GEOMETRY-AWARE FIRST-LAYER INITIALIZATION FOR RELU IMPLICIT NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit neural representations (INRs) provide a flexible framework for continuous signal modeling, yet ReLU MLPs under default initialization often struggle to recover fine-grained details despite their substantial representational capacity. Analyses based on representation rank and the neural tangent kernel (NTK) explain important aspects of this difficulty, but do not by themselves establish where the dominant limitation arises within the network. We investigate this question through Inlet Rank Collapse: despite the substantial dimensional expansion from input coordinates to first-layer features, default initialization of ReLU MLPs fails to produce a corresponding increase in effective representation dimensionality. We examine this mismatch as a bottleneck in the effective use of network capacity and use it to develop a common perspective on existing remedies, including positional encoding, sinusoidal activations, and batch normalization. This diagnosis motivates Geometric Initialization, a coordinate-only construction of first-layer weights and biases that enriches the initial feature representation while preserving the ReLU architecture and subsequent training procedure. Experiments show improved reconstruction and greater benefits from increasing network width, demonstrating that first-layer initialization alone can help ordinary ReLU networks make better use of their existing representational capacity.

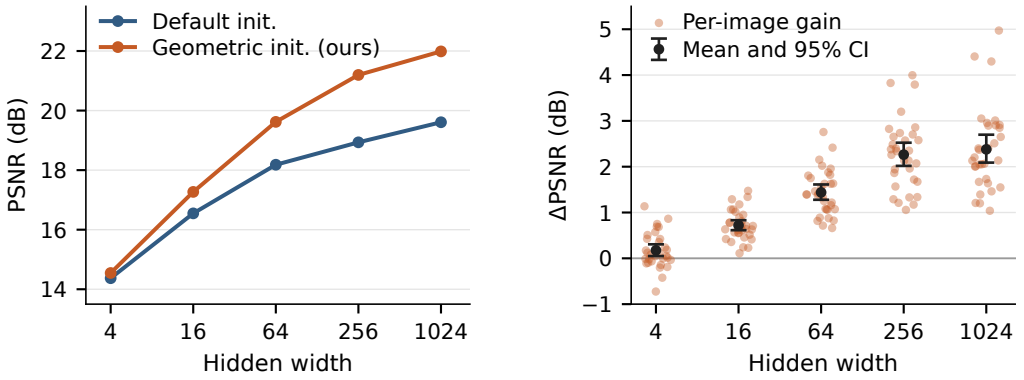
1 INTRODUCTION

Implicit neural representations (INRs) model continuous signals using coordinate-based neural networks. Despite their substantial representational capacity, ReLU multilayer perceptrons (MLPs) with default initialization often struggle to recover fine-scale details. Positional encoding (PE), sinusoidal activations, and batch normalization (BN) can alleviate this difficulty (Tancik et al., 2020; Sitzmann et al., 2020; Cai et al., 2024). Neural tangent kernel (NTK) and rank-based analyses connect these improvements to kernel spectra and representation diversity (Jacot et al., 2018; Zheng et al., 2021; Ramasinghe & Lucey, 2022). These perspectives raise a more localized question: where within a default-initialized ReLU network does the gap between nominal capacity and reconstruction performance emerge?

Our layer-wise decomposition of the NTK motivates examining the feature representations that enter each layer, beginning with the first coordinate-to-feature mapping. This first layer occupies a distinctive position: it maps low-dimensional coordinates into a much higher-dimensional hidden representation, whereas subsequent hidden layers typically operate between feature spaces of comparable size. Default initialization controls signal variance, but variance preservation alone does not ensure that this dimensional expansion produces a correspondingly diverse set of features. We use the term *Inlet Rank Collapse* to describe the resulting mismatch between nominal inlet width and initial feature diversity. Layer-wise NTK analysis connects these features to downstream parameter sensitivities, motivating the inlet as a potential site for improving the use of network capacity.

This inlet perspective also offers a common lens on several INR remedies. PE enriches the coordinate representation supplied to the MLP, showing that ReLU networks can fit fine-scale structure more effectively when given a different input representation (Tancik et al., 2020). Sinusoidal activations and BN are commonly applied throughout the network (Sitzmann et al., 2020; Cai et al., 2024); our experiments further show that restricting these interventions to the first layer can substantially improve reconstruction while retaining ReLU activations in subsequent layers. Together, these observations

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Fitting performance (b) Per-image gains

Figure 1: **First-layer initialization affects the fitting benefit of increasing network width.** Default and geometric initialization use the same ReLU architecture and training recipe, differing only in the initial parameters of the first affine layer. (a) Mean reconstruction PSNR after 5,000 updates on 32 RGB images at 512×512 resolution. Results are averaged over three seeds within each image and then across images. (b) Per-image gains, $\Delta\text{PSNR} = \text{PSNR}_{\text{geometric}} - \text{PSNR}_{\text{default}}$. Orange points show seed-averaged gains for individual images; black markers and error bars indicate the mean and 95% confidence intervals from an image bootstrap stratified by archive group. Geometric initialization provides larger gains at wider network settings. Widths are shown on a logarithmic axis, and PSNR is evaluated on the fitted pixels. Dataset and protocol details appear in Appendix D.

motivate targeting the inlet directly. They also suggest a complementary route to input encoding and activation modification: improving the initial coordinate-to-feature mapping without changing the network architecture or activation functions.

We pursue this route through *Geometric Initialization*, which uses coordinate geometry to arrange the directions and thresholds of first-layer ReLU neurons. The method changes only the initial weights and biases of the first layer; all layers are subsequently trained as usual. As shown in Figure 1, this intervention increases the reconstruction gains obtained from wider networks under a fixed training budget. These results demonstrate that an ordinary ReLU network can make better use of its existing width through first-layer initialization alone. More broadly, they motivate initialization schemes that explicitly consider feature diversity during dimensional expansion, alongside the conventional goal of controlling signal variance.

Our contributions are threefold:

- **Characterizing Inlet Rank Collapse.** We characterize the mismatch between nominal inlet width and initial feature diversity in default-initialized ReLU coordinate networks. Layer-wise NTK analysis and feature-spectrum measurements motivate a targeted first-layer intervention.
- **An inlet-centered perspective on INR remedies.** We connect input encoding and first-layer activation or normalization interventions through their effects on the coordinate-to-feature mapping, motivating initialization as a complementary route to improving INR fitting.
- **Geometric Initialization.** We develop a coordinate-based initialization that arranges first-layer ReLU directions and thresholds while preserving the architecture, activation functions, and subsequent training procedure. A fixed-budget width sweep with repeated runs and component controls evaluates its reconstruction benefits and shows improved use of network width.

2 Related Work

Implicit Neural Representations. Existing approaches improve INR fidelity through three broad types of intervention: *input representations*, such as Fourier features, positional encoding, hash

encoding, and learned coordinate mappings (Tancik et al., 2020; Zheng et al., 2021; Müller et al., 2022; Xie et al., 2023); *activation functions*, including periodic, Gaussian, wavelet, and variable-periodic alternatives (Sitzmann et al., 2020; Ramasinghe & Lucey, 2022; Saragadam et al., 2023; Liu et al., 2024); and *architectural components*, such as normalization and higher-order feature interactions (Cai et al., 2024; Chen et al., 2024). Structured ReLU formulations further demonstrate that accurate signal reconstruction does not require periodic activations (Shenouda et al., 2024). Our work explores a complementary direction: improving INR fitting solely through first-layer initialization, while preserving the ordinary ReLU architecture and activation functions.

Spectral Bias, Rank, and the NTK. Studies of spectral bias and the neural tangent kernel characterize frequency-dependent learning dynamics and the influence of input representations on optimization (Rahaman et al., 2019; Cao et al., 2021; Jacot et al., 2018; Tancik et al., 2020). Rank-based perspectives further connect positional encoding to representation diversity (Zheng et al., 2021). Layer-wise tangent-kernel analyses relate forward and backward signal propagation to feature learning (Lou et al., 2022), while NTK spectral analyses examine the effects of input geometry and activation functions (Murray et al., 2023). Closely related to our work, McGinnis et al. (2026) study stable-rank degradation during INR training and investigate rank-preserving pretraining and orthogonalized optimizer updates. Building on the layer-wise NTK decomposition, we examine how the feature embeddings entering each layer shape its tangent-kernel contribution. This analysis directs our attention to the initial coordinate-to-feature mapping, where our feature-spectrum measurements reveal limited feature diversity relative to nominal width. We characterize this mismatch as *Inlet Rank Collapse*, motivating a targeted intervention through first-layer initialization.

