What Architectural Inductive Bias Makes Diffusion Models Succeed? A Perspective from the Implicit Regularization of Gradient Descent

Anonymous Authors[1]

Abstract

Diffusion and flow-based models succeed by training a neural network to predict noise or velocity from corrupted inputs. But why this training succeeds is not fully explained by the denoising objective alone, because the same objective can fail completely when the architecture of the denoiser changes. We study the role of architecture through the lens of gradient dynamics. The key property we identify is sparse connectivity: each neuron receives input from only a small subset of coordinates, a design shared across convolutional and transformer denoisers. We prove that sparse connectivity makes memorization strictly harder than in fully connected networks by shifting the implicit regularization of gradient descent away from the ambient input geometry and onto a collection of low-dimensional patches. Controlled denoising experiments corroborate this theory, and an extension to deep denoisers shows that clean-data prediction keeps internal representations lower-dimensional across layers. Our results point to a concrete mechanism of architectural inductive bias: the architecture determines the geometry on which gradient descent operates, and through this geometry it shapes which solutions training can find.

1. Introduction

Diffusion and flow-based generative models learn to sample from a data distribution by training a neural network to reverse a noise process (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Liu et al., 2023). At each training step, the network receives a corrupted input—a clean sample with Gaussian noise added—and is asked to recover information about the original data. The standard objective trains the network to predict the noise component directly, which is equivalent to learning the score function of the noisy distribution (Song et al., 2021). The denoising neural network is thus the central learnable component of these systems.

Substantial effort has gone into understanding what makes the denoising objective effective. But the objective is only part of the story. A recent experiment by Li & He (2025) makes this visible in a simple setting. They trained a fully connected network (FCN) to predict noise on a two-dimensional spiral distribution embedded in a high-dimensional space. The objective, noise schedule, and optimizer were all standard. Yet the network failed: it could not recover the spiral structure, producing diffuse samples that spread into irrelevant ambient directions. This is the exact same objective that powers state-of-the-art image generators. The variable that changes between failure and success is not the objective. It is the architecture. The denoisers used in practice are never fully connected. They are either convolutional U-Nets (Ronneberger et al., 2015; Song & Ermon, 2020; Dhariwal & Nichol, 2021) or patch-based transformers (Peebles & Xie, 2023; Bao et al., 2023; Ma et al., 2024), and when the architecture is right, the same objective can learn complex distributions in far higher dimensions.

Why does the architecture matter so much, and what property of these architectures is responsible? In this paper, we approach this question through the lens of how architectural design changes the gradient dynamics of training. We argue that the architecture is not merely a means of restricting the function class. It determines the data geometry on which gradient descent operates, and through this geometry, it shapes the implicit regularization of gradient descent.

We isolate one property shared across these architectures: each neuron receives input from only a small subset of coordinates. We formalize this as *sparse connectivity* and analyze how it reshapes the implicit regularization of gradient descent. Under this framework, we prove that sparse connectivity fundamentally changes how the implicit regularization acts: instead of operating on the full high-dimensional input, gradient descent operates on a collection of low-dimensional patches, and the geometry of this patch collection governs the strength of the resulting regularization (Theorem 4.2). We prove that when the patches are small and concentrated,

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

sparsely connected networks are harder to memorize on and admit non-vacuous generalization guarantees on spherical data where fully connected networks provably fail (Theorem 4.3), and we construct a matching failure mode when the patch geometry is itself unstructured (Theorem 4.4).

We complement this theory with empirical evidence on three fronts. First, we reproduce the fully connected failure in a controlled denoising setting, and show that performance degrades systematically as the local input dimension per neuron increases toward the fully connected limit (Section 3). Second, we verify the quantitative predictions of the stability framework on synthetic data (Section 4). Third, we extend this geometric perspective to deep denoisers, and show that the output parameterization plays a role analogous to the input interface: clean-data prediction provides an analytic long skip that keeps the internal representations of the network lower-dimensional across layers (Section 5). This provides a geometric interpretation, from the perspective of gradient dynamics, for the empirical success of clean-data prediction reported by Li & He (2025).

Taken together, our results identify a concrete mechanism through which architectural inductive bias operates in denoising generative models. The architecture does not merely restrict the function class. It determines the data geometry on which gradient descent acts, and through this geometry, it shapes which solutions training can find.

2. Related Work

Diffusion and Flow generative modeling. Diffusion-based generative models (Ho et al., 2020; Song et al., 2021) cast generation as the iterative denoising of a Gaussian-corrupted input by a learned denoiser. Subsequent work has refined individual components: noise schedules and sampler design (Karras et al., 2022; Lin et al., 2024), timestep-weighted optimization (Hang et al., 2023), latent-space modeling with cross-attention conditioning (Rombach et al., 2022), few-step distillation (Song et al., 2023), and continuous-time transport through flow matching and rectified flow (Lipman et al., 2023; Liu et al., 2023). Comparative studies that hold the denoising objective fixed and vary the remaining components (Karras et al., 2022; Ma et al., 2024) establish that the denoiser network itself is a first-order determinant of generative performance, leaving open the question of *which* architectural properties are responsible.

Architectures for denoising generative models. The U-Net (Ronneberger et al., 2015) has served as the canonical denoiser since the earliest score-based generative models (Song & Ermon, 2020); Dhariwal & Nichol (2021) subsequently characterized the contributions of its residual blocks, skip connections, and attention placements through systematic ablations. Beyond U-Net variants, transformer-based denoisers operating on tokenized image patches have been introduced as alternatives: U-ViT preserves the U-Net's long skip connections in token space (Bao et al., 2023); DiT establishes power-law scaling for latent-patch denoisers (Peebles & Xie, 2023); DiffiT introduces time-conditional self-attention tailored to the denoising trajectory (Hatamizadeh et al., 2024); and SiT decouples backbone, objective, and noise transport to isolate the contribution of each (Ma et al., 2024). A common structural feature across these architectures is that each hidden unit receives input from a spatially localized region of the noisy input, a property our analysis formalizes as *sparse connectivity*.

Generalization and memorization in diffusion models. A recent line of diffusion-specific theory attributes generalization to locality in the denoiser. Kamb & Ganguli (2025) derive an analytic patch-mosaic optimum under locality and equivariance constraints that closely predicts trained U-Nets; Niedoba et al. (2025) approximate pretrained denoisers across architectures by patch-level empirical aggregation; and An et al. (2025) report that early-layer attention locality is predictive of generalization in DiTs. From a training-dynamics perspective, Bonnaire et al. (2025) identify a timescale separation between generation and memorization. Memorization itself has been localized to cross-attention magnitudes, specific neurons, and local image regions (Wen et al., 2024; Ren et al., 2024; Hintersdorf et al., 2024; Chen & collaborators, 2025; Zhang et al., 2026). We provide optimization-theoretic counterpart of these observations.

3. Successful Denoisers are Patch-Based

Diffusion models are typically defined by their training objective. A neural network receives a corrupted input and predicts a quantity related to either the clean data or the injected noise. This view explains why denoising is a statistically meaningful learning problem, but it does not explain why the denoising network used in practice can learn the structure of the target distribution through gradient descent. In this section, we take architecture as the variable and show that patch-based processing can substantially change the outcome of the same noise-prediction task.

3.1. Background: diffusion and flows

Diffusion models define a path from clean data to noise and train a neural network to reverse this path (Ho et al., 2020). Let $\boldsymbol{x}_0 \sim p_{\text{data}}$ be a clean sample and let $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ be Gaussian noise. A common forward process produces a noisy input

$$\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon},$$

where t indexes the noise level. In the noise-prediction parameterization, the denoising network $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is trained

by

$$\mathcal{L}_{\text{noise}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}\|_2^2\right].$$

This is the objective used in our experiments.

Noise prediction is directly connected to score estimation. For the additive noising model $\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, Tweedie's formula gives

$$\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t) = \frac{\mathbb{E}[\boldsymbol{x}_0 \mid \boldsymbol{x}_t] - \boldsymbol{x}_t}{\sigma_t^2}.$$

Since $\boldsymbol{x}_t = \boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, learning the conditional mean of $\boldsymbol{\epsilon}$ is equivalent to learning the score up to a scalar factor. For general affine schedules $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$, the same relationship holds after rescaling. This is why noise prediction became a standard way to train score-based diffusion models (Ho et al., 2020; Song et al., 2021).

The same family of models also has an ODE or flow view. If p_t denotes the marginal distribution of $\boldsymbol{x}_t$, sampling can be written through a probability-flow ODE whose vector field is determined by the score (Song et al., 2021). Flow matching and rectified flow formulate this idea more directly by choosing an interpolation $\boldsymbol{z}_t = a_t\boldsymbol{x}_0 + b_t\boldsymbol{\epsilon}$ and training a neural network to predict the velocity $\boldsymbol{v}_t = a_t'\boldsymbol{x}_0 + b_t'\boldsymbol{\epsilon}$ along the path (Lipman et al., 2023; Liu et al., 2023). These formulations differ in parameterization and weighting, but they share the same basic structure. The model is trained on corrupted high-dimensional inputs and must learn a vector field that separates data structure from injected noise.

Modern diffusion models typically parameterize this denoising network with architectures built around spatial structure. Early score-based and diffusion models largely used convolutional U-Net backbones (Ronneberger et al., 2015; Song & Ermon, 2020; Song et al., 2021; Dhariwal & Nichol, 2021). More recent transformer-based denoisers process noisy images or latents as patch tokens, as in U-ViT, DiT, DiffiT, and SiT (Bao et al., 2023; Peebles & Xie, 2023; Hatamizadeh et al., 2024; Ma et al., 2024). Thus, across convolutional and transformer families, patch-based processing has become a common design pattern in denoising generative models.

3.2. A toy experiment with architecture as the variable

Inspired by the toy setup of Li & He (2025), we construct a simple high-dimensional denoising problem whose intrinsic structure is easy to visualize. Let $\hat{\boldsymbol{x}} \in \mathbb{R}^2$ be a sample from a two-dimensional spiral distribution and let $\boldsymbol{P} \in \mathbb{R}^{D\times 2}$ be a fixed column-orthogonal matrix. The observed clean sample is

$$\boldsymbol{x}_0 = \boldsymbol{P}\hat{\boldsymbol{x}} \in \mathbb{R}^D.$$

The model never observes $\hat{\boldsymbol{x}}$ or $\boldsymbol{P}$. It only sees corrupted samples $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$ in the ambient space and is trained with the noise-prediction loss $\mathcal{L}_{\text{noise}}$. We set $D = 256$, which makes the observed space much larger than the intrinsic dimension while keeping the generated samples easy to visualize through projection by $\boldsymbol{P}^\top$.

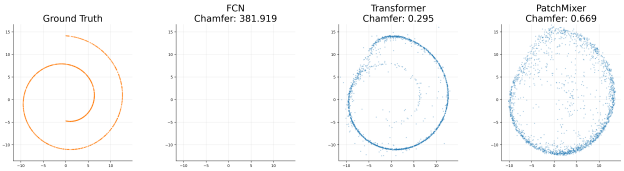


Figure 1. **Architecture affects noise prediction on a randomly embedded spiral.** The ground-truth spiral is embedded into $\mathbb{R}^{256}$, models are trained in the ambient space, and samples are visualized after projection back to the original two-dimensional coordinates. Chamfer distance compares generated and ground-truth point clouds, where lower is better. The FCN fails, while the Transformer and PatchMixer recover the spiral much more accurately under the same noise-prediction objective.

