details for Figure 4

We compare convolutional patch geometry with full-image geometry on the CIFAR-10 training set. We extract 3×3 patches with stride 1 and no padding, sample 10^7 patches, and apply standard torchvision normalization. Each patch lies in $\mathbb{R}^{27}$, while each flattened full image lies in $\mathbb{R}^{3072}$. We report two geometry metrics: (a) the cumulative PCA explained-variance ratio, including the number of principal components needed to explain 90% of the variance; and (b) the half-space (Tukey) depth concentration curve $\Psi(T) = \Pr[\hat{d}(\boldsymbol{z}) \geq T]$ for $T \in [0, 0.5]$, whose area is larger for distributions that are harder to shatter. We approximate half-space depth using the random-projection method of Liang et al. [2026], with 512 uniformly random unit directions and 4096 histogram bins per direction. Full Image Space uses randomized PCA for the spectrum and 10000 query points.

G.4 Experimental details for Figure 5

We repeat the above setup in G.1 with two changes. First, inputs are CIFAR-10 images $\boldsymbol{x} \in \mathbb{R}^{3072}$, and SCN uses patches ($3 \times 3 \times 3$, stride 1; $J = 900$, $m = 27$). Second, targets are generated by a *frozen* ground-truth SCN f_{true} of width $K_{\text{true}} = 20$ with the same patchification, $y_i = f_{\text{true}}(\boldsymbol{x}_i) + \xi_i$, $\xi_i \sim \mathcal{N}(0, 1)$; excess risk is reported as the MSE against f_{true} on the standard CIFAR-10 test split ($N = 10000$). We train SCN and FCN, both at width $K = 1024$, on $n_{\text{train}} = 1024$ images for 5000 epochs with $\eta = 0.2$ and gradient clipping. Both networks have more trainable parameters than training examples ($K = n_{\text{train}} = 1024$), and SCN's training loss reaches the noise floor σ^2, ruling out underfitting as the explanation for its small excess risk.

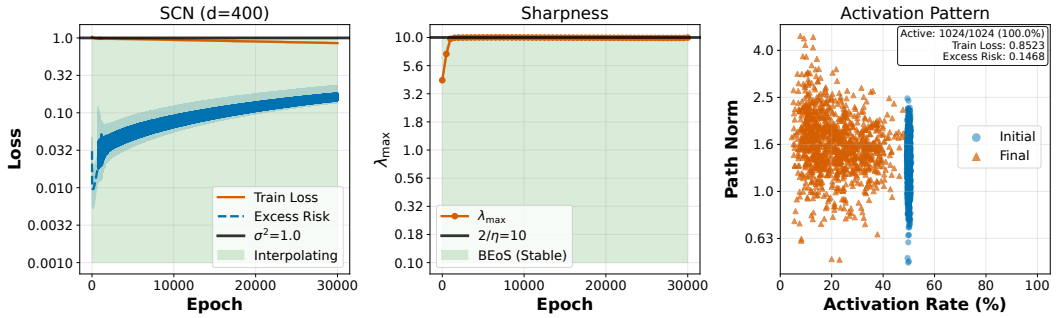


Figure 6: **SCN generalizes on high-dimensional spherical data.(Left)** With $d = 400$ and fixed patch size $m \ll d$, train loss plateaus near σ^2 while excess risk decreases to 0.15, confirming generalization rather than memorization. **(Middle)** Sharpness saturates at BEoS ($\lambda_{\max} \approx 2/\eta$). **(Right)** Neurons spread across moderate activation rates, unlike the sparse isolation in flat interpolation (Figure 3). This validates Theorem 4.2: when $m \ll d$, stability-induced regularization prevents overfitting. Results averaged over 5 seeds with $\eta = 0.2$, trained for 30k epochs.

Table 4: Generalization gap and excess risk for SCN on spherical data (corresponding to Figure 2a). As ambient dimension d increases with fixed patch size m, both metrics decreaseconfirming the "blessing of dimensionality".

d	J	Gen Gap	Excess Risk
100	10	1.283 ± 0.114	0.613 ± 0.101
200	20	0.780 ± 0.076	0.352 ± 0.047
400	40	0.295 ± 0.004	0.147 ± 0.010

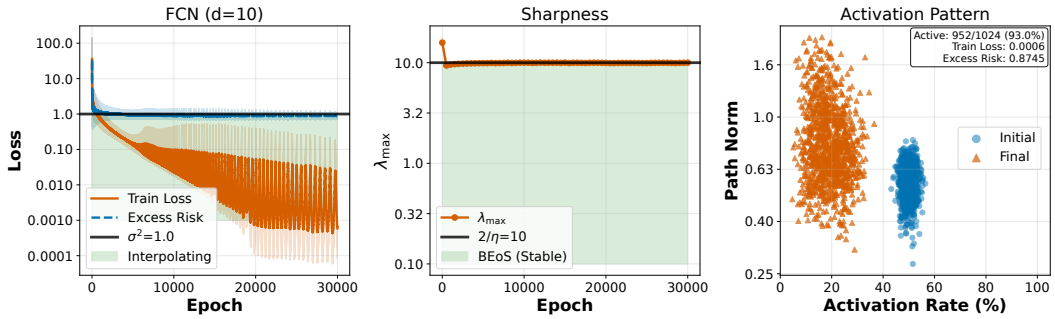


Figure 7: **FCN satisfies BEoS but still memorizes (Left)** FCN ($d = 10$) interpolates noisy labels (train loss $\to 0$) while excess risk remains $\approx \sigma^2$. **(Middle)** Sharpness saturates at BEoS. **(Right)** Activation pattern after training. Despite satisfying BEoS, FCN fails to generalizeconfirming that on spherical data, stability constraints alone are insufficient without convolutional structure. Averaged over 5 seeds; with similar setting as Figure 6