Initialization Strategies. Xavier and Kaiming initialization primarily aim to control signal variance (Glorot & Bengio, 2010; He et al., 2015). Prior work also incorporates data geometry into parameter initialization and construction: data-dependent ReLU initialization addresses neuron trainability (Shin & Karniadakis, 2020), while sampled networks construct hidden-layer weight–bias pairs from data geometry through sampling informed by training inputs and outputs (Bolager et al., 2023). For INRs, related studies examine the influence of first-layer selection on SIREN reconstruction (Vonderfecht & Liu, 2024), select embedding hyperparameters by matching initial output spectra to target signals (Kania et al., 2025), and derive initialization schemes for stable forward and backward variance propagation (Koneputugodage et al., 2025). Our method assigns first-layer weights and biases directly from coordinate geometry, retaining ordinary ReLU activations and subsequent unconstrained training of all layers. We evaluate this construction through width sweeps and component controls, examining initial inlet feature diversity alongside reconstruction performance.

3 Diagnosing Inlet Rank Collapse

Building on layer-wise NTK analysis (Lou et al., 2022), we compare weight and feature spectra and trace limited feature diversity back to the coordinate inlet. We further examine how reconstruction changes when normalization is introduced at different layers. These comparisons provide evidence for the role of embedding formation in fitting performance and motivate targeting the initial coordinate-to-feature mapping.

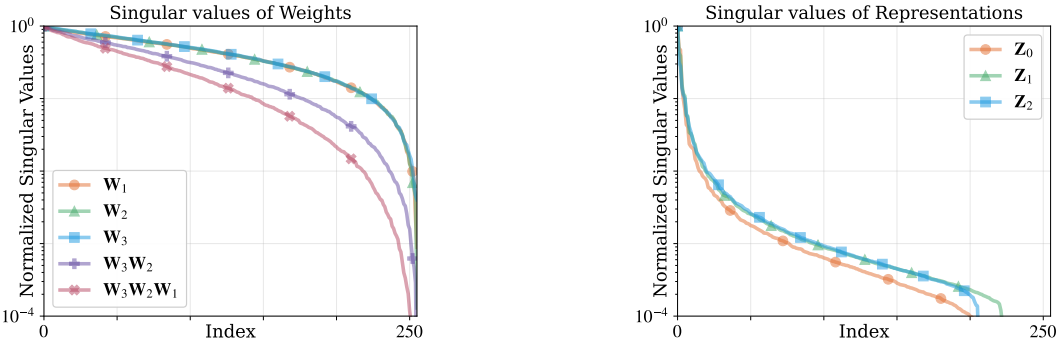
3.1 Layer-wise Parameter Sensitivities

Let $\mathbf{X} \in \mathbb{R}^{N\times P}$ denote the sampled coordinates. We consider an MLP consisting of an inlet embedding ϕ, L subsequent hidden layers, and a linear output head:

$$\begin{aligned} \mathbf{Z}_0 &= \phi(\mathbf{X}), \\ \mathbf{A}_i &= \mathbf{Z}_{i-1}\mathbf{W}_i + \mathbf{1}\mathbf{b}_i^\top, \qquad i = 1,\dots,L, \\ \mathbf{Z}_i &= \rho(\mathbf{A}_i), \qquad\qquad\quad i = 1,\dots,L, \\ \mathbf{Y} &= \mathbf{Z}_L\mathbf{W}_{\text{out}} + \mathbf{1}\mathbf{b}_{\text{out}}^\top. \end{aligned} \tag{1}$$

Here $\mathbf{Z}_i \in \mathbb{R}^{N\times D}$, $\mathbf{W}_i \in \mathbb{R}^{D\times D}$, $\mathbf{b}_i \in \mathbb{R}^D$, $\mathbf{W}_{\text{out}} \in \mathbb{R}^{D\times M}$, and $\mathbf{b}_{\text{out}} \in \mathbb{R}^M$. The inlet acts row-wise on the coordinates. For an ordinary ReLU network,

$$\phi(\mathbf{X}) = \text{ReLU}\left(\mathbf{X}\mathbf{W}_{\text{in}} + \mathbf{1}\mathbf{b}_{\text{in}}^\top\right),$$



(a) Weight spectra (b) Representation spectra

Figure 2: **Spectral concentration in weights and embeddings.** Embedding singular values decay more rapidly than those of the weight matrices and their products. The similar embedding spectra across layers motivate examining how feature diversity is established at the inlet. Weight-product labels use the column-vector convention; their transposes correspond to the forward products in our row-vector notation and have identical singular values.

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{P\times D}$ and $\mathbf{b}_{\text{in}} \in \mathbb{R}^{D}$.

We distinguish Feature Jacobians $\mathbf{J}_j^i = \partial \operatorname{vec}(\mathbf{Z}_i)/\partial \operatorname{vec}(\mathbf{Z}_j)$ from Weight Jacobians $\mathbf{G}_j^i = \partial \operatorname{vec}(\mathbf{Z}_i)/\partial \theta_j$, where θ_j includes weights and biases. Write $\mathbf{J}_i = \mathbf{J}_{i-1}^i$ and $\mathbf{G}_i = \mathbf{G}_i^i$. For the output head, define

$$\mathbf{J}_{\text{out}} = \frac{\partial \operatorname{vec}(\mathbf{Y})}{\partial \operatorname{vec}(\mathbf{Z}_L)}.$$

Let $\mathbf{G}_{\text{in}}$ and $\mathbf{G}_{\text{out}}$ denote the local parameter Jacobians of the inlet and output head, respectively. The complete output parameter Jacobian is

$$\begin{aligned}\mathbf{G}_{\text{all}}^{\text{out}} = \big[&\mathbf{G}_{\text{out}},\ \mathbf{J}_{\text{out}}\mathbf{G}_L,\ \mathbf{J}_{\text{out}}\mathbf{J}_L\mathbf{G}_{L-1},\ \ldots,\\ &\mathbf{J}_{\text{out}}\mathbf{J}_L\cdots\mathbf{J}_2\mathbf{G}_1,\ \mathbf{J}_{\text{out}}\mathbf{J}_L\cdots\mathbf{J}_1\mathbf{G}_{\text{in}}\big],\end{aligned} \tag{2}$$

and the empirical NTK is

$$\mathbf{K} = \mathbf{G}_{\text{all}}^{\text{out}}\left(\mathbf{G}_{\text{all}}^{\text{out}}\right)^{\top}. \tag{3}$$

This decomposition connects the rank structure of the NTK to weight transformations and feature embeddings. For a linear network, the weight-only output Jacobian of hidden layer k factorizes as

$$\mathbf{G}_{k,W}^{\text{out}} = \left(\mathbf{W}_{k+1}\cdots\mathbf{W}_L\mathbf{W}_{\text{out}}\right)^{\top} \otimes \mathbf{Z}_{k-1}.$$

Its rank is therefore the product of the ranks of the downstream weight product and the incoming embedding. ReLU additionally modulates these sensitivities through activation masks. This motivates examining the ranks and singular-value distributions of both weights and embeddings. Complete definitions, derivations, and rank bounds are provided in Appendix A.

3.2 From Feature Spectra to the Coordinate Inlet

Figure 2 contrasts the spectra of weights and their products with those of intermediate embeddings. The weight spectra decay relatively gradually, whereas the embedding spectra concentrate sharply in a few dominant directions and remain similar across the displayed layers. This contrast motivates tracing limited feature diversity back through the mappings that generate these embeddings.

When an embedding $\mathbf{Z}_i$ has limited feature diversity, we inspect the preceding representation and the mapping that produced it. Tracing this question backwards leads to $\mathbf{Z}_0 = \phi(\mathbf{X})$. Unlike hidden-to-hidden mappings of comparable width, the inlet expands low-dimensional coordinates into a high-dimensional feature space. Under default initialization, this increase in nominal dimension is not accompanied by a comparable increase in feature diversity, leaving the inlet embedding spectrally concentrated. We term this mismatch *Inlet Rank Collapse*.

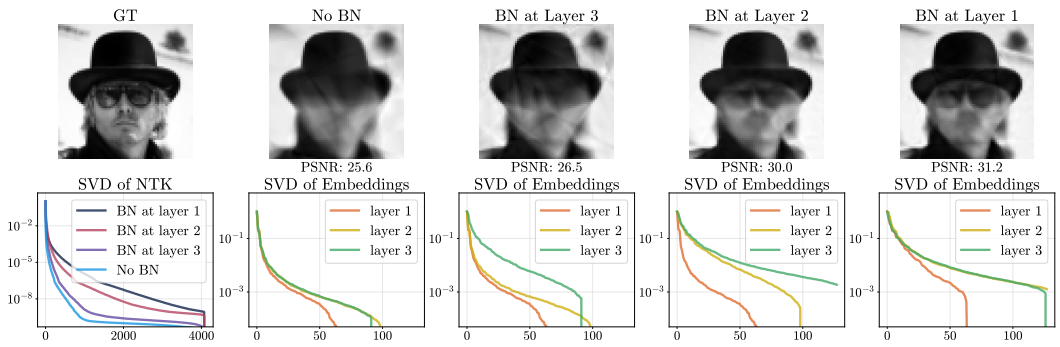


Figure 3: **Embedding spectra and reconstruction under different BN placements.** We compare the default network with BN introduced at each of three layer positions. Reconstruction, embedding spectra, and NTK spectra are shown together. In this example, earlier BN placement yields better reconstruction, with the best result obtained at the inlet. The figure numbers hidden stages from one: its Layers 1–3 correspond to $\mathbf{Z}_0, \mathbf{Z}_1, \mathbf{Z}_2$.