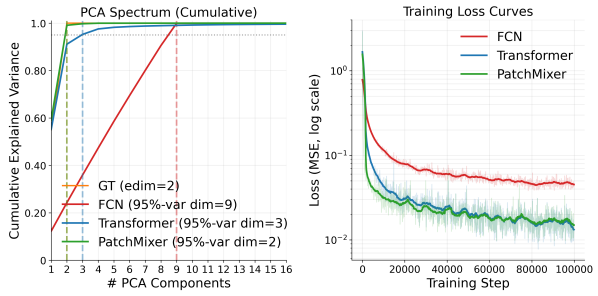
All models predict noise, so the prediction target, loss, noise schedule, optimizer, and sampling procedure are fixed. We vary only the architecture of the score estimator. We compare an FCN with two patch-based architectures, a vision Transformer (Dosovitskiy et al., 2021) and an MLP-Mixer-style model (Tolstikhin et al., 2021), which we call PatchMixer. Both patch-based models partition the 256 ambient coordinates into 16 patches of dimension 16. The Transformer linearly embeds each patch into a 96-dimensional token and processes the token sequence with single-head self-attention layers. PatchMixer uses the same patch tokens and alternates token-mixing and channel-mixing ReLU layers. The comparison therefore keeps the denoising task fixed while changing how the high-dimensional noisy vector is first presented to the network.

Figure 1 shows a clear separation between the architectures. The FCN does not recover the spiral and produces samples that spread away from the intrinsic support, while the Transformer and PatchMixer produce much more faithful samples under the same objective. The training curves and PCA spectra in Figure 2 give a more detailed view of the failure.

3.3. Patch size controls the difficulty of noise prediction

The previous comparison separates full-vector processing from patch-based processing. We now vary the patch size within the patch-based models to test whether the local input dimension itself matters. We embed the same two-dimensional spiral into a 32×32 dimensional ambient space and train Transformer and PatchMixer denoisers with the same noise-prediction objective. Only the patch size is changed.

Figure 3 shows that the quality of noise prediction depends strongly on patch size. For both Transformer and PatchMixer, smaller patches generally produce samples that bet-



(a) PCA spectrum. *(b)* Training curves.

Figure 2. **Optimization and geometry of generated samples.** **(a)** The ground-truth distribution is intrinsically two-dimensional. PatchMixer reaches 95% explained variance with 2 principal components and the Transformer with 3, while the FCN requires 9, indicating that its generated samples spread into extra ambient directions. **(b)** The models have comparable parameter counts, with the FCN slightly larger in our implementation, but the FCN decreases the noise-prediction loss more slowly and plateaus at a higher value.

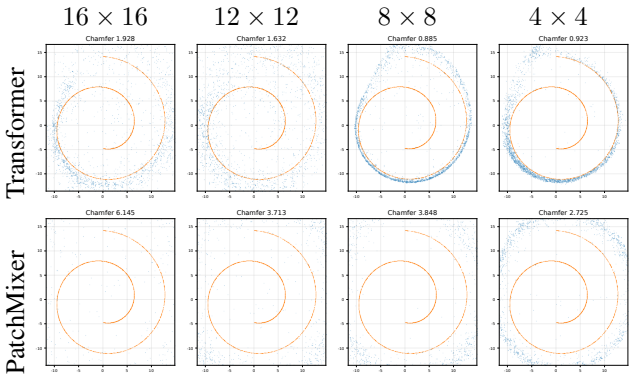


Figure 3. **Patch-size ablation for noise prediction.** A two-dimensional spiral is embedded into a 32×32 dimensional ambient space and generated samples are visualized after projection to the intrinsic two-dimensional coordinates. Columns vary the patch size and rows compare Transformer and PatchMixer denoisers. Smaller patches generally improve sample quality, while larger patches produce more diffuse generations and higher Chamfer distances.

ter concentrate around the ground-truth spiral, while larger patches make each token process a higher-dimensional noisy input and lead to more diffuse generations. The benefit of patch-based architectures therefore comes not from patching itself, but from the patches being sufficiently small. It comes from presenting the denoising network with sufficiently small local views of the corrupted input.

The takeaway of this section is that architecture changes the learnability of the same noise-prediction task. Patch-based models can recover low-dimensional structure from high-dimensional noisy observations in settings where an FCN fails, and reducing the patch size improves this effect. The next section formalizes this observation through a stability-based analysis of sparsely connected networks and shows how patch size enters the implicit regularization induced by gradient descent.

4. Theoretical Analysis: Sparsely Connected Networks Below the Edge of Stability

The experiments in Section 3 isolate a clean empirical pattern: under the same noise-prediction objective, optimizer, and noise schedule, a fully connected denoiser fails to learn the low-dimensional signal buried in high-dimensional noise, while a patch-based denoiser succeeds. This section explains why. The answer lies in how the data geometry interacts with the implicit regularization of gradient descent via the design of network architecture.

For FCNs on high-dimensional isotropic inputs, Liang et al. (2025; 2026) identified a failure mode called *neural shattering*: if the inputs of training data sets concentrate near a hypersphere (e.g. high-dimensional Gaussian distribution), one can separate any single training point from all others by a hyperplane that cuts off a tiny cap. A ReLU neuron oriented along that direction and thresholded at the base of the cap activates only on very few samples. Because the neuron fires on a negligible fraction of the data, its gradient updates are controlled entirely by that single sample. The neuron learns a dedicated per-sample correction rather than a shared feature. The resulting network consists of many rare-activation, high-magnitude units that collectively interpolate the training labels.

Crucially, Liang et al. (2025; 2026) prove that when the input distribution is spherical, these memorization ReLU FCNs exist in a regime where gradient descent on neural networks typically enters, the *below the edge of stability* (BEoS) regime: the largest eigenvalue of the training loss Hessian increases, and then hovers around a stability threshold controlled by the step size, where (Cohen et al., 2021). They also empirically verified that GD indeed finds these memorization solutions below the edge of stability. Moreover, they proved that the generalization performance smoothly degrades as data concentrates near a hypersphere. However, when it comes to diffusion, at moderate to high noise levels, each training sample $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ is dominated by the Gaussian term, which yields this bad data geometry making FCNs vulnerable to memorization. We propose that this could be a potential reason to explain why gradient-descent trained FCNs cannot be a good denoiser/score estimator in practice.

Definition 4.1 (Below Edge of Stability (Qiao et al., 2024, Definition 2.3))**.** Let $\{\boldsymbol{\theta}_t\}_{t\geq 1}$ be a GD trajectory on $\mathcal{L}$ with step size η. Any parameter $\boldsymbol{\theta}_t$ with $\lambda_{\max}\big(\nabla^2 \mathcal{L}(\boldsymbol{\theta}_t)\big) \leq \frac{2}{\eta}$ is called a solution *Below the Edge of Stability (BEoS)*.

The question then becomes: what changes when the denoiser is patch-based? A patch-based architecture departs from the fully connected case in one essential way. Instead of presenting each neuron with the full d-dimensional input, it partitions the input into patches. To capture this property, we study *sparsely connected networks* (SCNs).

4.1. Problem Setup

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 $[n] = \{1, 2, \ldots, n\}$.

Sparse connectivity pattern. 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. Patch-based architectures, such as the patch based denoisers in Section 3, are instances of this design with $m \ll d$.

Sparsely connected network with weight sharing (SCN). This is the minimal architecture that captures patch-based processing. Given receptive fields $\mathcal{S}$ and width $K \in \mathbb{N}$,

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

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 $\boldsymbol{p}_j$ if $\boldsymbol{w}_k^\mathsf{T} \boldsymbol{p}_j > b_k$. Let $\boldsymbol{\Theta}^{\mathcal{S}\mathrm{CN}}$ be the parameter set of all $\mathcal{S}$CNs of *any finite* width. When $m = d$ and $J = 1$, the network reduces to a fully connected architecture, and our analysis recovers the FCN results of (Liang et al., 2026).

Data and loss. 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 $\mathcal{L}(\boldsymbol{\theta}) := \frac{1}{2n} \sum_{i=1}^n (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2$. Throughout, GD refers to vanilla gradient descent with learning rate $0 < \eta < 2$.

Now we define the stability regularized parameter subset:

$$\boldsymbol{\Theta}^{\mathcal{S}\mathrm{CN}}_{\mathrm{BEoS}}(\eta, \mathcal{D}) := \big\{ \boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{S}\mathrm{CN}} \mid \lambda_{\max}\big(\nabla^2 \mathcal{L}(\boldsymbol{\theta})\big) \leq \frac{2}{\eta} \big\}. \quad (2)$$

Our goal is to understand how the architecture, encoded by the receptive field system $\mathcal{S}$, changes the effective capacity of this BEoS class.

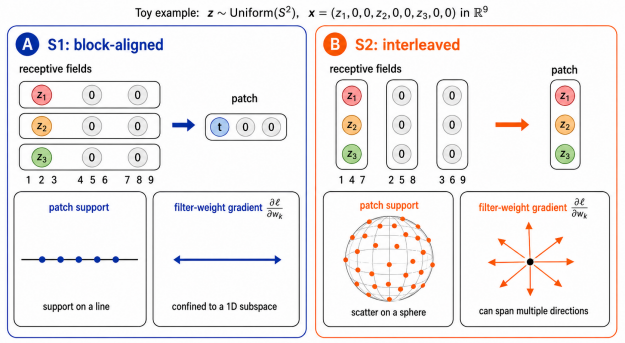


Figure 4. **Data $\times$ architecture determines the geometry that gradient descent interacts with.** The same spherical ambient data produces a one-dimensional patch cloud under $\mathcal{S}_1$ and a three-dimensional spherical patch cloud under $\mathcal{S}_2$.

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 $\mathrm{GenGap}(f, \mathcal{D}) := |\mathcal{R}(f) - \widehat{\mathcal{R}}_{\mathcal{D}}(f)|$.

4.2. A Toy Example: How Sparse Connectivity Shapes the Gradient Dynamics

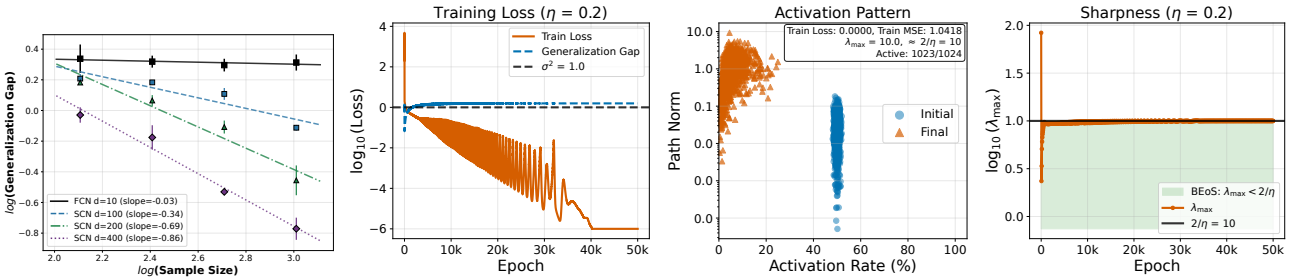
We use a toy example to preview how sparse connectivity in the network shapes the gradient dynamics itself. Consider an ambient input $z \sim \mathrm{Uniform}(\mathbb{S}^2)$ embedded into $\mathbb{R}^9$ by padding each coordinate with two zeros. The ambient data is spherical, so under the analysis of (Liang et al., 2026), a fully connected network trained on this data receives no meaningful constraint from the BEoS condition. Now equip a two-layer SCN with two different receptive field systems:

- $\mathcal{S}_1$ (block-aligned): each receptive field groups 3 consecutive coordinates. Every patch has the form $(t, 0, 0)$, so the patch geometry is one-dimensional.
- $\mathcal{S}_2$ (interleaved): each receptive field groups coordinates modulo the stride. A patch recovers (z_1, z_2, z_3), so the patch geometry is spherical.

The gradient of the training loss with respect to a shared filter $\boldsymbol{w}_k$ exposes the difference. For the model (1),

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{w}_k} = \frac{v_k}{nJ} \sum_{i,j}^{n,J} \underbrace{r_i \mathbb{1}\{\boldsymbol{w}_k^\mathsf{T} \pi_j(\boldsymbol{x}_i) > b_k\}}_{\text{residue and activation scalars}} \underbrace{\pi_j(\boldsymbol{x}_i)}_{\text{patch vectors}} \quad (3)$$

where $r_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i$ is the residual. The update to $\boldsymbol{w}_k$ is a linear combination of the patches 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.



(a) Log–log scaling with n. *(b)* Interpolation after 40k. *(c)* Memorization filters exist. *(d)* Below the EoS.