H Comparison with Sparse-Network and Stability-Based Bounds

Table 5 expands the comparison in Section 2. Norm-based analysis controls a class through a prescribed budget on local filter norms, while sparsity-aware certification uses the trained predictor's inactive sets, margins, and PAC-Bayes complexity. Stability-based analysis begins with the curvature condition imposed by the optimization algorithm and derives a function-space constraint. In our setting, Theorem 4.1 makes this implication explicit: the step size controls a weighted path norm, and the receptive fields determine its weights through the training patch multiset. Theorem 4.2 then turns this constraint into a generalization guarantee for the BEoS class.

Table 5: **Extended comparison of sparse-network and stability-based guarantees.** The upper block uses prescribed norm budgets or measured predictor statistics. The lower block derives complexity control from optimization stability for two-layer ReLU networks. Wu–Su study interpolation on a noiseless sphere with no hidden bias, and Qiao et al. control the gap on an interior interval. The final three rows show how data geometry and connectivity affect BEoS guarantees.

Work	Architecture	Starting point	Guarantee
[Galanti et al., 2023]	Deep sparse nets / CNNs	Prescribed product of local filter norms	Bound depending on filter norms, connectivity, and local input energy
[Muthukumar and Sulam, 2023]	Deep ReLU FCNs	Stable inactive sets, margin, PAC-Bayes KL	Predictor-dependent bound using activation sparsity
[Wu and Su, 2023]	FCN, no hidden bias	GD/SGD stability; noiseless sphere	$\tilde{O}(d/(\eta^2 n))$ (GD), $\tilde{O}(1/(\eta^2 n))$ (SGD)
[Qiao et al., 2024]	Univariate FCN	BEoS	$\tilde{O}(n^{-2/5})$ on an interior interval
[Liang et al., 2025]	FCN with bias	BEoS; uniform ball	$\tilde{O}(n^{-1/(2d+4)})$ upper, $\Omega(n^{-2/(d+1)})$ lower
[Liang et al., 2026]	FCN with bias	BEoS; uniform sphere	$\Omega(1)$ lower bound
Ours	**SCN with bias**	**BEoS; uniform sphere**	$\tilde{O}\left(n^{-\frac{(d-m)(d+3)}{2(3d^2-md+3d-3m)}}\right)$

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 list specific contributions (numbered 1–4) that are explicitly linked to the paper's sections, and the scope (twolayer ReLU networks, idealized data settings) is honestly stated in the "Scope of analysis" paragraph.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The end of Section 1 ("Scope of analysis") discuss the restriction to twolayer networks, the need for idealized data distributions to enable analysis, and the challenge of extending the framework to deep networks. These limitations are openly acknowledged.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All theorems (4.1, 4.2, 4.3) are stated with explicit assumptions, and complete proofs are provided in Appendices B, C, and D, with supporting lemmas properly referenced.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Appendix E provides detailed descriptions of the data generation (uniform on sphere, noise model), model architecture (SCN with given m, K, etc.), training procedure (fullbatch GD, learning rate, number of epochs, number of seeds), and the metrics used. This information is sufficient to reproduce the synthetic experiments; CIFAR10 is a public benchmark.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We will release code and scripts to reproduce all experiments upon acceptance. The repository will include a README with environment setup and execution instructions. CIFAR-10 is a publicly available benchmark; synthetic data can be generated by the provided code.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix E reports learning rate, batch mode (fullbatch), number of epochs, network width, patch size, number of seeds, and the procedure for measuring the generalization gap and sharpness.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Table 2 reports mean $\pm$ standard deviation over 5 seeds; Figures 1, 5, and 6 similarly state that results are averaged over multiple seeds, and the quantity of variability (across different random seeds) is indicated.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: All experiments were run on a MacBook Air (M1, 2020) with 8 GB RAM. Each experimental run (30k epochs) completed in under 2 hours, demonstrating that the experiments require minimal compute resources.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work is foundational theory with smallscale synthetic experiments and analysis of a public benchmark (CIFAR10). There are no ethical concerns regarding human subjects, privacy, or fairness.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This is a theoretical paper that studies the implicit regularization of gradient descent in convolutional architectures; it does not introduce new applications, datasets, or deployed models, and therefore has no direct societal impact that requires discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release any models or datasets that could pose a misuse risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The paper uses CIFAR-10 only as a standard public benchmark for empirical analysis and does not redistribute or repackage the dataset. We access it through standard public tooling and use it solely for research experiments.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not introduce any new datasets, models, or code packages.

Guidelines:

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

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

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

Answer: [N/A]

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

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve human subjects research.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: No LLMs were used as a component of the research methodology.

Guidelines:

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