3.3 Embedding Interventions at Different Layers

BN modifies feature statistics and can improve coordinate-network fitting (Cai et al., 2024). We introduce BN at different layers to examine how changes in embedding spectra relate to reconstruction and whether intervention at the inlet is more effective than intervention at later stages.

Figure 3 shows concentrated embedding spectra and poor reconstruction in the default network. Introducing BN changes the embedding spectra and improves fitting, with reconstruction improving as the intervention moves towards the inlet. These observations support the role of embedding formation in the fitting difficulty and identify the initial coordinate-to-feature mapping as an effective intervention site.

This placement effect is consistent with the layer-wise analysis: inlet features supply the representations from which subsequent embeddings are formed, allowing an early intervention to influence multiple stages of feature formation. Appendix B provides a complementary thought experiment connecting feature diversity to width utilization.

4 Geometry-Aware First-Layer Initialization

Based on our preceding analysis, the inlet embedding layer is the primary bottleneck in ReLU-based coordinate MLPs. Under standard random initialization, a ReLU layer fails to expand the numerical rank of representations as the embedding dimension grows. Consequently, the rank deficiency introduced at the inlet persists throughout the network, locking the model into a low-capacity regime.

While most existing remedies introduce new architectural components, such as Fourier features or periodic activations, our theory suggests a more fundamental solution. If representation rank alone governs effective capacity, it should be possible to "unlock" the network without altering its architecture:

Can we restore the rank of the input embedding purely through initialization?

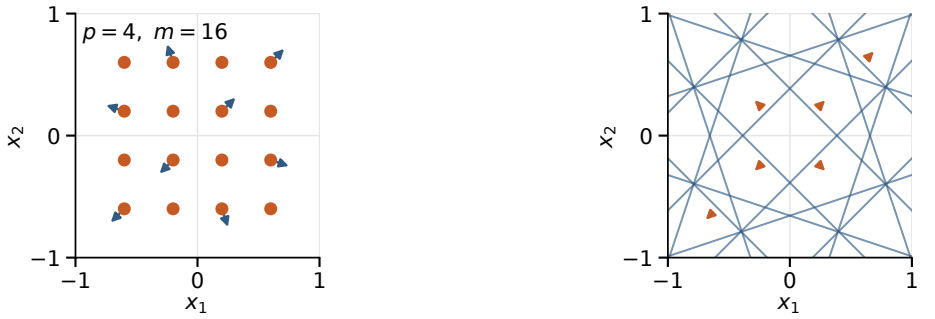
We show that the answer is affirmative. By geometrically designing the initial weights $\mathbf{W}^*$ and biases $\mathbf{b}^*$, we can force the input ReLU layer to generate a high-rank representation. We initialize the first layer of a ReLU MLP directly from the coordinate grid. For a hidden width $m = p^2$, we place $p \times p$ anchors uniformly inside the coordinate domain $[-1, 1]^2$:

$$v_{ab} = \left(\frac{2a}{p+1} - 1, \frac{2b}{p+1} - 1 \right), \qquad a, b \in \{1, \ldots, p\}. \tag{4}$$

We use even p so that no anchor lies at the origin. Enumerating these anchors by j, we assign one neuron to each anchor and set its weight and bias as

$$w_j^G = s\frac{v_j}{\|v_j\|_2}, \qquad b_j^G = s\left(-\|v_j\|_2 + \epsilon\right). \tag{5}$$

As illustrated in Figure 4, the weight points from the origin toward the anchor, and the bias places the ReLU activation boundary near that anchor, perpendicular to this direction. The offset ϵ makes the neuron slightly active at its anchor, with response $s\epsilon$. The common positive scale s controls response magnitude without moving the activation boundary.



(a) Anchors and normals (b) Activation boundaries

Figure 4: **Geometry-aware first-layer initialization.** Left: regularly spaced anchors and their radial weight directions. Right: the corresponding ReLU activation boundaries. Each boundary is perpendicular to its weight direction and lies near the associated anchor; arrows indicate the active side.

We use $\epsilon = 0.01$ and $s = 8$, fixed before the main study following the exploratory scale experiment. Only the first-layer initialization is changed. The remaining layers retain their default initialization, and all parameters are subsequently trained. The construction uses coordinates, not image intensities, and requires no change to the network architecture.

5 A Unified Inlet Perspective on Neural Representations

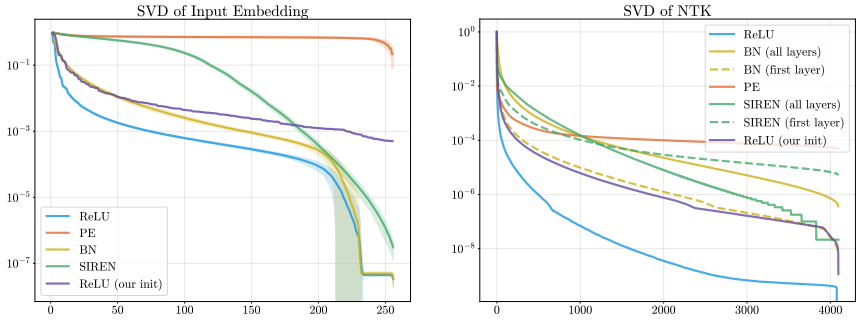
PE transforms input coordinates, SIREN introduces periodic activations, and BN normalizes intermediate features. Each changes the representation passed to subsequent layers, motivating an inlet perspective: how much reconstruction performance can be retained by modifying only the first-layer coordinate-to-feature mapping? We examine this question through spectral analysis and first-layer-only comparisons in 2-D image and 3-D occupancy representation, including our initialization as an intervention within a standard ReLU architecture.

5.1 Spectral Structure at the Inlet

We compare the singular value spectra of first-layer embeddings and the corresponding NTKs for default ReLU, PE, SIREN, BN, and our initialization, using a 256×256 coordinate grid and network width 256.

Figure 5(a) shows that our initialization lifts the smaller singular values of the inlet embedding relative to default ReLU. PE and SIREN also produce richer embedding spectra, suggesting that inlet spectra provide a useful basis for comparing these distinct approaches.

The corresponding NTK spectra also change (Figure 5(b)). In particular, the first-layer-only configurations show that inlet modifications can affect the whole-network kernel while subsequent layers remain standard ReLU. These paired spectra are consistent with a connection between inlet representations and downstream optimization geometry; the following comparisons examine reconstruction performance.



(a) Embedding spectrum (b) NTK spectrum

Figure 5: Singular value spectra of (a) first-layer embeddings and (b) the corresponding NTKs. Shaded regions indicate the standard deviation over 100 runs. Dashed curves denote configurations in which SIREN or BN is applied only at the first layer, with standard ReLU layers thereafter.

Table 1: Reconstruction PSNR (dB) averaged over 16 images from DIV2K Agustsson & Timofte (2017). "Standard configuration" denotes each method's usual placement of encoding, activation, or normalization. "First layer only" restricts SIREN or BN to the first layer, with standard ReLU layers thereafter.

Configuration	ReLU (default)	PE	SIREN	BN	ReLU (ours)
Standard configuration	23.57	27.15	27.29	25.04	25.61
First layer only	–	–	26.15	25.82	–

5.2 First-Layer Interventions in 2-D Image Representation

On 16 images from DIV2K Agustsson & Timofte (2017), first-layer-only SIREN achieves 26.15 dB, compared with 27.29 dB for its standard configuration and 23.57 dB for default ReLU (Table 1). It therefore retains a substantial portion of the improvement over ReLU, with a further gain from periodic activations throughout the network. First-layer-only BN instead achieves a higher mean PSNR than its standard configuration (25.82 versus 25.04 dB). Thus, the benefit of modifying subsequent layers differs between these methods.

Our initialization improves ReLU from 23.57 to 25.61 dB, a gain of 2.04 dB without changing the architecture. Although PE and SIREN achieve higher PSNR in this comparison, this gain shows that first-layer initialization alone can improve reconstruction quality.

5.3 First-Layer Interventions in 3-D Occupancy Representation

Table 2 extends the comparison to 3-D occupancy representation, with reconstructed meshes shown in Figure 8. Our initialization increases ReLU's IoU score from 90.98 to 95.75. First-layer-only SIREN and BN achieve 97.01 and 96.15, respectively, exceeding their standard configurations (96.36 and 95.17). In this comparison, extending either modification throughout the network provides no additional benefit.