Figure 5. **Numerical validations via synthetic experiments. (a)** $\mathrm{GenGap}(f_{\boldsymbol{\theta}}, \mathcal{D})$ 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. **(b)** Training loss and generalization gap, showing that the SCN interpolates noisy labels in the random spherical-patch setting. **(c)** 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, consistent with the failure mode in Theorem 4.4: rare activations can fit labels while keeping the sharpness small. **(d)** The largest Hessian eigenvalue hovers near $2/\eta \approx 10$, indicating the EoS regime.

The patch multiset is highly concentrated, and any filter that activates on a meaningful fraction of patches must align with the shared direction. Under $\mathcal{S}_2$, activated patches span the entire three-dimensional patch space.

The two receptive field systems thus produce fundamentally different gradient dynamics on the same ambient data, and the sets of solutions that GD can stably reach differ dramatically. The rest of this section makes this intuition precise through the lens of the BEoS condition. Theorem 4.2 shows that the BEoS condition bounds a weighted path norm whose weight is governed by the patch geometry. Theorem 4.3 and Theorem 4.4 then develop the two contrasting regimes previewed by $\mathcal{S}_1$ and $\mathcal{S}_2$: one enables generalization, and the other fails.

4.3. Characterization of the Stability Regularization

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 $g_{\mathcal{D},\mathcal{S}} : \mathbb{S}^{m-1} \times \mathbb{R} \to \mathbb{R}$ by

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

This function evaluates a neuron's activation boundary $(\boldsymbol{u}, t)$ against the patch multiset. The term $\mathbb{P}(\boldsymbol{u}^\top \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t)^2$ is the squared activation mass: a neuron that fires on a large fraction of patches incurs a large penalty on its path norm, while one that fires on few patches incurs a weak penalty.

Theorem 4.2 (BEoS implies patch-geometry-weighted path norm bound)**.** *Fix $\mathcal{D}$ and $\mathcal{S}$. For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathrm{SCN}}$,*

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

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

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

The proof is in Appendix D. The core connection between the Hessian and $g_{\mathcal{D},\mathcal{S}}$ is detailed in Proposition D.10. The architecture $\mathcal{S}$ enters solely through this weight function: different receptive field systems produce different patch multisets, hence different penalty structures. When $m = d$ and $J = 1$, the network reduces to a fully connected architecture and (6) recovers the result of (Liang et al., 2026).

Returning to our toy example makes this concrete. Under $\mathcal{S}_1$, the patch multiset is essentially one-dimensional and highly concentrated. Any neuron that activates on a non-negligible fraction of patches must cut through the dense interior, so the activation probability $\mathbb{P}(\boldsymbol{u}^\top \boldsymbol{X}_{\mathcal{D}}^{\mathcal{S}} > t)$ is bounded away from zero and $g_{\mathcal{D},\mathcal{S}}$ has a positive lower bound. The BEoS condition therefore enforces a meaningful path-norm constraint. Under $\mathcal{S}_2$, the patch multiset is a three-dimensional sphere. A neuron can isolate an individual patch with a carefully placed hyperplane, driving the activation probability arbitrarily close to zero and making $g_{\mathcal{D},\mathcal{S}}$ arbitrarily small. The BEoS constraint becomes vacuous. The next two subsections develop the quantitative consequences of this dichotomy.

4.4. Two Regimes: Generalization versus Memorization

The next two theorems formalize the contrasting behaviors previewed by $\mathcal{S}_1$ and $\mathcal{S}_2$.

Theorem 4.3 (Generalization under structured patch geometry)**.** *Let the marginal distribution of $\boldsymbol{x}$ be* $\mathrm{Uniform}(\mathbb{S}^{d-1})$*, and let $\mathcal{D}$ be a dataset of n i.i.d. samples. Assume $d > 3$ and $1 \le m < \frac{d(d-3)}{d+3}$. For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathrm{SCN}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$ with $\|f_{\boldsymbol{\theta}}\|_\infty \le M$, with probability $\ge 1 - 2\delta$,*

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

The crucial feature of this bound is its scaling with ambient dimension. When the patch size m is fixed and $d \to \infty$, the exponent approaches $-\frac{1}{6} + O(m/d)$, so there is *no curse of dimensionality*. In the same spherical setting, FCNs admit no non-trivial guarantee under the BEoS condition (Liang et al., 2026). The geometric reason, formalized in Proposition D.11, is that when $m \ll d$, the patch projections concentrate near the origin of $\mathbb{R}^m$. The patch multiset becomes low-dimensional and concentrated, resisting the kind of pointwise isolation that weakens the stability constraint. The path-norm penalty in Theorem 4.2 therefore becomes effective. The formal proof appears in Appendix E, and Figure 5 numerically verifies the predicted scaling.

Theorem 4.4 (Memorization below the edge of stability)**.** *Assume all training patches have norm at most one, and every nonzero-labeled training example contains a unit-norm patch that appears nowhere else in the full training patch multiset. There exists a network of the form (1) with width $K \le n$ that interpolates $\mathcal{D}$ and satisfies*

$$\lambda_{\max}\big(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}\big) \le 1 + \frac{D^2 + 2/J^2}{n}. \tag{8}$$

Theorem 4.4 shows that the BEoS condition alone does not prevent memorization when the patch multiset admits single-patch isolation. From the weighted path-norm perspective of Theorem 4.2, the same rare activations correspond to small values of $g_{\mathcal{D},\mathcal{S}}$, making the stability-induced constraint weak. The detailed proof can be found in Appendix F. An empirical demonstration is shown in Figure 5. The experiment uses random spherical patches (case B in Figure 4). Since these patches are uniformly sampled from $\mathbb{S}^{m-1}$, the uniqueness condition in Theorem 4.4 holds generically.

Taken together, Theorems 4.3 and 4.4 delineate two regimes governed by the patch geometry induced by the receptive-field system $\mathcal{S}$. When $\mathcal{S}$ and the data produce a concentrated patch multiset that resists single-patch isolation, the BEoS condition yields an effective regularity constraint and generalization is possible. When the patch multiset contains patches that can be isolated by ReLU half-spaces, the constraint collapses and memorization becomes compatible with stability. For a fixed training dataset, the architectural design $\mathcal{S}$ determines the induced patch multiset and hence which regime the network operates in.

5. Towards Deep Denoisers: Output Skips Shape Representation Geometry

The previous sections focus on the geometry of the input interface. In a two-layer sparsely connected model, each filter acts on a patch of the corrupted input, so patch size controls the dimension of the object processed by gradient descent. Deep denoisers introduce another effect. A block at depth k does not process the raw noisy input directly; it processes the representation produced by all previous blocks. Thus the relevant geometry is no longer fixed at the input layer, but evolves along the residual stream.

This perspective is useful for interpreting the clean-prediction result of Li & He (2025). They show that large-patch pixel Transformers can fail catastrophically when directly trained to predict noise or velocity, while the same architecture becomes effective when it predicts the clean image. Their explanation is based on the manifold assumption: clean images are structured and low-dimensional, whereas noise and velocity are high-dimensional quantities. We agree with this target-space intuition, but we use it to highlight an architectural mechanism. Clean prediction changes the output parameterization of the denoiser. It gives the model an analytic long skip from the noisy input to the final noise or velocity predictor, so that the learned branch can focus on producing clean structure.

We illustrate this effect in the same $D = 256$ randomly embedded spiral setup used above. Both models use the same deep fully connected backbone and the same training procedure in Section 3. The only change is the output parameterization. Direct ϵ-prediction produces diffuse samples, while x-prediction recovers the spiral much more accurately in Figure 6.

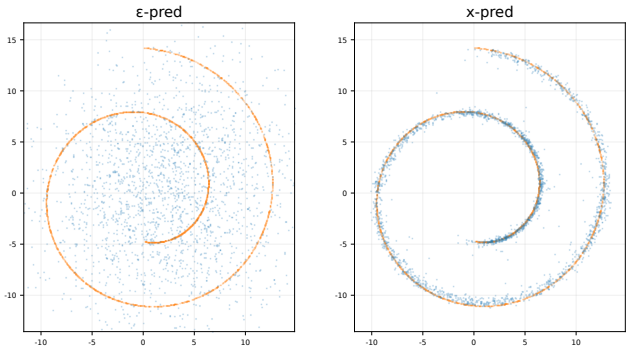
To see the architectural content of this parameterization, consider the interpolation

$$\boldsymbol{z}_t = (1-t)\boldsymbol{x} + t\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}),$$

where $t = 0$ is clean data and $t = 1$ is pure noise. If the learned branch predicts the clean sample, $f_\theta(\boldsymbol{z}_t, t) \approx \boldsymbol{x}$, then the corresponding noise predictor is

$$\widehat{\boldsymbol{\epsilon}} = \frac{\boldsymbol{z}_t - (1-t) f_\theta(\boldsymbol{z}_t, t)}{t} = \frac{1}{t}\boldsymbol{z}_t - \frac{1-t}{t} f_\theta(\boldsymbol{z}_t, t).$$

Thus, as a noise predictor, x-prediction is a fixed output-side skip from $\boldsymbol{z}_t$ plus a learned clean branch. The same



(a) ϵ-prediction. *(b)* x-prediction.

Figure 6. **Clean prediction rescues the same deep denoising backbone.** Both models are trained on a two-dimensional spiral embedded into $\mathbb{R}^{256}$ and visualized after projection to the intrinsic coordinates. Direct ϵ-prediction produces diffuse samples, while x-prediction recovers the spiral support much more accurately.

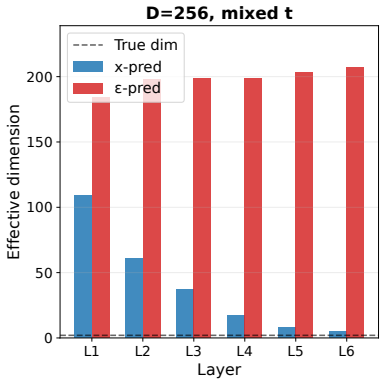


Figure 7. **Effective dimension under mixed noise levels.** Hidden representations are aggregated over the training distribution of t. The x-prediction model forms lower-dimensional representations than the direct ϵ-prediction model across layers.

observation holds for velocity prediction after the corresponding change of coefficients. The coefficients are determined by the schedule, so this is a specific parameterization rather than a generic residual connection. Under the noise-prediction loss, the learned branch is trained by

$$\|\widehat{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}\|_2^2 = \frac{(1-t)^2}{t^2} \|f_\theta(\boldsymbol{z}_t, t) - \boldsymbol{x}\|_2^2.$$

The high-dimensional noisy input is carried to the output analytically, while the network branch receives a clean-image regression objective.

This changes the forward pressure on the residual stream. Write a depth-L backbone schematically as

$$\boldsymbol{h}^0 = E\boldsymbol{z}_t,\ \boldsymbol{h}^k = \boldsymbol{h}^0 + \sum_{\ell=1}^{k} B_\ell(\boldsymbol{h}^{\ell-1}, t),\ f_\theta(\boldsymbol{z}_t, t) = R(\boldsymbol{h}^L).$$

The input to block k is $\boldsymbol{h}^{k-1} = \boldsymbol{h}^0 + \sum_{\ell=1}^{k-1} B_\ell(\boldsymbol{h}^{\ell-1}, t)$. Under direct ϵ-prediction, the final readout must recover high-dimensional noise. Even if early blocks begin to extract the clean low-dimensional signal, later blocks are pressured to transform the residual stream back into a representation from which noise can be read out. The same stream has to both build useful denoising features and carry high-dimensional noisy information.

Under x-prediction, the learned branch has a consistent target across depth. Each block improves the representation only insofar as it helps the final readout approximate $\boldsymbol{x}$. The fixed skip already supplies the explicit $\boldsymbol{z}_t$ term needed to form $\widehat{\boldsymbol{\epsilon}}$, so the residual stream does not need to preserve every high-dimensional noisy direction. Forward propagation is therefore biased toward a progressively cleaner representation: as depth increases, the input to the next block becomes more structured and lower-dimensional. This is the sense in which clean prediction can be viewed as an architectural design. It shapes the geometry seen by later layers.

We measure this effect by computing the effective dimension of hidden states, defined as the number of principal components needed to explain 95% of the variance. Figure 7 reports the layerwise effective dimension. For x-prediction, the dimension decreases across layers, while for direct ϵ-prediction, the representation remains high-dimensional through most of the network. This matches the residual-stream interpretation: the clean branch learns a progressively denoised geometry, while the noise-prediction stream stays tied to noisy sources. In this way, the x-prediction model forms lower-dimensional internal representations across the network, while direct ϵ-prediction stays far from the intrinsic dimension.

This lower-dimensional representation geometry is also relevant to memorization. A model whose intermediate features align with the shared low-dimensional structure has less incentive to allocate capacity to isolated high-dimensional residuals. In contrast, a high-dimensional residual stream gives the later layers more room to fit idiosyncratic noise directions. Thus output skips, like sparse connectivity, can be understood as architectural choices that alter the geometry on which gradient descent operates.

6. Discussion

This work shows that the architecture of a denoiser matters because it determines the geometry on which gradient descent operates: sparse connectivity at the input restricts each neuron to patches, making memorization harder; output skips such as clean-data prediction keep the internal representations lower-dimensional. Together, these findings suggest a unified geometric principle for understanding architectural inductive bias in generative models: the crucial variable is not just what functions can be represented, but what geometry the optimizer sees during training.

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).

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

An, J., Wang, D., Guo, P., Luo, J., and Schwing, A. On inductive biases that enable generalization in diffusion transformers. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=lE2cD7C9fk.

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

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., and Zhu, J. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22669–22679, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Bao_All_Are_Worth_Words_A_ViT_Backbone_for_Diffusion_Models_CVPR_2023_paper.html.

Beyer, L., Izmailov, P., Kolesnikov, A., Caron, M., Kornblith, S., Zhai, X., Minderer, M., Tschannen, M., Alabdulmohsin, I., and Pavetic, F. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14496–14506, 2023.

Bonnaire, T., Urfin, R., Biroli, G., and Mezard, M. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=BSZqpqgqM0.

Brutzkus, A., Globerson, A., Malach, E., Netser, A. R., and Shalev-Schwartz, S. Efficient learning of cnns using patch based features. In *International Conference on Machine Learning*, pp. 2336–2356. PMLR, 2022.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Chen, C. and collaborators. Exploring local memorization in diffusion models via bright ending attention. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.21665. Accepted spotlight; author list in source should be checked against final proceedings if needed.

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

Cohen, J., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Ding, L., Drusvyatskiy, D., Fazel, M., and Harchaoui, Z. Flat minima generalize for low-rank matrix recovery. *Information and Inference: A Journal of the IMA*, 13(2): iaae009, 2024.

Ding, X., Zhang, X., Zhou, Y., Han, J., Ding, G., and Sun, J. 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)*, pp. 11963–11975, 2022.

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

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018b.

Hang, T., Gu, S., Li, C., Bao, J., Chen, D., Hu, H., Geng, X., and Guo, B. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pp. 7441–7451, 2023. URL https://arxiv.org/abs/2303.09556.

Hatamizadeh, A., Song, J., Liu, G., Kautz, J., and Vahdat, A. Diffit: Diffusion vision transformers for image generation. In *European Conference on Computer Vision*, 2024. URL https://openreview.net/forum?id=uAKk0I3xxm.

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

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

Hintersdorf, D., Struppek, L., Kersting, K., Dziedzic, A., and Boenisch, F. Finding nemo: Localizing neurons responsible for memorization in diffusion models. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=YAEKMFZyJm.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

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

Kamb, M. and Ganguli, S. An analytic theory of creativity in convolutional diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28795–28831, 2025. URL https://proceedings.mlr.press/v267/kamb25a.html.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022. URL https://arxiv.org/abs/2206.00364.

Lahoti, A., Karp, S., Winston, E., Singh, A., and Li, Y. 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.

Li, T. and He, K. Back to basics: Let denoising generative models denoise, 2025. arXiv preprint arXiv:2511.13720.

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

Liang, T., Qiao, D., Wang, Y.-X., and Parhi, R. 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.

Liang, T., Cloninger, A., Parhi, R., and Wang, Y.-X. Generalization below the edge of stability: The role of data geometry. In *International Conference on Learning Representations (ICLR)*, 2026.

Lin, S., Liu, B., Li, J., and Yang, X. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5404–5411, 2024. URL https://arxiv.org/abs/2305.08891.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Liu, H., Chen, M., Zhao, T., and Liao, W. Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. In *International Conference on Machine Learning*, pp. 6770–6780. PMLR, 2021a.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021b.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11966–11976, 2022.

Ma, N., Goldstein, M., Albergo, M. S., Boffi, N. M., Vanden-Eijnden, E., and Xie, S. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vi-*

sion, 2024. URL https://arxiv.org/abs/2401.08740.

Mao, T., Shi, Z., and Zhou, D.-X. Theory of deep convolutional neural networks iii: Approximating radial functions. *Neural Networks*, 144:778–790, 2021.

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

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

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

Nacson, M. S., Mulayoff, R., Ongie, G., Michaeli, T., and Soudry, D. The implicit bias of minima stability in multivariate shallow ReLU networks. In *International Conference on Learning Representations*, 2023.

Nguyen, D.-K., Assran, M., Jain, U., Oswald, M. R., Snoek, C. G. M., and Chen, X. An image is worth more than 16x16 patches: Exploring transformers on individual pixels. *arXiv preprint arXiv:2406.09415*, 2024.

Niedoba, M., Zwartsenberg, B., Murphy, K. P., and Wood, F. Towards a mechanistic explanation of diffusion model generalization. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Hrp6jRIKdX.

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

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

Paulin, M., Mairal, J., Douze, M., Harchaoui, Z., Perronnin, F., and Schmid, C. Convolutional patch representations for image retrieval: an unsupervised approach. *International Journal of Computer Vision*, 121(1):149–168, 2017.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://arxiv.org/abs/2212.09748.

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

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

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

Qiao, D., Zhang, K., Singh, E., Soudry, D., and Wang, Y.-X. Stable minima cannot overfit in univariate ReLU networks: Generalization by large step sizes. In *Advances in Neural Information Processing Systems*, volume 37, pp. 94163–94208, 2024.

Ren, J., Li, Y., Zeng, S., Xu, H., Lyu, L., Xing, Y., and Tang, J. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2403.11052.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://arxiv.org/abs/2112.10752.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Shi, Z., Fang, Z., and Cao, Y. Approximation and estimation capability of vision transformers for hierarchical compositional models. *Applied and Computational Harmonic Analysis*, pp. 101849, 2025.

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

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12438–12448, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92c3b916311a5517d9290576e3ea37ad-Abstract.html.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*,

2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *International Conference on Machine Learning*, 2023. URL https://openreview.net/forum?id=FmqFfMTNnv.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19 (70):1–57, 2018.

Thiry, L., Arbel, M., Belilovsky, E., and Oyallon, E. The unreasonable effectiveness of patches in deep convolutional kernels methods. In *International Conference on Learning Representations*, 2021.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. 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*, pp. 10347–10357. PMLR, 2021.

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

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

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

Wang, F., Yang, T., Yu, Y., Ren, S., Wei, G., Wang, A., Shao, W., Zhou, Y., Yuille, A., and Xie, C. Adventurer: Optimizing vision mamba architecture designs for efficiency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30157–30166, June 2025a.

Wang, F., Yu, Y., Wei, G., Shao, W., Zhou, Y., Yuille, A., and Xie, C. 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*, pp. 65278–65290. PMLR, 2025b.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

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

Wen, Y., Liu, Y., Chen, C., and Lyu, L. Detecting, explaining, and mitigating memorization in diffusion models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=84n3UwkH7b.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., and Zhang, L. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31, 2021.

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

Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., and Girshick, R. Early convolutions help transformers see better. In *Advances in Neural Information Processing Systems*, volume 34, pp. 30392–30400, 2021.

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. W. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, 2020.

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

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E. H., Feng, J., and Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 558–567, 2021.

Zhang, Z., Zhang, K., Chen, M., Takeda, Y., Wang, M., Zhao, T., and Wang, Y.-X. Nonparametric classification on low dimensional manifolds using overparameterized convolutional residual networks. *Advances in Neural Information Processing Systems*, 37:65738–65764, 2024.

Zhang, Z., Li, X., Li, X., Shi, L., Wu, M., Tao, M., and Qu, Q. Generalization of diffusion models arises with a balanced representation space. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=57THeGgNAN.

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

A. More 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 & 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 & Kolter (2022) identify patch extraction, rather than attention, as the critical component, further reinforcing the strength of sparsely connected architectures. These results, however, either assume the labeling function satisfies a compositional structure or operate in settings with explicit regularization. Our work requires no assumptions on the labeling function and operates in the overparameterized regime.

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 & 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 & 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. Lastly, the significance of the input data distribution, rather than the labeling function, was not discovered in prior work.

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.

Edge of stability, minima stability, and generalization by large stepsizes. Our approach builds upon a recent line of work that studies the set of solutions that gradient descent can visit (or converge to) via either Edge-of-stability observation or the minima stability theory (Ding et al., 2024; Mulayoff et al., 2021; Nacson et al., 2023; Wu & Su, 2023; Qiao et al., 2024; Liang et al., 2025; 2026). These approaches enable formal analysis of the generalization properties without having to analyze gradient dynamics. To the best of our knowledge, they all focused on feedforward neural networks, and we are the first to study CNNs and the impact of model architecture choices in the implicit regularization of large stepsizes.

B. Broader (Informal) Discussion on Network Architecture Design: Bridge 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

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

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

$$\nabla_{\mathbf{W}}\mathcal{L}(\boldsymbol{\theta}) \;=\; \mathbf{G}(\boldsymbol{\theta})^\mathsf{T}\mathbf{X}. \tag{9}$$

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 (9) 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 9a 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 (9): gate statistics induced by $\mathbf{S}(\boldsymbol{\theta})$ and geometric moments of the patch cloud encoded by $\mathbf{X}$. Theorem 4.3 can therefore be viewed as a static stability proxy for the backpropagation geometry in (9). It does not say that stability alone explains generalization. Indeed, Theorem 4.4 shows that stable solutions may still memorize when the patch cloud admits isolating half-spaces. If corresponding backpropagation direction may enable a filter move to a location such that it only activates on a rare, nearly isolated patch, then the network may fit labels while paying little stability cost. 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 (9) suggests a second interpretation. Patchification also chooses the first local geometry seen by training. A 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

Table 1. 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)

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 1 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 $\nabla_{\mathbf{W}}\mathcal{L} = \mathbf{G}^{\mathsf{T}}\mathbf{X}$: 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 (9), 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

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	$96 \to 192 \to 384 \to 768$

operator at layer ℓ again has the schematic form

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

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

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

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

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

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.

C. Functional Analysis of Shallow ReLU Networks

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

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

Definition C.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

$$\|f_{\boldsymbol{\theta}}\|_{\mathrm{path}} := \sum_{k=1}^{K} |v_k|\, \|\boldsymbol{w}_k\|_2 \,. \tag{10}$$

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

$$f(\boldsymbol{x}) = \sum_{k=1}^{K'} a_k\, \phi(\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{x} - t_k) + \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + c_0. \tag{11}$$

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

$$\mathcal{F}_{\mathrm{fin}} := \bigcup_{K \geq 1} \left\{ \sum_{k=1}^{K} a_k\, \phi(\boldsymbol{u}_k^{\mathsf{T}}\cdot - t_k) + \boldsymbol{c}^{\mathsf{T}}(\cdot) + c_0 \right\}, \tag{12}$$

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

$$\|f\|_{\mathrm{path,min}} := \inf\left\{\|f_{\boldsymbol{\theta}}\|_{\mathrm{path}} :\ f_{\boldsymbol{\theta}} \equiv f \text{ of the form (11)}\right\}.$$

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* (12). Concretely, let ν be a finite signed Radon measure on $\mathbb{S}^{d-1} \times [-R, R]$ and consider

$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}\times[-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x} - t)\ \mathrm{d}\nu(\boldsymbol{u}, t) + c^{\mathsf{T}}\boldsymbol{x} + c_0. \tag{13}$$

Every finite network of the form (11) 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, (13) should be viewed as a *width-agnostic relaxation* aligned with (10), rather than as an assumption of an infinite-width limit.