Across these 2-D and 3-D comparisons, inlet-only interventions yield substantial gains over default ReLU, while the benefit of modifying downstream layers depends on the method and task. Together with the spectral observations, these results motivate examining the inlet as a common design choice across architectural and initialization-based interventions.

6 Width Scaling, Ablations, and Limitations

We examine how first-layer initialization affects the benefit of widening an ordinary ReLU network. We compare six initialization variants across five widths on 32 images, using three seeds and 5,000

Table 2: IoU scores for 3-D occupancy representation. "First layer only" restricts SIREN or BN to the first layer, retaining standard ReLU layers thereafter. Corresponding mesh visualizations are provided in Appendix C.

Configuration	ReLU (default)	PE	SIREN	BN	ReLU (ours)
Standard configuration	90.98	97.50	96.36	95.17	95.75
First layer only	–	–	97.01	96.15	–

Table 3: Reconstruction PSNR (dB) after $5{,}000$ updates, averaged over 32 images after seed averaging. Geometric denotes our initialization; Std-matched denotes the weight-standard-deviation-matched control. Bold indicates the highest mean at each width.

Method	4	16	64	256	1024
Default	14.37	16.55	18.18	18.93	19.61
Std-matched	14.26	16.62	18.37	19.02	18.43
Geometric	14.55	**17.27**	**19.62**	**21.20**	**21.99**
Shuffled	14.55	17.13	19.43	21.12	21.93
Weights-only	**14.64**	17.08	18.74	19.57	18.98
Bias-only	13.38	15.25	18.25	20.27	21.88

updates per run. Only the first-layer initialization changes, and all parameters are subsequently trained. We report reconstruction PSNR on the fitted pixels, averaging seeds within each image and then averaging over images. Experimental settings and statistical procedures are provided in Appendix D.

Initialization changes the benefit of width. Figure 1 and Table 3 show that Geometric's advantage over Default increases from 0.17 dB at width 4 to 2.38 dB at width 1024. At every tested width from 16 onward, Geometric outperforms both Default and Std-matched on all 32 seed-averaged images. At width 1024, the mean gain over Default has a paired-image bootstrap 95% interval of $[2.09, 2.70]$ dB. From width 64 to 1024, Geometric gains 2.37 dB, compared with 1.43 dB for Default. Geometric at width 256 also exceeds Default at width 1024 by 1.59 dB. These are fixed-update comparisons, rather than claims about converged performance or equal compute.

Scale and component effects. Matching weight standard deviation does not reproduce the gain, while shuffling the geometric biases retains most of it. At width 1024, Shuffled reaches 21.93 dB, compared with 21.99 dB for Geometric. Weights-only outperforms Bias-only at widths 4, 16, and 64, but the ordering reverses at 256 and 1024. Bias-only comes within 0.11 dB of Geometric at width 1024, suggesting a potentially stronger role for bias initialization in wider networks. However, the variants differ in their realized bias scales and threshold distributions, so this comparison does not identify independent causal contributions of weights and biases. Exact definitions and paired analyses are provided in Appendix E.

Spatial coverage and limitations. Geometric initially produces a central region where all first-layer features vanish (Figure 6). Its pixel fraction decreases from 21.15% at width 4 to 0.10% at width 1024. In a pilot on one image at width 64 with two seeds, flipping two or four nearest-anchor biases improves PSNR inside the original inactive region by 0.99–2.06 dB despite reducing initial stable rank. Whole-image changes range from -0.14 to $+0.03$ dB, without a consistent global benefit. All main results use the original Geometric initialization without repair. These findings motivate spatially resolved diagnostics alongside global spectral summaries; the geometry and pilot results are detailed in Appendix F.

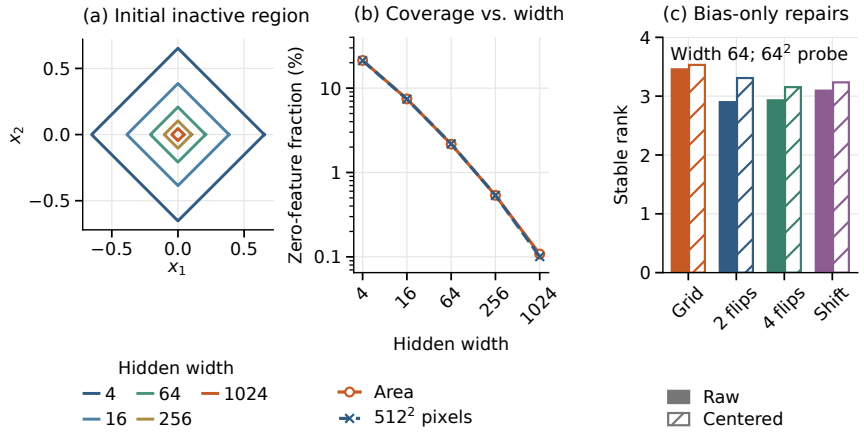


Figure 6: Spatial coverage of Geometric initialization. Left: the central inactive region across widths. Middle: its continuous area fraction and discrete pixel fraction. Right: raw and centered stable ranks at width 64 on a 64×64 coordinate probe. Bias flips and a common bias shift remove zero-feature pixels on the full image grid, while reducing stable rank on this probe. The right panel reports initialization diagnostics.

7 Conclusion

We investigated Inlet Rank Collapse: the mismatch between nominal first-layer width and initial feature diversity in default-initialized ReLU coordinate networks. Layer-wise NTK analysis and feature-spectrum measurements motivate examining the coordinate-to-feature mapping as a target for improving reconstruction. This inlet perspective also provides a common lens on positional encoding, sinusoidal activations, and normalization, highlighting how changes at the network entrance can benefit subsequent learning.

Guided by this perspective, we introduced Geometric Initialization, a coordinate-only construction of first-layer weights and biases that preserves the ReLU architecture and subsequent training procedure. Across a fixed-budget image-fitting study, this intervention improves reconstruction and increases the benefit obtained from wider networks. The component controls show that matching weight standard deviation alone does not reproduce these gains, while the spectral and spatial diagnostics indicate that a single global rank statistic does not fully explain fitting behavior. Together, our findings establish first-layer initialization as a practical means of improving the use of existing network width and motivate future initialization designs that jointly consider feature diversity, response scale, and spatial coverage.

Ethics Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

AI Use Statement

Generative AI tools assisted with drafting and polishing the manuscript text. The authors take full responsibility for the final content.

Reproducibility Statement

The geometric initialization is specified in Section 4. Appendix D details the image-fitting protocol, including the network architecture, optimization settings, random seeds, evaluation procedure, and statistical analysis. The exact initialization controls and component comparisons are provided in Appendix E, and the spatial-coverage diagnostics and repair-pilot protocol are described in Appendix F. These details specify the experimental conditions and the scope of the reported results.

References

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017_workshops/w12/papers/Agustsson_NTIRE_2017_Challenge_CVPR_2017_paper.pdf.

Erik L. Bolager, Iryna Burak, Chinmay Datar, Qing Sun, and Felix Dietrich. Sampling weights of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/c7201deff8d507a8fe2e86d34094e154-Abstract-Conference.html.

Zhicheng Cai, Hao Zhu, Qiu Shen, Xinran Wang, and Xun Cao. Batch normalization alleviates the spectral bias in coordinate networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25160–25171, 2024.

Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 2205–2211, 2021. doi: 10.24963/ijcai.2021/304. URL https://www.ijcai.org/proceedings/2021/304.

Yang Chen, Ruituo Wu, Yipeng Liu, and Ce Zhu. HOIN: High-order implicit neural representations, 2024. URL https://arxiv.org/abs/2404.14674. arXiv:2404.14674.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015. URL https://arxiv.org/abs/1502.01852.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1806.07572.

Adam Kania, Marko Mihajlovic, Sergey Prokudin, Jacek Tabor, and Przemysław Spurek. FreSh: Frequency shifting for accelerated neural representation learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/be4f28f5db4bd729c0360487c0822349-Abstract-Conference.html.

Chamin Hewa Koneputugodage, Yizhak Ben-Shabat, Sameera Ramasinghe, and Stephen Gould. VI3NR: Variance informed initialization for implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13477–13486, 2025. URL https://arxiv.org/abs/2504.19270.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. FINER: Flexible spectral-bias tuning in implicit NEural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2713–2722, 2024.

Yizhang Lou, Chris E. Mingard, and Soufiane Hayou. Feature learning and signal propagation in deep neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14248–14282, 2022. URL https://proceedings.mlr.press/v162/lou22a.html.

Julian McGinnis, Florian A. Hölzl, Suprosanna Shit, Florentin Bieder, Paul Friedrich, Mark Mühlau, Bjoern Menze, Daniel Rueckert, and Benedikt Wiestler. Optimizing rank for high-fidelity implicit neural representations. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2512.14366v3.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4): 102:1–102:15, 2022. doi: 10.1145/3528223.3530127. URL https://nvlabs.github.io/instant-ngp/.

Michael Murray, Hui Jin, Benjamin Bowman, and Guido Montúfar. Characterizing the spectrum of the NTK via a power series expansion. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Tvms8xrZHyR.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310, 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-MLPs. In *European Conference on Computer Vision*, pp. 142–158. Springer, 2022. URL https://arxiv.org/abs/2111.15135.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. WIRE: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/papers/Saragadam_WIRE_Wavelet_Implicit_Neural_Representations_CVPR_2023_paper.pdf.

Joseph Shenouda, Yamin Zhou, and Robert D. Nowak. ReLUs are sufficient for learning implicit neural representations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 44800–44814, 2024. URL https://proceedings.mlr.press/v235/shenouda24a.html.

Yeonjong Shin and George Em Karniadakis. Trainability of ReLU networks and data-dependent initialization. *arXiv preprint arXiv:1907.09696*, 2020. URL https://arxiv.org/abs/1907.09696v2.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020. URL https://arxiv.org/abs/2006.09661.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020. URL https://arxiv.org/abs/2006.10739.

Jeremy Vonderfecht and Feng Liu. Predicting the encoding error of SIRENs. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=iKPC7N85Pf.

Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. DINER: Disorder-invariant implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6143–6152, 2023.

Jianqiao Zheng, Sameera Ramasinghe, and Simon Lucey. Rethinking positional encoding. *arXiv preprint arXiv:2107.02561*, 2021. URL https://arxiv.org/abs/2107.02561.

A LAYER-WISE JACOBIANS AND DIAGNOSTIC SCOPE

This appendix defines the Jacobians used in Section 3 and derives their layer-wise relations. We use column-major vectorization throughout. The decomposition follows established tangent-kernel analysis (Lou et al., 2022); our notation distinguishes derivatives with respect to intermediate representations from derivatives with respect to model parameters.

A.1 NOTATION AND JACOBIAN DEFINITIONS

Let $\mathbf{Z}_0 = \mathbf{X} \in \mathbb{R}^{N\times P}$ denote the sampled input coordinates. We consider a network with L affine layers, including the linear output layer:

$$\begin{aligned} \mathbf{A}_i &= \mathbf{Z}_{i-1}\mathbf{W}_i + \mathbf{1}\mathbf{b}_i^\top, && i = 1,\dots,L, \\ \mathbf{Z}_i &= \rho(\mathbf{A}_i), && i = 1,\dots,L-1, \\ \mathbf{Z}_L &= \mathbf{A}_L. \end{aligned} \tag{6}$$

Here $\mathbf{Z}_i \in \mathbb{R}^{N\times d_i}$, $\mathbf{W}_i \in \mathbb{R}^{d_{i-1}\times d_i}$, $\mathbf{b}_i \in \mathbb{R}^{d_i}$, $d_0 = P$, and $d_L = M$. The vector $\mathbf{1} \in \mathbb{R}^N$ contains all ones. The first hidden representation $\mathbf{Z}_1$ is the coordinate-to-feature inlet embedding; equivalently, $\phi(\mathbf{X}) = \mathbf{Z}_1$.

For each affine layer, define

$$\theta_i = \begin{bmatrix} \mathrm{vec}(\mathbf{W}_i) \\ \mathbf{b}_i \end{bmatrix} \in \mathbb{R}^{p_i}, \qquad p_i = d_{i-1}d_i + d_i.$$

The term "Weight Jacobian" below includes derivatives with respect to both weights and biases.

Definition: Feature Jacobian. For $0 \le j < i \le L$, define

$$\mathbf{J}_j^i \coloneqq \frac{\partial\,\mathrm{vec}(\mathbf{Z}_i)}{\partial\,\mathrm{vec}(\mathbf{Z}_j)} \in \mathbb{R}^{Nd_i\times Nd_j}. \tag{7}$$

This derivative treats $\mathbf{Z}_j$ as the input to the subnetwork consisting of layers $j+1,\dots,i$, with its parameters held fixed. For adjacent layers, write

$$\mathbf{J}_i \coloneqq \mathbf{J}_{i-1}^i.$$

We also set $\mathbf{J}_i^i = \mathbf{I}_{Nd_i}$. The chain rule gives

$$\mathbf{J}_j^i = \mathbf{J}_i\mathbf{J}_{i-1}\cdots\mathbf{J}_{j+1}, \qquad j < i. \tag{8}$$

Definition: Weight Jacobian. For $1 \le j \le i \le L$, define

$$\mathbf{G}_j^i \coloneqq \frac{\partial\,\mathrm{vec}(\mathbf{Z}_i)}{\partial\theta_j} \in \mathbb{R}^{Nd_i\times p_j}, \tag{9}$$

with the input coordinates and all other parameter blocks held fixed. For a layer's sensitivity to its own parameters, write

$$\mathbf{G}_i \coloneqq \mathbf{G}_i^i.$$

Consequently,

$$\mathbf{G}_j^i = \mathbf{J}_j^i\mathbf{G}_j = \mathbf{J}_i\mathbf{J}_{i-1}\cdots\mathbf{J}_{j+1}\mathbf{G}_j. \tag{10}$$

For $j = i$, the product of Feature Jacobians is empty and equals the identity.

Complete parameter Jacobian and NTK. Order the complete parameter vector as

$$\theta_{\mathrm{all}} = [\theta_L^\top, \theta_{L-1}^\top, \dots, \theta_1^\top]^\top.$$

Then

$$\begin{aligned} \mathbf{G}_{\mathrm{all}}^L &\coloneqq \frac{\partial\,\mathrm{vec}(\mathbf{Z}_L)}{\partial\theta_{\mathrm{all}}} \\ &= [\mathbf{G}_L^L, \mathbf{G}_{L-1}^L, \dots, \mathbf{G}_1^L] \in \mathbb{R}^{NM\times\sum_i p_i}. \end{aligned} \tag{11}$$

Outputs index rows and parameters index columns. The empirical NTK is therefore

$$\mathbf{K} = \mathbf{G}_{\mathrm{all}}^L(\mathbf{G}_{\mathrm{all}}^L)^\top \in \mathbb{R}^{NM\times NM}. \tag{12}$$

For scalar outputs, this reduces to an $N\times N$ sample kernel. For multiple outputs, it retains cross-output kernel entries.

A.2 AFFINE-LAYER JACOBIANS

The vectorization identity

$$\operatorname{vec}(\mathbf{AXB}) = (\mathbf{B}^\top \otimes \mathbf{A}) \operatorname{vec}(\mathbf{X})$$

gives the Feature Jacobian of the affine preactivation:

$$\mathbf{J}_i^{\text{aff}} := \frac{\partial \operatorname{vec}(\mathbf{A}_i)}{\partial \operatorname{vec}(\mathbf{Z}_{i-1})} = \mathbf{W}_i^\top \otimes \mathbf{I}_N. \tag{13}$$

Differentiating with respect to weights and biases yields

$$\mathbf{G}_i^{\text{aff}} := \frac{\partial \operatorname{vec}(\mathbf{A}_i)}{\partial \theta_i} = \left[\mathbf{I}_{d_i} \otimes \mathbf{Z}_{i-1}, \quad \mathbf{I}_{d_i} \otimes \mathbf{1}\right]. \tag{14}$$

The first block has $d_{i-1}d_i$ columns and the second has d_i columns, matching the parameter ordering in θ_i.

Up to a column permutation, $\mathbf{G}_i^{\text{aff}}$ equals $\mathbf{I}_{d_i} \otimes [\mathbf{Z}_{i-1}, \mathbf{1}]$. Using $\operatorname{rank}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{rank}(\mathbf{A}) \operatorname{rank}(\mathbf{B})$, we obtain

$$\operatorname{rank}(\mathbf{G}_i^{\text{aff}}) = d_i \operatorname{rank}\left([\mathbf{Z}_{i-1}, \mathbf{1}]\right). \tag{15}$$

This identity characterizes the local affine parameter Jacobian. Its propagation to the network output additionally depends on subsequent layers.

A.3 RELU LAYERS AND THE OUTPUT JACOBIAN

For an elementwise activation $\mathbf{Z}_i = \rho(\mathbf{A}_i)$, define

$$\mathbf{M}_i = \operatorname{diag}\left(\operatorname{vec}(\rho'(\mathbf{A}_i))\right) \in \mathbb{R}^{Nd_i \times Nd_i}.$$

At differentiable points, the chain rule yields

$$\begin{aligned} \mathbf{J}_i &= \mathbf{M}_i \mathbf{J}_i^{\text{aff}} = \mathbf{M}_i (\mathbf{W}_i^\top \otimes \mathbf{I}_N), \\ \mathbf{G}_i &= \mathbf{M}_i \mathbf{G}_i^{\text{aff}} \\ &= \mathbf{M}_i \left[\mathbf{I}_{d_i} \otimes \mathbf{Z}_{i-1}, \quad \mathbf{I}_{d_i} \otimes \mathbf{1}\right]. \end{aligned} \tag{16}$$

For the linear output layer, set $\mathbf{M}_L = \mathbf{I}_{NM}$. At ReLU kinks, numerical implementations use a chosen derivative convention.