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

$$|f|_{\mathrm{V}} := \inf\left\{\|\nu\|_{\mathcal{M}} : \ f \text{ admits (11) for some } (\nu, c, c_0)\right\}, \tag{14}$$

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

$$\mathrm{V}_C(\Omega) := \left\{ f\colon \Omega \to \mathbb{R} \mid f = \int_{\mathbb{S}^{d-1}\times[-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x} - t)\,\mathrm{d}\nu(\boldsymbol{u}, t) + \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + b,\ |f|_{\mathrm{V}} \le C \right\}. \tag{15}$$

In particular, identifying (11) with the atomic measure $\nu = \sum_k a_k \delta_{(\boldsymbol{u}_k, t_k)}$ yields

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

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

$$\|f\|_{\mathrm{path,min}} = |f|_{\mathrm{V}}. \tag{16}$$

Thus, (14) provides a *nonparametric* analogue of the path-norm: it encodes the same complexity notion while *not committing to a fixed width* K.

Remark C.3 ("Arbitrary width" $\neq$ "infinite width")**.** All statements here are about $\mathcal{F}_{\mathrm{fin}}$ in (12), namely networks of *finite* (but unconstrained) width. The integral representation (13) 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 $\|f\|_{\mathrm{path,min}}$.

C.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 C.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:

$$N(t, A, \rho_X) := \min\left\{ N \in \mathbb{N} : \exists\, x_1, \dots, x_N \in X \text{ s.t. } A \subset \bigcup_{i=1}^{N} \mathbb{B}(x_i, t) \right\}, \tag{17}$$

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

$$H_t(A)_X := \log N(t, A, \rho_X). \tag{18}$$

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

$$\log N(t, \mathrm{V}_C(\mathbb{B}_R^d), \|\cdot\|_\infty) \lessapprox_d \left(\frac{C}{t}\right)^{\frac{2d}{d+3}}. \tag{19}$$

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

C.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 C.5.

Proposition C.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

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

Then

$$\mathcal{G}_n(\mathcal{F}) \;\lesssim\; \frac{16}{\sqrt{n}} \int_0^{\operatorname{diam}(\mathcal{F},\|\cdot\|_n)} \sqrt{\log N\big(t,\ \mathcal{F},\ \|\cdot\|_n\big)}\ \mathrm{d}t, \tag{20}$$

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* $1-\delta$*,*

$$\widehat{\mathcal{G}}_n(\mathcal{F}) \;\le\; \mathcal{G}_n(\mathcal{F}) + \operatorname{diam}(\mathcal{F}, \|\cdot\|_n)\,\frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} \qquad (\delta > 0). \tag{21}$$

Lemma C.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 $1-\delta$,*

$$\sup_{f\in\mathcal{F}_{M,C}} \big|R(f) - \widehat{R}_{\mathcal{D}}(f)\big| \;\lesssim_d\; C^{\frac{d}{d+3}}\, M^{\frac{d+6}{d+3}}\, n^{-\frac{1}{2}} \;+\; M^2\left(\frac{\log(1/\delta)}{n}\right)^{\frac{1}{2}}. \tag{22}$$

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 $\{\boldsymbol{x}_i\}_{i=1}^n$,

$$\|\ell_f - \ell_g\|_n \le 4M\,\|f - g\|_n \le 4M\,\|f - g\|_\infty, \tag{23}$$

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 $1-\delta$,

$$\sup_{f\in\mathcal{F}_{M,C}} \big|R(f) - \widehat{R}_{\mathcal{D}}(f)\big| \;\lesssim\; \widehat{\mathcal{G}}_n(\mathcal{L}_{M,C}) \;+\; M^2\left(\frac{\log(1/\delta)}{n}\right)^{\frac{1}{2}}. \tag{24}$$

It remains to bound $\widehat{\mathcal{G}}_n(\mathcal{L}_{M,C})$ by chaining. By (23), 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 C.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 C.6 with $\mathcal{G} = \mathcal{L}_{M,C}$ and $\operatorname{diam}(\mathcal{L}_{M,C}, \|\cdot\|_n) \le 8M^2$ yields

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

Combining with (24) proves (22). □

D. Proof of Theorem 4.2

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

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) \;=\; \sum_{k=1}^{K} \frac{1}{J} \sum_{j=1}^{J} v_k \, \phi\big(\iota_j(\boldsymbol{w}_k)^{\mathsf{T}} \boldsymbol{x} - b_k\big) + \beta. \tag{25}$$

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

$$\|f_{\boldsymbol{\theta}}\|_{\mathrm{path}} = \frac{1}{J} \sum_{k=1}^{K} \sum_{j=1}^{J} |v_k| \, \|\iota_j(\boldsymbol{w}_k)\|_2 = \sum_{k=1}^{K} |v_k| \, \|\boldsymbol{w}_k\|_2 \,. \tag{26}$$

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

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L} = \underbrace{\frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i) \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)^{\mathsf{T}}}_{\boldsymbol{T}_{\mathcal{D}}} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i) \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)}_{\boldsymbol{R}_{\mathcal{D}}} \,. \tag{27}$$

Definition D.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 draw 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

$$g_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) = \min\Big\{\tilde{g}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t),\; \tilde{g}_{\mathcal{D},\mathcal{S}}(-\boldsymbol{u}, -t)\Big\}. \tag{28}$$

where

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

Proposition D.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*

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

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,

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

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

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

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

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

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

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\text{)} \\
&= 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]\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 D.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 $\|\boldsymbol{x}\|_2 \le R$, 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

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

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

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

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

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

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:

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

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,

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

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 D.4 (Restate Theorem 4.2)**.** *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 D.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 $|\boldsymbol{\omega}^\top \nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\boldsymbol{\omega}|$, as detailed in Lemma D.3:

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

Thus, we have

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

Finally, we plug (43) into (45)

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

□

D.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 D.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

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

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

$$p_j(\boldsymbol{u},t) := \mathbb{P}\big(\boldsymbol{u}^\top\pi_j(\boldsymbol{X}) > t\big), \quad r_j(\boldsymbol{u},t) := \mathbb{E}\big[\phi(\boldsymbol{u}^\top\pi_j(\boldsymbol{X}) - t)\big], \quad \boldsymbol{a}_j(\boldsymbol{u},t) := \mathbb{E}\big[\pi_j(\boldsymbol{X})\,\mathbb{1}\{\boldsymbol{u}^\top\pi_j(\boldsymbol{X}) > t\}\big].$$

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

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

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

and

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

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 & 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 D.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 $1-\delta$,*

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

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

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

By the VC-subgraph entropy bound and maximal inequality (van der Vaart & 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 D.6, for every fixed j and $\eta \in (0,1)$, with probability at least $1-\eta$,

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

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

For the vector moment, use duality:

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

The last supremum is over the class $\mathcal{A}$ from Lemma D.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 $(\boldsymbol{u},t) \in \mathbb{S}^{m-1} \times [-1,1]$,

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

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. □

D.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 D.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 one–coordinate 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 D.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 D.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 D.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 D.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 $t \downarrow 0$. □

E. Proof of Theorem 4.3

Theorem E.1 (Detailed version of Theorem 4.3)**.** *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 $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{SCN}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$ and $\|f_{\boldsymbol{\theta}}\|_\infty \le M$, then, with probability $\ge 1 - 2\delta$, we have that for the plug-in risk estimator $\widehat{\mathcal{R}}_{\mathcal{D}}(f) := \frac{1}{n}\sum_{i=1}^n \big(f(\boldsymbol{x}_i) - y_i\big)^2$,*

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

for any $\varepsilon \in (0,1)$ such that $\varepsilon^d \gtrsim d^2\sqrt{\frac{m+\log(J/\delta)}{n}}$[1]*. 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

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

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

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

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

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

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

Therefore we have

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

According to (47) and the definition $g = \min\{\tilde{g}(\boldsymbol{u},t), \tilde{g}(-\boldsymbol{u},-t)\}$, 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 $\mathbb{P}(\boldsymbol{u}^\top \boldsymbol{X}^{(S_j)} > t)$ and $\mathbb{E}[\phi(\boldsymbol{u}^\top \boldsymbol{X}^{(S_j)} - t)]$ 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 D.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 $\boldsymbol{u} \in \mathbb{R}^m$,

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

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 C.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 D.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 $\varepsilon \in (0,1)$,

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

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 D.10 (see (62)) gives the explicit lower bound

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

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

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

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 D.10 satisfies

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

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 E.2. *Under the same conditions as Theorem 4.3, assume $d > 3$ and $1 \le m < \frac{d(d-3)}{d+3}$. Let*

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

Define the optimal choice

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

and choose the feasible/truncated value

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

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

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

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

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

where

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

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

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

where

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

Plugging ε^* into (91) yields

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

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

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

□

F. Proof of Theorem 4.4

Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset with $\boldsymbol{x}_i \in \mathbb{R}^d$, and let $\mathcal{S} = \{\pi_j : \mathbb{R}^d \to \mathbb{R}^m\}_{j=1}^J$ be a collection of coordinate projections (patch extractors). For each $j \in [J]$, abbreviate the extracted patch by $\boldsymbol{x}^{(S_j)} := \pi_j(\boldsymbol{x})$, and for each sample write $\boldsymbol{x}_i^{(S_j)} := \pi_j(\boldsymbol{x}_i)$.

Assume labels are uniformly bounded: $|y_i| \le D$ for all $i \in [n]$. Define $I_{\neq 0} := \{i \in [n] : y_i \neq 0\}$.

Consider width-K two-layer sparsely connected ReLU models with Global Average Pooling (GAP),

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^K \frac{v_k}{J} \sum_{j=1}^J \phi\Big(\boldsymbol{w}_k^\top \boldsymbol{x}^{(S_j)} - b_k\Big) + \beta, \qquad \phi(t) = \max\{t, 0\}, \tag{102}$$

where $\boldsymbol{w}_k \in \mathbb{R}^m$, $b_k \in \mathbb{R}$ are the shared filter weights and bias, $v_k \in \mathbb{R}$ is the output weight, and $\beta \in \mathbb{R}$. We write $\boldsymbol{\theta} = \{(\boldsymbol{w}_k, b_k, v_k)\}_{k=1}^K \cup \{\beta\}$.

Theorem F.1 (Flat interpolation with width $\le n$)**.** *Assume that $\big\|\boldsymbol{x}_i^{(S_j)}\big\|_2 \le 1$ for all $i \in [n]$, $j \in [J]$, and there exists a map $\tau : I_{\neq 0} \to [J]$ that assigns $\boldsymbol{p}_i := \boldsymbol{x}_i^{(S_{\tau(i)})}$ such that $\|\boldsymbol{p}_i\|_2 = 1$ and $\boldsymbol{p}_i \neq \boldsymbol{x}_\ell^{(S_j)}$ for all $(\ell, j) \neq (i, \tau(i))$. There exists a width $K \le n$ network of the form (1) that interpolates the dataset and whose Hessian operator norm satisfies*

$$\lambda_{\max}\big(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}\big) \le 1 + \frac{D^2 + 2/J^2}{n}. \tag{103}$$

Construction F.2. Let $K \coloneqq |I_{\neq 0}| \le n$ and index the hidden units by $k \in I_{\neq 0}$. For each $k \in I_{\neq 0}$ define the anchor patch $\boldsymbol{p}_k \coloneqq \boldsymbol{x}_k^{(S_{\tau(k)})}$ and set

$$\rho_k \coloneqq \max_{(\ell,j)\neq(k,\tau(k))} \big(\boldsymbol{x}_\ell^{(S_j)}\big)^\top \boldsymbol{p}_k \;<\; 1, \qquad b_k \in (\rho_k, 1), \qquad \boldsymbol{w}_k \coloneqq \boldsymbol{p}_k. \tag{104}$$

Set the output bias and output weights

$$\beta \coloneqq 0, \qquad v_k \coloneqq \frac{J\, y_k}{1 - b_k}, \qquad k \in I_{\neq 0}. \tag{105}$$

(Justification of $\rho_k < 1$.) According to the assumption, for any $(\ell, j) \neq (k, \tau(k))$,

$$\big(\boldsymbol{x}_\ell^{(S_j)}\big)^\top \boldsymbol{p}_k \le \big\|\boldsymbol{x}_\ell^{(S_j)}\big\|_2 \, \|\boldsymbol{p}_k\|_2 \le 1.$$

If equality held then necessarily $\|\boldsymbol{x}_\ell^{(S_j)}\|_2 = 1$ and $\boldsymbol{x}_\ell^{(S_j)} = \boldsymbol{p}_k$, contradicting to the assumption. Hence $\rho_k < 1$.