The output sensitivity to layer k is

$$\mathbf{G}_k^L = \mathbf{J}_L \mathbf{J}_{L-1} \cdots \mathbf{J}_{k+1} \mathbf{G}_k. \tag{17}$$

Thus, each local parameter sensitivity is propagated through the Feature Jacobians of subsequent layers.

These are finite-network chain-rule identities. They require no statistical independence between forward and backward quantities. The activation masks remain interleaved with the weight factors; in general, they cannot be removed to obtain a single Kronecker product of a bare downstream weight product and an incoming embedding.

A.4 THE LINEAR SPECIAL CASE

For a linear network, define the downstream weight product in forward order:

$$\mathbf{B}_k = \mathbf{W}_{k+1} \mathbf{W}_{k+2} \cdots \mathbf{W}_L \in \mathbb{R}^{d_k \times M}, \qquad \mathbf{B}_L = \mathbf{I}_M.$$

Holding all other parameters fixed, a perturbation of $\mathbf{W}_k$ changes the output by

$$\delta \mathbf{Z}_L = \mathbf{Z}_{k-1} \, \delta \mathbf{W}_k \, \mathbf{B}_k.$$

Vectorization gives the weight-only output Jacobian

$$\mathbf{G}_{k,W}^L = \mathbf{B}_k^\top \otimes \mathbf{Z}_{k-1}. \tag{18}$$

Consequently,

$$\operatorname{rank}(\mathbf{G}_{k,W}^{L}) = \operatorname{rank}(\mathbf{B}_k)\operatorname{rank}(\mathbf{Z}_{k-1}). \tag{19}$$

Including the bias parameters gives

$$\mathbf{G}_k^L = \left[\mathbf{B}_k^\top \otimes \mathbf{Z}_{k-1}, \quad \mathbf{B}_k^\top \otimes \mathbf{1}\right],$$

and therefore

$$\operatorname{rank}(\mathbf{G}_k^L) = \operatorname{rank}(\mathbf{B}_k)\operatorname{rank}\left([\mathbf{Z}_{k-1}, \mathbf{1}]\right). \tag{20}$$

The transpose reverses the downstream product order: $\mathbf{B}_k^\top = \mathbf{W}_L^\top \cdots \mathbf{W}_{k+1}^\top$. This special case makes the coupling between incoming representations and downstream sensitivities explicit. Its factorization does not extend unchanged to networks with interleaved ReLU masks.

A.5 Layer-wise Rank Bounds

The complete output parameter Jacobian is

$$\mathbf{G}_{\text{all}}^L = [\mathbf{G}_L^L, \mathbf{G}_{L-1}^L, \ldots, \mathbf{G}_1^L].$$

Its column space contains the column space of each block and is contained in their sum. Hence,

$$\begin{aligned} \max_k \operatorname{rank}(\mathbf{G}_k^L) &\le \operatorname{rank}(\mathbf{G}_{\text{all}}^L) \\ &\le \sum_{k=1}^{L} \operatorname{rank}(\mathbf{G}_k^L). \end{aligned} \tag{21}$$

For the affine and elementwise-activation layers defined above, the chain rule and the local affine rank identity give

$$\operatorname{rank}(\mathbf{G}_k^L) \le \operatorname{rank}(\mathbf{G}_k) \le d_k \operatorname{rank}\left([\mathbf{Z}_{k-1}, \mathbf{1}]\right). \tag{22}$$

Combining these inequalities yields

$$\operatorname{rank}(\mathbf{G}_{\text{all}}^L) \le \sum_{k=1}^{L} d_k \operatorname{rank}\left([\mathbf{Z}_{k-1}, \mathbf{1}]\right). \tag{23}$$

This upper bound can be loose; the output dimension also imposes $\operatorname{rank}(\mathbf{G}_{\text{all}}^L) \le NM$.

The NTK decomposes into positive semidefinite terms:

$$\mathbf{K} = \sum_{k=1}^{L} \mathbf{G}_k^L (\mathbf{G}_k^L)^\top, \qquad \operatorname{rank}(\mathbf{K}) = \operatorname{rank}(\mathbf{G}_{\text{all}}^L). \tag{24}$$

For any $\mathbf{u} \in \mathbb{R}^{NM}$,

$$\mathbf{u}^\top \mathbf{K} \mathbf{u} = \sum_{k=1}^{L} \left\| (\mathbf{G}_k^L)^\top \mathbf{u} \right\|_2^2.$$

Since every summand is nonnegative,

$$\ker \mathbf{K} = \bigcap_{k=1}^{L} \ker\left((\mathbf{G}_k^L)^\top\right). \tag{25}$$

A direction absent from one parameter block need not be absent from the complete NTK. For example, $\operatorname{diag}(1, 0)$ and $\operatorname{diag}(0, 1)$ each have rank one, while their sum has rank two.

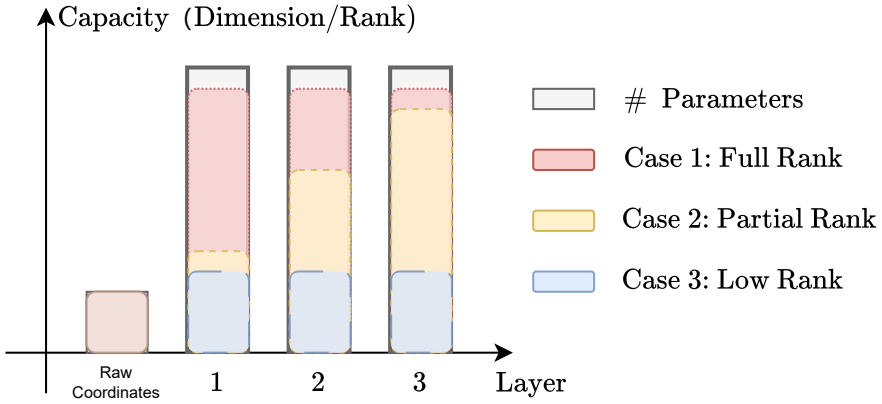


Figure 7: **Feature diversity and its location in the network.** The three cases illustrate persistent feature redundancy (Case 3), later diversification (Case 2), and diversification at the inlet (Case 1). Gray regions represent nominal feature dimensions, while colored regions schematically represent feature diversity available at each stage. The illustration does not quantify effective capacity or the number of usable parameters.

A.6 Scope and Interpretation

The identities above motivate tracing limited feature diversity back through the mappings that generate successive embeddings. This diagnostic does not assume that representation rank decreases monotonically: nonlinear activations can increase matrix rank, and a downstream intervention can alter the backward sensitivities of earlier parameter blocks. The stated bounds concern algebraic rank, rather than stable rank or conditioning.

Training-mode BN couples samples through batch statistics. Its input Jacobian is therefore not merely a diagonal channel scaling; applying the chain rule requires the full batch Jacobian and inclusion of its trainable affine parameters. The local affine and elementwise-activation formulas should not be applied unchanged to a complete BN-containing block. The BN-position comparison consequently probes intervention location rather than an isolated change in rank.

Finally, these finite-network Jacobian identities hold instantaneously. They do not require infinite width, but neither do they imply fixed-kernel dynamics throughout the finite-width Adam training used in our experiments.

B A Thought Experiment on Width Utilization

Consider networks with the same architecture but different feature responses. Three schematic cases distinguish where diverse spatial responses become available: persistent redundancy from the inlet onwards, diversification only at a later stage, and diversification beginning at the inlet.

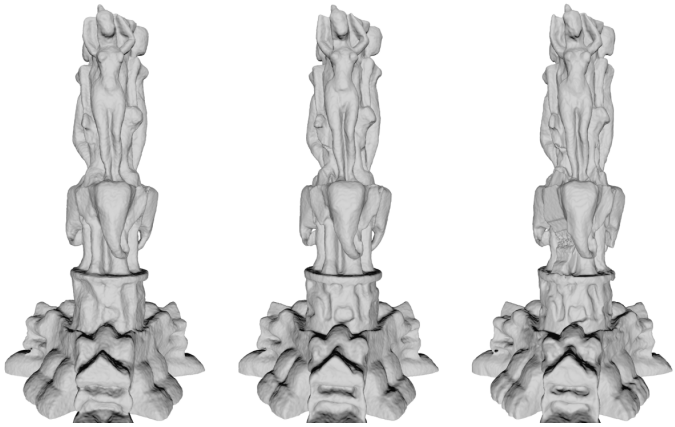
Persistent redundancy. Additional channels may reproduce identical or highly similar spatial responses. In the limiting case of exact duplication, widening adds columns to the feature matrix without enlarging its column space. Nominal feature dimension and algebraic rank therefore need not increase together.

Later diversification. A later nonlinear mapping may generate new feature directions from the representations it receives. These directions become available to subsequent layers. This case does not imply that earlier parameters are unaffected: their output sensitivities also depend on the downstream Feature Jacobians.

Diversification at the inlet. An inlet mapping with varied spatial responses supplies a richer initial feature set to the next layer and can influence the representations formed thereafter. This motivates



(a) ReLU (default) (b) PE (c) SIREN (d) SIREN (first layer)



(e) BN (f) BN (first layer) (g) ReLU (ours)

Figure 8: Meshes reconstructed from occupancy volumes. First-layer-only SIREN and BN retain standard ReLU layers after the first layer. Quantitative IoU scores are reported in Table 2.

examining first-layer initialization as a means of changing how an otherwise fixed architecture uses additional channels.

For approximately redundant features, spectral concentration is more informative than a binary distinction between full and deficient algebraic rank. Furthermore, all layers remain trainable, so initially similar channels may diversify during optimization. These schematic cases therefore motivate an empirical question rather than determine a training outcome: under a fixed optimization budget, does changing only first-layer initialization alter the fitting benefit of increasing network width?

The width sweep in Section 6 examines this question directly. Its reconstruction results and the spectral measurements in Appendix E.5 are evaluated separately; neither a schematic notion of feature diversity nor a single rank statistic substitutes for the observed training outcomes.

C Qualitative Comparison of 3-D Occupancy Representations

Figure 8 shows the reconstructed meshes corresponding to the IoU scores in Table 2. The comparison includes default ReLU, PE, SIREN, BN, their first-layer-only variants where applicable, and ReLU with our initialization.

D WIDTH-STUDY PROTOCOL AND STATISTICAL ANALYSIS

D.1 IMAGES AND NETWORK ARCHITECTURE

The width study uses all 32 RGB images in a supplied DIV2K-derived collection, organized into two disjoint groups of 16 images. Each image is stored at 512×512 resolution and is used without additional resizing or cropping. The group labels are used for aggregation and do not denote verified official DIV2K splits. Each image is fitted independently, with all pixels used for both training and evaluation. The reported metric is therefore reconstruction PSNR on the fitted coordinates.

The network contains four affine layers with dimensions

$$2 \to m \to m \to m \to 3,$$

with ReLU activations after the first three layers and a linear output. We use $m \in \{4, 16, 64, 256, 1024\}$. The number of trainable parameters is

$$2m^2 + 8m + 3.$$

No positional encoding or normalization layer is used.

Default follows the standard PyTorch `nn.Linear` initialization: for a layer with fan-in d, both weights and biases are sampled from $[-1/\sqrt{d}, 1/\sqrt{d}]$. In particular, the default first-layer biases are not zero. All interventions are applied after constructing the same base model, preserving the downstream tensors at each paired width and seed. All parameters remain trainable.

D.2 OPTIMIZATION AND EVALUATION

Each run uses $5{,}000$ full-image Adam updates with $(\beta_1, \beta_2) = (0.9, 0.999)$, optimizer epsilon 10^{-8}, and zero weight decay. At update index $t = 0, \ldots, 4999$, the learning rate is

$$\eta_t = 10^{-3}(0.1)^{t/5000}. \tag{26}$$

The scheduler steps after each update. Each update uses all $262{,}144$ pixels. The objective is RGB mean squared error.

Runs use seeds 0, 1, and 2, PyTorch $2.5.1 + \mathrm{cu}121$ on native Windows, and an RTX 4090. Computation uses FP32, deterministic algorithms, and disabled TF32 for both CUDA matrix multiplication and cuDNN. The complete study contains

$$32 \times 5 \times 3 \times 6 = 2{,}880$$

image–width–seed–method combinations.

Evaluation is recorded at initialization and every 100 updates through update $5{,}000$. The reported endpoint is the final update rather than the best checkpoint. With RGB targets normalized to $[0, 1]$, PSNR is

$$\mathrm{PSNR} = -10 \log_{10}(\mathrm{MSE}). \tag{27}$$

Equal update counts do not imply equal training cost across widths, and the finite training budget does not establish convergence.

D.3 AGGREGATION AND UNCERTAINTY

Let $P^a_{i,m,r}$ denote the PSNR for method a, image i, width m, and seed r. We first average seeds within each image:

$$\overline{P}^a_{i,m} = \frac{1}{3} \sum_{r=0}^{2} P^a_{i,m,r}. \tag{28}$$

The reported mean is

$$\overline{P}^a_{m} = \frac{1}{32} \sum_{i=1}^{32} \overline{P}^a_{i,m}. \tag{29}$$

Table 4: Implemented first-layer parameters after initialization scaling. P_π permutes the bias vector using a dedicated seed.

Method	Weights	Biases
Default	W^D	b^D
Std-matched	αW^D	αb^D
Geometric	sU	sc
Shuffled	sU	$sP_\pi c$
Weights-only	sU	sb^D
Bias-only	αW^D	αc

This is a mean of per-image PSNR values, rather than PSNR computed from pooled pixel MSE. Method differences are paired within images, widths, and seeds. An image win denotes a positive difference after seed averaging.

The main width-analysis intervals use $20{,}000$ paired-image bootstrap replicates. Within each replicate, 16 images are sampled with replacement from each of the two collection groups. All methods and widths for a selected image are retained together. We report percentile 95% intervals.

The component-analysis intervals use $20{,}000$ paired-image bootstrap replicates from the pooled 32 images, also after seed averaging. These pooled intervals and the group-stratified main-study intervals are distinct analysis conventions; their means agree, but their interval endpoints need not coincide. Both are descriptive, unadjusted intervals conditional on the available seeds. Repeated seeds are not treated as independent images.

D.4 Additional width comparisons

At width 4, Geometric improves over Default on 20 of 32 image averages. At every tested width from 16 onward, it improves over both Default and Std-matched on all 32 images. At width 1024, the image-wise gain over Default ranges from 1.04 to 4.97 dB.

From width 64 to 1024, the additional widening gain of Geometric relative to Default is 0.94 dB, with a paired 95% interval of $[0.74, 1.16]$. This contrast is positive on 31 of 32 images. From width 256 to 1024, Geometric gains 0.79 dB and Default gains 0.67 dB.

Geometric at width 256 exceeds Default at width 1024 by 1.59 dB, with an exploratory pointwise 95% interval of $[1.39, 1.81]$. The difference is positive on all 32 image averages. The networks contain $133{,}123$ and $2{,}105{,}347$ parameters, respectively. This is a descriptive cross-width contrast within the completed sweep, rather than a matched-compute comparison or a compression result.

E Initialization Controls and Component Analysis

E.1 Exact initialization rules

Let (W^D, b^D) denote the unscaled default first-layer parameters and (U, c) the unscaled geometric construction. We use the fixed multiplier $s = 8$ and define

$$\alpha = s\,\frac{\mathrm{std}(U)}{\mathrm{std}(W^D)}, \tag{30}$$

where standard deviation is the population statistic over all entries of the weight matrix. Components are assembled first, after which weights and biases are jointly multiplied by the specified scalar.

The names Weights-only and Bias-only refer to the retained unscaled geometric component. Bias-only uses αc, rather than Geometric's final bias vector sc. Weights-only uses sb^D, rather than the Std-matched bias vector αb^D. These comparisons therefore do not independently hold bias amplitude, activation RMS, or threshold distributions fixed.

Positive joint scaling preserves each neuron's activation boundary:

$$\{x : (aw)^\top x + ab = 0\} = \{x : w^\top x + b = 0\}, \qquad a > 0. \tag{31}$$

Table 5: Geometric minus each component at $5{,}000$ updates, in dB. Positive values favor Geometric. Intervals are pooled paired-image bootstrap 95% intervals. Wins count positive seed-averaged image differences.

Comparator	Width	Mean	95% interval	Wins /32
Shuffled	4	0.000002	$[-0.000033, 0.000049]$	13
	16	0.1356	$[0.0933, 0.1814]$	29
	64	0.1858	$[0.1460, 0.2286]$	32
	256	0.0733	$[0.0390, 0.1029]$	29
	1024	0.0536	$[0.0225, 0.0856]$	24
Weights-only	4	-0.0978	$[-0.2071, 0.0003]$	13
	16	0.1865	$[0.1099, 0.2596]$	25
	64	0.8797	$[0.7699, 0.9941]$	32
	256	1.6285	$[1.4403, 1.8392]$	32
	1024	3.0086	$[2.6532, 3.3976]$	32
Bias-only	4	1.1669	$[0.8866, 1.4741]$	31
	16	2.0176	$[1.7297, 2.3267]$	32
	64	1.3750	$[1.1725, 1.6125]$	32
	256	0.9246	$[0.7858, 1.0774]$	32
	1024	0.1074	$[0.0517, 0.1693]$	23

It scales the ReLU response but does not make the boundaries of different component variants coincide. Shuffled preserves the weights, bias multiset, and common scale, while changing the association between directions and thresholds.