By (104), for any sample index i and any patch index j,

$$\boldsymbol{w}_k^\top \boldsymbol{x}_i^{(S_j)} - b_k = \begin{cases} 1 - b_k \;>\; 0, & (i,j) = (k, \tau(k)), \\ \le \rho_k - b_k \;<\; 0, & (i,j) \neq (k, \tau(k)). \end{cases} \tag{106}$$

Thus neuron k is activated on exactly one patch in the entire collection $\{\boldsymbol{x}_i^{(S_j)}\}_{i,j}$, namely $\boldsymbol{x}_k^{(S_{\tau(k)})}$, and inactivated on all other patches. Using (106) and (105), at $\boldsymbol{x}_k$ we have

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}_k) = \frac{v_k}{J}\, \phi(1 - b_k) = \frac{v_k}{J}\,(1 - b_k) = y_k, \qquad k \in I_{\neq 0},$$

and for $i \notin I_{\neq 0}$ all constructed units are inactivated on all patches of $\boldsymbol{x}_i$, hence $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \beta = 0 = y_i$. Therefore, $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = y_i$ for all $i \in [n]$.

For each constructed unit, define

$$\tilde{v}_k \coloneqq \operatorname{sign}(v_k) \in \{\pm 1\}, \qquad \tilde{\boldsymbol{w}}_k \coloneqq |v_k|\, \boldsymbol{w}_k, \qquad \tilde{b}_k \coloneqq |v_k|\, b_k. \tag{107}$$

Then for any input $\boldsymbol{x}$,

$$\frac{\tilde{v}_k}{J} \sum_{j=1}^{J} \phi\Big(\tilde{\boldsymbol{w}}_k^\top \boldsymbol{x}^{(S_j)} - \tilde{b}_k\Big) = \frac{v_k}{J} \sum_{j=1}^{J} \phi\Big(\boldsymbol{w}_k^\top \boldsymbol{x}^{(S_j)} - b_k\Big), \tag{108}$$

so interpolation is preserved. Moreover, the activation pattern on the full patch collection is unchanged because (106) has strict inequalities and $|v_k| > 0$. At the unique activated patch $(k, \tau(k))$, the (post-rescaling) pre-activation is

$$\tilde{z}_k \coloneqq \tilde{\boldsymbol{w}}_k^\top \boldsymbol{x}_k^{(S_{\tau(k)})} - \tilde{b}_k = |v_k|\,(1 - b_k) = J|y_k| \;>\; 0, \qquad |\tilde{v}_k| = 1. \tag{109}$$

In what follows we work with the reparameterized network and drop tildes for readability, implicitly assuming $|v_k| = 1$ for all $k \in I_{\neq 0}$ and

$$z_k \coloneqq \boldsymbol{w}_k^\top \boldsymbol{x}_k^{(S_{\tau(k)})} - b_k = J|y_k|. \tag{110}$$

Proposition F.3. *Let $\boldsymbol{\theta}$ be the model in Construction F.2 after the reparameterization (107). Then*

$$\lambda_{\max}\big(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}\big) \le 1 + \frac{D^2 + 2/J^2}{n}.$$

Proof. By direct computation, the Hessian $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}$ is given by

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)^\top + \frac{1}{n} \sum_{i=1}^{n} \big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\big) \nabla^2_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i). \tag{111}$$

Since the model interpolates $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = y_i$ for all i, we have

$$\nabla^2_{\boldsymbol{\theta}}\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)^{\mathsf{T}}. \tag{112}$$

Denote the tangent features matrix by

$$\Phi = \left[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_1), \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_2), \cdots, \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\right]. \tag{113}$$

Then (112) can be written as $\nabla^2_{\boldsymbol{\theta}}\mathcal{L} = \Phi\Phi^{\mathsf{T}}/n$, and thus

$$\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) = \max_{\boldsymbol{\gamma}\in\mathbb{S}^{(m+2)K}} \frac{1}{n}\|\Phi^{\mathsf{T}}\boldsymbol{\gamma}\|^2 = \max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \frac{1}{n}\|\Phi\boldsymbol{u}\|^2. \tag{114}$$

For the gate $m_k^{(S_j)}(\boldsymbol{x}) \coloneqq \mathbb{1}\left\{\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x}^{(S_j)} > b_k\right\}$, let $m_{k,i}^{(S_j)} \coloneqq m_k^{(S_j)}(\boldsymbol{x}_i)$. From direct computation,

$$\begin{aligned}
\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\partial v_k} &= \frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)} \cdot \left(\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x}_i^{(S_j)} - b_k\right), & \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\partial \boldsymbol{w}_k} &= \frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)} \cdot v_k \cdot \boldsymbol{x}_i^{(S_j)}, \\
\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\partial b_k} &= -\frac{1}{J}\sum_{j=1}^{J} m_{k,i}^{(S_j)} \cdot v_k, & \frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)}{\partial \beta} &= 1.
\end{aligned} \tag{115}$$

By the one-to-one activation property (106), each sample $\boldsymbol{x}_i$ with $i \in I_{\neq 0}$ activates exactly one unit (the unit $k = i$) on exactly one patch $S_{\tau(i)}$, and samples with $i \notin I_{\neq 0}$ activate none. Hence the sample-wise gradient $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ has support only on the parameter triplet $(\boldsymbol{w}_i, b_i, v_i, \beta)$ when $i \in I_{\neq 0}$, and is zero on $(\boldsymbol{w}_k, b_k, v_k)$ for all k when $i \notin I_{\neq 0}$. Writing the nonzero gradient block explicitly (recall $|v_i| = 1$ after reparameterization and (110)),

$$\begin{aligned}
\nabla_{(\boldsymbol{w}_i, b_i, v_i, \beta)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) &= \begin{pmatrix} \nabla_{(\boldsymbol{w}_i, b_i, v_i)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) \\ 1 \end{pmatrix}, \\
\nabla_{(\boldsymbol{w}_i, b_i, v_i)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) &= \begin{cases} \begin{pmatrix} \frac{v_i}{J}\boldsymbol{x}_i^{(S_{\tau(i)})} \\ -\frac{v_i}{J} \\ \frac{1}{J}\left(\boldsymbol{w}_i^{\mathsf{T}}\boldsymbol{x}_i^{(S_{\tau(i)})} - b_i\right) \end{pmatrix} = \begin{pmatrix} \frac{v_i}{J}\boldsymbol{x}_i^{(S_{\tau(i)})} \\ -\frac{v_i}{J} \\ |y_i| \end{pmatrix}, & (i \in I_{\neq 0}), \\ \mathbf{0}, & (i \notin I_{\neq 0}). \end{cases}
\end{aligned} \tag{116}$$

Let $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathbb{S}^{n-1}$ and plug (116) into (114). As in the fully connected case, after a row permutation (grouping neuron parameters) we obtain

$$\begin{aligned}
\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) &= \max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \frac{1}{n}\|\Phi\boldsymbol{u}\|^2 \\
&= \frac{1}{n}\max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \sum_{i=1}^{n} u_i^2 \left\|\nabla_{(\boldsymbol{w}_i, b_i, v_i)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right\|_2^2 + \left(\sum_{i=1}^{n} u_i\right)^2 && (117) \\
&= \frac{1}{n}\max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \sum_{i\in I_{\neq 0}} u_i^2 \left(\frac{1}{J^2}\left\|\boldsymbol{x}_i^{(S_{\tau(i)})}\right\|_2^2 + \frac{1}{J^2} + y_i^2\right) + \left(\sum_{i=1}^{n} u_i\right)^2. && (118)
\end{aligned}$$

According to the assumption, $\|\boldsymbol{x}_i^{(S_{\tau(i)})}\|_2 = 1$ for all $i \in I_{\neq 0}$, and by bounded labels $y_i^2 \le D^2$ for all i. Thus

$$\begin{aligned}
\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) &\le \frac{1}{n}\left(\max_{i\in[n]}\left(\frac{1}{J^2}\left\|\boldsymbol{x}_i^{(S_{\tau(i)})}\right\|_2^2 + \frac{1}{J^2} + y_i^2\right) + \max_{\boldsymbol{u}\in\mathbb{S}^{n-1}}\left(\sum_{i=1}^{n} u_i\right)^2\right) \\
&\le \frac{1}{n}\left(\frac{2}{J^2} + D^2 + n\right) = 1 + \frac{D^2 + 2/J^2}{n}.
\end{aligned}$$

If we remove the output bias term β from the parameters, then the last term $(\sum_i u_i)^2$ in (117) is removed. □

G. 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.

G.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 G.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

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) \;=\; \sum_{k=1}^{K}\sum_{j=1}^{J} v_{k,j}\,\phi\big(\boldsymbol{w}_{k,j}^{\mathsf{T}}\pi_j(\boldsymbol{x}) - b_{k,j}\big) \;+\; \beta, \tag{119}$$

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 (119) 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

$$g_{\mathcal{D},\mathcal{S}}^{(j)}(\boldsymbol{u}, t) \;:=\; \mathbb{E}\Big[\phi\big(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{D}}^{(j)} - t\big)\Big]\,\sqrt{\mathbb{P}\big(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{D}}^{(j)} > t\big)^2 + \Big\|\mathbb{E}\Big[\boldsymbol{X}_{\mathcal{D}}^{(j)}\,\mathbb{1}\Big\{\boldsymbol{u}^{\mathsf{T}}\boldsymbol{X}_{\mathcal{D}}^{(j)} > t\Big\}\Big]\Big\|_2^2}. \tag{120}$$

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

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

We establish the counterpart of Theorem 4.2 for the unshared architecture (119). 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 G.2 (Hessian residual bound for the unshared SCN)**.** *For the architecture* (119) *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 D.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 G.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*

$$\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}}) \;\ge\; 1 \;+\; 2\sum_{k=1}^{K}\sum_{j=1}^{J} |v_{k,j}|\;\|\boldsymbol{w}_{k,j}\|\;g_{\mathcal{D},\mathcal{S}}^{(j)}\!\left(\frac{\boldsymbol{w}_{k,j}}{\|\boldsymbol{w}_{k,j}\|},\,\frac{b_{k,j}}{\|\boldsymbol{w}_{k,j}\|}\right). \tag{121}$$

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

$$\frac{\partial f}{\partial \boldsymbol{w}_{k,j}}(\boldsymbol{x}_i) = v_{k,j}\, m_{k,j,i}\, \pi_j(\boldsymbol{x}_i), \qquad \frac{\partial f}{\partial b_{k,j}}(\boldsymbol{x}_i) = -v_{k,j}\, m_{k,j,i}, \tag{122}$$

$$\frac{\partial f}{\partial v_{k,j}}(\boldsymbol{x}_i) = \phi\big(\boldsymbol{w}_{k,j}^\top \pi_j(\boldsymbol{x}_i) - b_{k,j}\big), \qquad \frac{\partial f}{\partial \beta}(\boldsymbol{x}_i) = 1. \tag{123}$$