E.2 Paired component results

Table 5 reports Geometric minus each component variant. Differences are computed before rounding the method means displayed in the main text.

At width 1024, Shuffled reaches 21.9331 dB and Geometric reaches 21.9867 dB. The original pairing gives a small positive mean difference, rather than accounting for most of the improvement over Default. At width 64, the pairing advantage is larger and positive on all 32 image averages. A small mean gap does not establish statistical equivalence.

At width 4, all four ideal geometric biases are equal. Permutation changes the recorded FP32 biases by less than 7.2×10^{-7}. The final mean PSNR difference is approximately 2×10^{-6} dB. This is an almost-null intervention rather than evidence about the effect of substantive pairing changes.

E.3 Seed and image-group sensitivity

At width 1024, averaging seeds 0 and 1 gives a Geometric–Shuffled difference of 0.0925 dB, whereas seed 2 alone gives -0.0243 dB. The corresponding Geometric–Bias-only differences are 0.1519 and 0.0184 dB. The small wide-network differences are therefore sensitive to the available seeds.

Using all three seeds, the two 16-image groups give Geometric–Shuffled differences of 0.0479 and 0.0592 dB. Their Geometric–Bias-only differences are 0.1082 and 0.1066 dB. The main results retain all images and seeds.

E.4 Realized scales and interpretation

At width 1024, Geometric has initial bias mean, standard deviation, and RMS

$$(\mu_b, \sigma_b, r_b) = (-5.8539,\ 2.2068,\ 6.2560),$$

whereas Std-matched has

$$(\mu_b, \sigma_b, r_b) = (-0.1064,\ 5.5665,\ 5.5685).$$

Table 6: Initial feature stable ranks at width 1024. These characterize first-layer features, not the whole-network NTK.

Method	Raw	Centered
Default	1.509	2.135
Std-matched	1.509	2.135
Geometric	3.322	3.510
Shuffled	3.294	3.339
Weights-only	1.966	2.273
Bias-only	3.092	3.035

Geometric thus has smaller bias standard deviation, but larger bias RMS and a strongly negative mean. Its geometric biases are all negative, whereas the mean negative-bias fraction of Std-matched is 0.5088.

The bias RMS values for Weights-only and Bias-only are 3.2245 and 10.8032, respectively. Bias-only therefore does not preserve the final bias amplitude of Geometric. These differences prevent attributing the component ordering solely to weight directions or bias geometry.

The exploratory width dependence nevertheless provides a useful observation: Bias-only exceeds Weights-only by 0.7039 dB at width 256 and 2.9012 dB at width 1024, while the ordering reverses at the three smaller widths. This motivates further controls on threshold distributions and response amplitudes.

E.5 Feature spectra

Feature diagnostics use first-layer post-ReLU responses on a 32×32 coordinate probe, formed by selecting approximately equally spaced pixel indices along each image axis. For a feature matrix H, stable rank is

$$\mathrm{sr}(H) = \frac{\|H\|_F^2}{\|H\|_2^2}. \tag{32}$$

Centered stable rank is computed after subtracting each feature column's mean.

Weights-only has higher initial raw and centered stable rank than Default, but lower final PSNR. Bias-only approaches Geometric's reconstruction performance despite lower stable rank. These observations limit a single-score explanation of optimization without implying that spectral structure is irrelevant.

F Spatial Coverage and Repair Pilot

F.1 The initial inactive region

Consider the geometric construction with $p^2 = m$ anchors, even p, and geometric offset ϵ. Write

$$t_0 = \frac{1}{p+1}, \qquad r_{\min} = \frac{\sqrt{2}}{p+1}, \qquad a_p = \frac{2}{p+1} - \sqrt{2}\epsilon.$$

For $0 \le \epsilon \le r_{\min}$, the common zero-feature region in $[-1,1]^2$ is

$$\mathcal{D}_p = \left\{ x \in [-1,1]^2 : |x_1| + |x_2| \le a_p \right\}. \tag{33}$$

Its area fraction is $a_p^2/2$. This is a region of input locations where all features vanish, rather than a collection of neurons that are inactive everywhere. A common positive rescaling of weights and biases does not move its boundary.

The four nearest anchors $(\pm t_0, \pm t_0)$ impose the diamond in Equation 33. To verify that more distant anchors impose no additional restriction, let the absolute coordinates of an anchor be $A \ge B \ge t_0$ and let $r = \sqrt{A^2 + B^2}$. For x in the diamond, the anchor's inner product with x is at most $a_p A$,

Table 7: Initial inactive-region size for Geometric. Continuous area fractions and discrete pixel fractions on the 512×512 coordinate grid are reported separately.

Width	Area (%)	Zero-feature pixels	Pixels (%)
4	21.2894	55,444	21.1502
16	7.4443	19,404	7.4020
64	2.1649	5,724	2.1835
256	0.5357	1,404	0.5356
1024	0.1079	264	0.1007

Table 8: Repair-pilot outcomes after $5{,}000$ updates. Whole and Region are PSNR values in dB. Differences are relative to original Geometric at the same seed. Region denotes the fixed original inactive mask.

Method	Seed	Whole	Region	Δ Whole	Δ Region	Δ Outside
Geometric	0	20.256	19.500	–	–	–
Two-bias flip	0	20.191	21.033	-0.065	$+1.533$	-0.101
Four-bias flip	0	20.118	20.803	-0.138	$+1.303$	-0.171
Geometric	1	20.126	19.193	–	–	–
Two-bias flip	1	20.152	20.186	$+0.026$	$+0.993$	$+0.002$
Four-bias flip	1	20.025	21.255	-0.101	$+2.062$	-0.148

while

$$r^2 - \epsilon r - a_p A = (A - t_0)^2 + B^2 - t_0^2 + \epsilon(\sqrt{2}A - r) \geq 0. \tag{34}$$

Hence the remaining features also vanish there.

At $\epsilon = r_{\min}$, the region reduces to the origin. For $\epsilon > r_{\min}$, the nearest-anchor constraints are incompatible and the common zero-feature set is empty. Table 7 reports the implemented setting $\epsilon = 0.01$.

F.2 Repair-pilot protocol

We fit one supplied RGB image at its cached resolution of 512×512 using width 64, three hidden layers, $s = 8$, $\epsilon = 0.01$, and seeds 0 and 1. The three configurations are original Geometric, a two-bias flip, and a four-bias flip. The two-bias intervention flips the signs of two nearest-anchor biases associated with orthogonal normals; the four-bias intervention flips all four nearest-anchor biases.

For each seed, the configurations share the original weight tensors and downstream initialization. Only the specified bias signs differ initially. All parameters are trainable. The optimizer, precision, and $5{,}000$-update budget follow Appendix D. No feature-RMS matching is applied. The common bias shift shown in Figure 6 is an initialization-only diagnostic, not a training arm in this pilot.

Regional evaluation uses the original Geometric initial zero-feature mask, containing $5{,}724$ pixels, or 2.1835% of the image. This mask is fixed across configurations, seeds, and training steps. PSNR uses unclipped floating-point RGB predictions and targets in $[0, 1]$. We separately evaluate the whole image, the original inactive region, and its complement. Regional MSEs, rather than regional PSNRs, combine by pixel-fraction weighting.

F.3 Paired reconstruction results

Both interventions improve reconstruction inside the fixed original region for both seeds. Whole-image effects are mixed for the two-bias flip and negative for the four-bias flip. Because the region covers only 2.1835% of the pixels, its gains can be outweighed by errors elsewhere.

After $5{,}000$ updates, original Geometric retains zero first-layer features at $2{,}931$ and $3{,}174$ original-region pixels for seeds 0 and 1, respectively. These correspond to 51.21% and 55.45% of the original region. Both interventions have no zero-feature pixels at any recorded evaluation.

F.4 Spectra and interpretation

On the pilot's 32×32 rounded-index coordinate probe, initial raw and centered stable ranks are $3.450/3.527$ for Geometric, $2.919/3.314$ for the two-bias flip, and $2.944/3.164$ for the four-bias flip. These probe settings differ from the 64×64 initialization diagnostic in Figure 6. Initial full-image feature RMS also changes, from 1.140 to 1.257 and 1.364, respectively.

The regional improvements therefore occur despite lower raw and centered stable ranks. This demonstrates that global spectral concentration does not fully describe local fitting behavior. It does not establish a coverage-only causal effect: the interventions also change feature amplitudes, gating, and correlations. The pilot is limited to one image, one width, and two seeds, and does not establish a general ranking of repaired and original initializations.