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))

$$\lambda_{\max}(\boldsymbol{T}_{\mathcal{D}}) = \frac{1}{n}\max_{\|\boldsymbol{u}\|=1} \|\Phi \boldsymbol{u}\|^2 \;\geq\; \frac{1}{n^2}\|\Phi\, \mathbf{1}\|^2 = \frac{1}{n^2}\Big\|\sum_{i=1}^n \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)\Big\|^2. \tag{124}$$

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

$$\frac{1}{n^2}\|\Phi\,\mathbf{1}\|^2 = 1 + \sum_{k=1}^K \sum_{j=1}^J \Bigg[\underbrace{v_{k,j}^2\left(\Big\|\frac{1}{n}\sum_i m_{k,j,i}\,\pi_j(\boldsymbol{x}_i)\Big\|^2 + \Big(\frac{1}{n}\sum_i m_{k,j,i}\Big)^2\right)}_{=:\; P_{k,j}} + \underbrace{\|\boldsymbol{w}_{k,j}\|_2^2\left(\frac{1}{n}\sum_i \phi\big(\boldsymbol{u}_{k,j}^\top \pi_j(\boldsymbol{x}_i) - t_{k,j}\big)\right)^2}_{=:\; Q_{k,j}}\Bigg]. \tag{125}$$

Here the "1" comes from $\left(\frac{1}{n}\sum_i \partial f/\partial\beta\right)^2 = 1$. 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 D.2):

$$P_{k,j} + Q_{k,j} \;\geq\; 2\,|v_{k,j}|\;\|\boldsymbol{w}_{k,j}\|_2 \underbrace{\left(\frac{1}{n}\sum_i \phi\big(\boldsymbol{u}_{k,j}^\top \pi_j(\boldsymbol{x}_i) - t_{k,j}\big)\right)\sqrt{\Big(\frac{1}{n}\sum_i m_{k,j,i}\Big)^2 + \Big\|\frac{1}{n}\sum_i m_{k,j,i}\,\pi_j(\boldsymbol{x}_i)\Big\|^2}}_{=\; g^{(j)}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}_{k,j},\, t_{k,j})}. \tag{126}$$

To verify the identification with (120), note that the empirical averages over $i \in [n]$ are precisely the expectation, probability, and conditional first moment under the uniform draw $\boldsymbol{X}_{\mathcal{D}}^{(j)} \sim \mathrm{Uniform}(\{\pi_j(\boldsymbol{x}_i)\}_{i=1}^n)$:

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

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

Summing (126) over (k,j) and combining with (124)–(125) yields (121). □

Remark G.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 D.2.

In the unshared architecture (119), 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 (125). This per-neuron decomposition makes the AM–GM step (126) 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 G.5 (BEoS constraint for the unshared SCN)**.** *For the architecture* (119)*, if* $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{\mathcal{SCN}}_{\mathrm{BEoS}}(\eta, \mathcal{D})$*, i.e.,* $\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) \le 2/\eta$*, then*

$$\sum_{k=1}^{K}\sum_{j=1}^{J} |v_{k,j}|\, \|\boldsymbol{w}_{k,j}\|\; g^{(j)}_{\mathcal{D},\mathcal{S}}\!\left(\frac{\boldsymbol{w}_{k,j}}{\|\boldsymbol{w}_{k,j}\|}, \frac{b_{k,j}}{\|\boldsymbol{w}_{k,j}\|}\right) \;\le\; \frac{1}{\eta} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{127}$$

Proof. Decompose the Hessian as $\nabla^2_{\boldsymbol{\theta}}\mathcal{L} = \boldsymbol{T}_{\mathcal{D}} + \boldsymbol{R}_{\mathcal{D}}$ (cf. (27)) 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))

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

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

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

Therefore,

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

Inserting the lower bound from Proposition G.3 and rearranging yields (127). □

G.3. Generalization analysis

We now derive the generalization guarantee for the unshared architecture (119), following the same interior/exterior decomposition as in Appendix E.

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

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

By a union bound and Proposition D.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 $\|\pi_j(\boldsymbol{x})\| \le 1-\varepsilon$ 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 D. By the same computation as in Proposition D.10 (see (59)), the population weight function at each location satisfies

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

with a constant $c_g(d) > 0$ depending only on d. The uniform deviation bound of Theorem D.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

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

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

$$g^{(j)}_{\mathcal{D},\mathcal{S}}(\boldsymbol{u}, t) \;\geq\; \tfrac{1}{2}\, c_g(d)\, \varepsilon^d \qquad \text{for all } j \in [J],\ \boldsymbol{u} \in \mathbb{S}^{m-1},\ |t| \leq 1-\varepsilon. \tag{128}$$

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 (128) into the BEoS constraint (127) and setting $A := \frac{1}{\eta} - \frac{1}{2} + 4M$ yields

$$\sum_{\substack{k,j:\\ |t_{k,j}| \leq 1-\varepsilon}} |v_{k,j}|\, \|\boldsymbol{w}_{k,j}\| \;\leq\; \frac{A}{\frac{1}{2} c_g(d)\, \varepsilon^d} \;\asymp\; A\, \varepsilon^{-d}. \tag{129}$$

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 (129)). The generalization gap decomposes as in (77)–(79):

- *Exterior contribution.* Since $\|f_{\boldsymbol{\theta}}\|_\infty \leq M$ and $|y| \leq D \leq M$, the exterior contributes at most $O(JM^2 \varepsilon^{(d-m)/2})$ (cf. (82)).
- *Interior contribution.* By Lemma C.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, $|\mathbb{P}(\mathsf{I}^{\mathrm{all}}_{\varepsilon}) - n_I/n| \lesssim \sqrt{\log(1/\delta)/n}$ 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 G.6 (Generalization for the unshared SCN)**.** *Assume $\boldsymbol{x} \sim \mathrm{Uniform}(\mathbb{S}^{d-1})$ and $|y| \leq D$. Let $M \geq D$, $\|f_{\boldsymbol{\theta}}\|_\infty \leq M$, and suppose $\boldsymbol{\theta}$ satisfies the BEoS condition $\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}) \leq 2/\eta$. Assume $d > 3$ and $1 \leq m < \frac{d(d-3)}{d+3}$. Then, with probability at least $1-2\delta$,*

$$\mathrm{GenGap}(f_{\boldsymbol{\theta}}; \mathcal{D}) \;\lessapprox_d\; JM^2\, \varepsilon^{\frac{d-m}{2}} + \big(A\, \varepsilon^{-d}\big)^{\frac{d}{d+3}} M^{\frac{d+6}{d+3}} n^{-1/2}, \tag{130}$$

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 D.11 and Hoeffding's inequality (see (82)). The interior bound uses the path-norm estimate (129) together with the metric entropy of variation spaces (Lemma C.7; see (90)). The concentration term $\big|\mathbb{P}(\mathsf{I}^{\mathrm{all}}_{\varepsilon}) - n_I/n\big|$ is controlled as in (84) and is of smaller order. □

Optimizing over ε as in Corollary E.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 4.3.

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.

H. The Role of Weight Sharing

The analysis in Section 4 shows that sparse connectivity strengthens implicit regularization by shifting the geometric input to gradient descent. But the SCN architecture bundles two distinct inductive biases together: sparse fan-in and weight sharing. Does sparse fan-in alone produce the effect, or is weight sharing essential?

To test this, we introduce a third architecture identical to the SCN except that the filters are no longer shared across locations. This *sparsely connected network without weight sharing* (SCN-noWS) has independent parameters per patch position:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K}\sum_{j=1}^{J} v_{k,j}\,\phi(\boldsymbol{w}_{k,j}^{\top}\pi_j(\boldsymbol{x}) - b_{k,j}) + \beta.$$

Under weight sharing, a filter is driven by patches from all locations. To memorize a sample, the filter must isolate a patch that is distinguishable from every other patch in the entire multiset. Without weight sharing, a location-specific neuron only needs to find a patch that stands out within its own location. In typical data such as images, recurring local structures make cross-location isolation far harder than within-location isolation. Thus, weight sharing raises the threshold for memorization, pushing the network toward features that are common across the patch multiset.

The following experiment tests this hypothesis in a controlled denoising setting. **Data.** We construct D-dimensional inputs composed of J equal-sized disjoint patches. Within the patch multiset, most patches are pure Gaussian noise, but a small fraction carry signal. Concretely, fix two signal centers $\boldsymbol{v}_+, \boldsymbol{v}_- \in \mathbb{S}^{m-1}$. For each sample, a label $y \in \{\boldsymbol{v}_+, \boldsymbol{v}_-\}$ is drawn uniformly, and a random location $j^\star$ is selected. The patch at $j^\star$ is set to y plus a small Gaussian perturbation; all other patches are independent Gaussian noise with matched total energy. This creates a sharp cluster structure: the vast majority of patches form a dense noise cluster around the origin, while two small signal clusters lie near $\boldsymbol{v}_+$ and $\boldsymbol{v}_-$. Spatially, the signal is extremely sparse (one informative patch per sample).

Setup. We train three two-layer ReLU networks of identical width and learning rate: a fully connected network (FCN), an SCN-noWS, and an SCN. The task is regression: predict the label from the noisy input. The training objective is standard denoising MSE, without explicit regularization.

Results. Figure 8 shows training and test loss. All three architectures fit the training data to near-zero loss. However, only the SCN generalizes to held-out samples; the FCN and SCN-noWS both overfit. This is exactly what the memorization-threshold argument predicts. In the SCN-noWS, a neuron at location j needs only to find a signal patch that is distinguishable from other patches at the *same* location; with high probability, the one signal patch in a training sample appears as a local outlier. The SCN, by contrast, forces each filter to respond to patches across all locations. A filter that fires on a signal patch at one location will also fire on the many noise patches elsewhere. This drives up its activation mass and the associated stability penalty, pushing the network away from sample-specific memorization and toward filters tuned to the recurring signal clusters.

This experiment reveals a deeper property of weight sharing in denoising settings. Because the added noise is independent across coordinates, patches at different locations are perturbed by independent noise realizations. A filter restricted to a single location cannot separate signal from noise using cross-location statistics; it is forced to rely on the local noise realization. Weight sharing lifts this restriction: a shared filter sees many independent noise realizations of the same underlying patch structure, and its aggregated gradient suppresses the noise while amplifying the consistent signal directions. Thus weight sharing acts as an implicit denoising mechanism at the level of gradient dynamics.

Remark H.1 (Weight sharing as a geometric form of translation invariance)**.** Translation invariance is usually invoked to explain parameter efficiency and equivariance. Our analysis suggests a complementary geometric perspective. By coupling each filter to all spatial locations, weight sharing effectively redefines the "unit of isolation" for memorization. A training sample can be memorized without sharing if it has *any* distinguishable patch at *any* location. With sharing, merely local distinctiveness is not enough; the patch must be globally distinctive across the entire multiset. For natural images, where

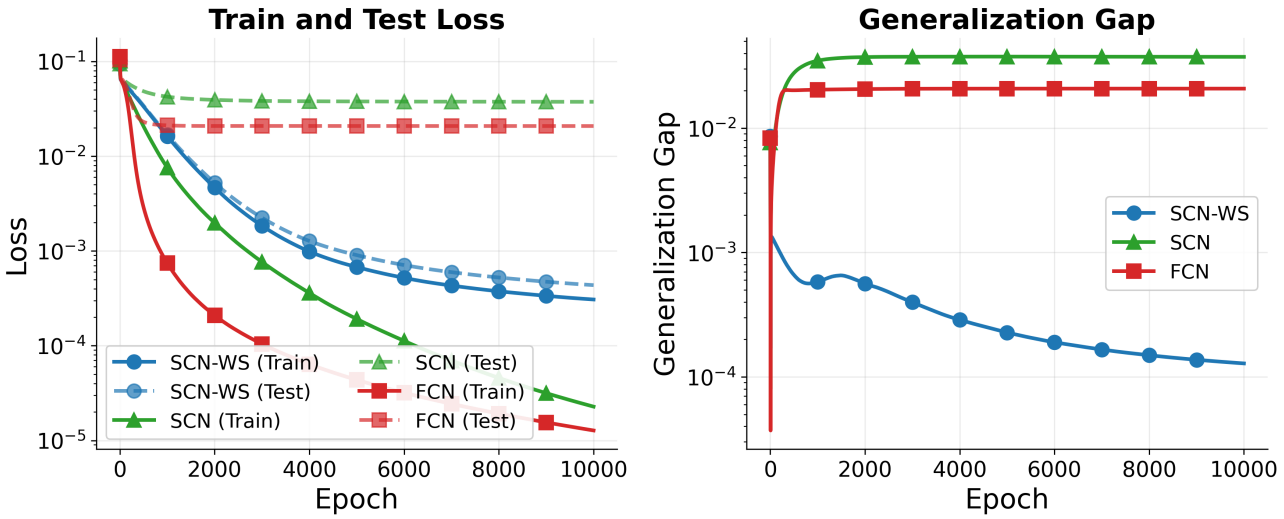
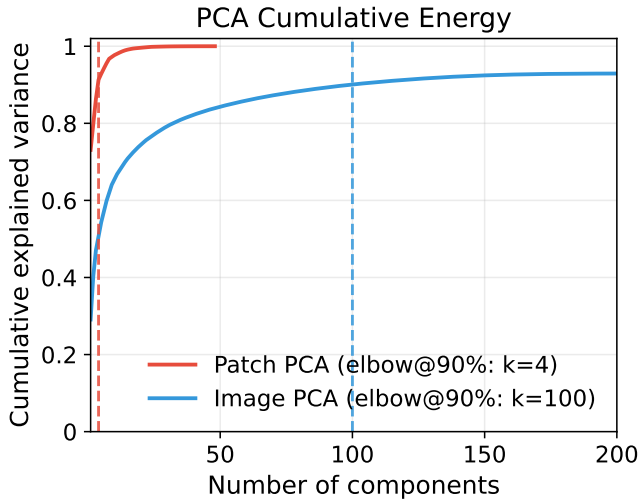


Figure 8. Training loss (solid) and test loss (dashed) for the three architectures on the clustered-patch denoising task. Only the SCN resists overfitting.

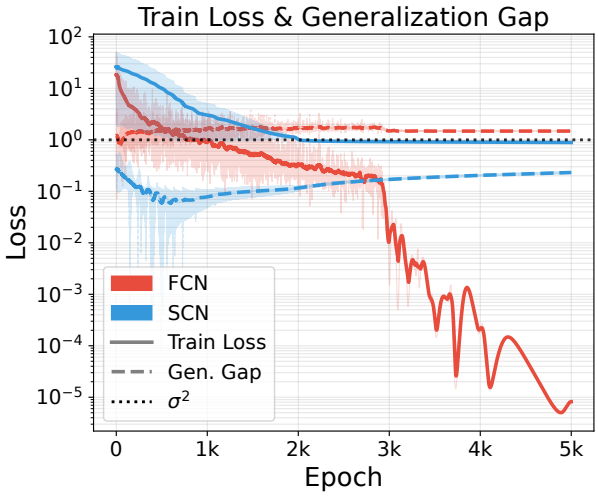
patches repeat across space, this condition is much harder to satisfy. Weight sharing thus funnels gradient descent away from sample-specific memorization and toward generalizable features, providing a form of implicit regularization rooted in the geometry of the patch multiset rather than in parameter counts.

I. Patch Geometry of Natural Images

We now examine which side of the dichotomy established in Figure 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. Figure 9a compares the patch cloud with the cloud of full images. The patch cloud is strongly concentrated: 90% of its variance lives in just 4 directions of $\mathbb{R}^{27}$, while the full images in $\mathbb{R}^{3072}$ need over 100 directions. Recalling the gradient view (3), 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 Figure 4. Moreover, Figure 9b indicates that SCN indeed benefit from the convolutional patch geometry.



(a) Patch vs Image.



(b) Regression on CIFAR-10.

Figure 9. **(a)** 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. **(b)** The SCN stabilizes near the noise floor; the FCN interpolates.

The takeaway message is that local receptive-field design can affect implicit regularization, not only feature extraction. Even under the same ambient data distribution, different receptive-field systems can induce very different patch geometries, and hence can change the balance between expressivity and regularization. In particular, the size of patch/local receptive field size in the hidden layers is a concrete perspective.

J. 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):

$$\mathrm{MSE}(f) = \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - f_{true}(\boldsymbol{x}_i))^2. \tag{131}$$

The population level of (131) is known as *excess risk*,

$$\mathrm{Excess}(f) \coloneqq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X}}}[(f(\boldsymbol{x}) - f_{true}(\boldsymbol{x}))^2], \tag{132}$$

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

$$\mathcal{R}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}}\left[(f(\boldsymbol{x}) - y)^2\right] = \mathrm{Excess}(f) + \sigma^2, \tag{133}$$

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

$$\mathrm{GenGap}(f, \mathcal{D}) = \left|\mathrm{Excess}(f) + \sigma^2 - \widehat{\mathcal{R}}_{\mathcal{D}}(f)\right|. \tag{134}$$

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.

J.1. Experimental details for the interpolation below the edge of stability

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 $f_{true} \in \mathbf{\Theta}^{\mathrm{SCN}}$ 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 4.3). (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

Excess(f) defined in Eq (132); (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 5) 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**: $d = 10$).

J.2. Experimental details for Figure 5

Similar setup and metrics as the spherical-data setup above in J.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 $\boldsymbol{x}^{(j)} \sim \mathrm{Uniform}(\mathbb{S}^{m-1})$, instead of the whole-vector sampling $\boldsymbol{x} \sim \mathrm{Uniform}(\mathbb{S}^{d-1})$. Second, we fix $m = 10$, $J = 8$ (so $d = 80$), $K_{\mathrm{true}} = 50$, $n = 512$, $\sigma = 1$, and train at width $K = 1024$ for 50,000 epochs at $\eta = 0.2$ with gradient clipping. Additional experiments with similar setup in Fig 10, 11 and Table 3.

J.3. Experimental details for Figure 9a

We evaluate four patchification schemes on CIFAR-10: (i) *Conv Patches*, 4×4 patches with stride 2 (baseline); (ii) *Random Patches*, 4×4 contiguous crops at random spatial locations (same patch content as Conv but irregular grid); (iii) *ShufflePixel*, conv-style patches taken after a global pixel permutation (negative control); and (iv) *Full Image Space*, the flattened 3072-D image. For each scheme we draw 2,000,000 reference and 50,000 query patches and report two geometry metrics: (a) the PCA *effective rank*, taken as the elbow of the cumulative explained variance at 90% (lower values indicate more structured distributions); and (b) the area under the half-space (Tukey) depth concentration curve $\Psi(T) = \Pr[\hat{d}(\boldsymbol{z}) \geq T]$ for $T \in [0, 0.5]$ (higher values indicate distributions that are harder to shatter), computed via the random-projection approximation 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.

J.4. Experimental details for Figure 9b

We repeat the above setup in J.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; $L = 900$, $m = 27$). Second, targets are generated by a *frozen* ground-truth SCN f_{true} of width $K_{\mathrm{true}} = 20$ with the same patchification, $y_i = f_{\mathrm{true}}(\boldsymbol{x}_i) + \xi_i$, $\xi_i \sim \mathcal{N}(0, 1)$; 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 $n_{\mathrm{train}} = 1024$ images for 5000 epochs with $\eta = 0.2$ and gradient clipping. Both networks are overparameterized in width ($K \gg n_{\mathrm{train}}$), and SCN's training loss reaches the noise floor σ^2, ruling out underfitting as the explanation for its small excess risk.

Table 3. Generalization gap and excess risk for SCN on spherical data (corresponding to Figure 5a). As ambient dimension d increases with fixed patch size m, both metrics decrease—confirming 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

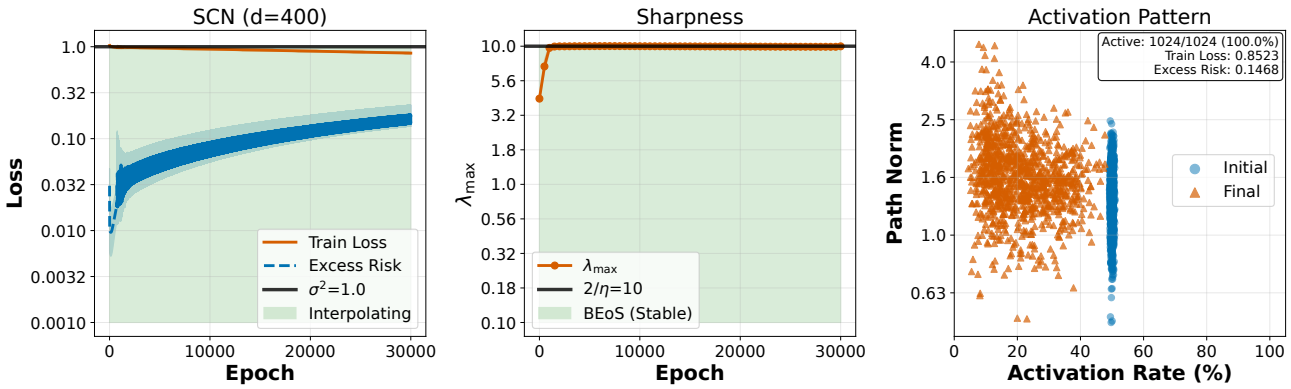


Figure 10. **SCN generalizes on high-dimensional spherical data.(Left)** With $d = 400$ and fixed patch size $m \ll d$, train loss plateaus near σ^2 while excess risk decreases to 0.15, confirming generalization rather than memorization. **(Middle)** Sharpness saturates at BEoS ($\lambda_{\max} \approx 2/\eta$). **(Right)** Neurons spread across moderate activation rates, unlike the sparse isolation in flat interpolation (Figure 5). This validates Theorem 4.4: when $m \ll d$, stability-induced regularization prevents overfitting. Results averaged over 5 seeds with $\eta = 0.2$, trained for 30k epochs.

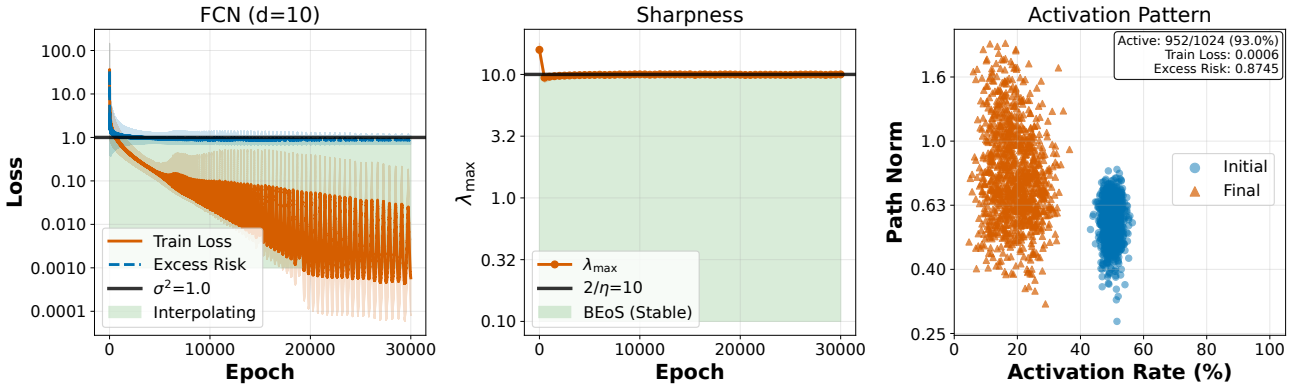


Figure 11. **FCN satisfies BEoS but still memorizes (Left)** FCN ($d = 10$) interpolates noisy labels (train loss $\to 0$) while excess risk remains $\approx \sigma^2$. **(Middle)** Sharpness saturates at BEoS. **(Right)** Activation pattern after training. Despite satisfying BEoS, FCN fails to generalize—confirming that on spherical data, stability constraints alone are insufficient without convolutional structure. Averaged over 5 seeds; with similar setting as Figure 10