OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Data Geometry Determines Generalization Below the Edge-of-Stability

Tongtong Liang TTLIANG@UCSD.EDU
Department of Mathematics, UCSD

Alexander Cloninger ACLONINGER@UCSD.EDU
Department of Mathematics, UCSD

Rahul Parhi RAHUL@UCSD.EDU
Department of Electrical and Computer Engineering, UCSD

Yu-Xiang Wang YUXIANGW@UCSD.EDU
Halıcıoğlu Data Science Institute, UCSD

Abstract

Gradient Descent (GD) with large learning rates often operates in the “Edge of Stability” (EoS) regime, where the sharpness of the loss landscape is implicitly constrained. However, the mechanism by which this stability-induced regularity translates into generalization remains elusive, particularly as neural networks can fit random noise under the same regime. In this work, we demonstrate that the implicit regularization enforced by EoS is *data-dependent and highly inhomogeneous*: it is stringent in regions where data concentrates but negligible in low-density regions. Consequently, the effective model capacity is determined by data geometry. We prove two complementary results: (1) For data supported on a mixture of low-dimensional subspaces, EoS dynamics yield generalization rates dependent on the *intrinsic dimension* rather than the ambient dimension. (2) Conversely, for data distributed on a high-dimensional sphere, we prove the existence of “flat” interpolating solutions that satisfy the stability constraint yet exhibit memorization. Our analysis establishes that stability alone is insufficient for generalization and its success depends on favorable data geometry.

1. Introduction

How does gradient descent (GD) discover well-generalizing representations in overparameterized neural networks, when these models possess sufficient capacity to memorize training data? While conventional wisdom attributes generalization to explicit regularization, empirical findings show that networks generalize well without it, yet can also fit random labels [44]. This paradox suggests that effective capacity is implicitly constrained by the optimizer’s preferences. A powerful proxy for this *implicit regularization* is the “*Edge of Stability*” (EoS) regime [7], where GD operates at a critical state balanced by local loss curvature. Theoretical analyses confirm that this stability-induced curvature constraint restricts the solution space [28, 29].

Crucially, recent breakthroughs have established that for two-layer ReLU networks, this implicit regularization acts as a *data-dependent penalty* on the network’s complexity, formalized as a weighted path norm [24, 34]. The defining characteristic of this regularization is that it is *spatially inhomogeneous*: the penalty strength varies drastically across the input domain depending on the data concentration. Liang et al. [24] analyzed this phenomenon in the specific case of *isotropic* distributions (e.g., Uniform on $\mathbb{B}_1^d$), where the analysis reduces to a 1D radial profile. However, the interaction between this inhomogeneous regularity and more complex geometries remains unexplored.

Contributions and Related Work. In this work, we investigate how this spatially inhomogeneous regularization interacts with more complex and degenerate data geometries. We propose that this non-uniformity creates a dichotomy in generalization performance.

1. **Adaptation (Better):** We prove that for data supported on a mixture of low-dimensional subspaces, the strong regularization in the data support effectively constrains the model complexity to the *intrinsic dimension*, significantly improving upon ambient-dimension bounds.

2. **Memorization (Worse):** Conversely, for data supported on the high-dimensional sphere, the data falls entirely into regions where regularization is negligible. We prove this allows for stable memorization of random noise.

Table 1: Comparison of our results with relevant prior works on path-norm and EoS generalization.

Work	Assumption	Rate	Context / Contribution
Parhi & Nowak[31]	**Unweighted** path norm is bounded (Static, no optimization dynamics)	$\tilde{O}(n^{-\frac{d+3}{4d+6}})$	Serves as a near-optimal baseline for unweighted path-norm constrain. Do not involve GD.
Qiao et al. [34]	BEoS dynamics, **univariate** input	$\tilde{O}(n^{-\frac{2}{5}})$	Used the technique of "chopping off the bad region" for 1D data. Only considering good regions.
Liang et al. [24]	BEoS dynamics, multivariate input **uniform** on the ball $\mathbb{B}_1^d$	$\tilde{O}(n^{-\frac{1}{2d+4}})$	Analyzes the "neural shattering" phenomenon for isotropic distributions to deduce upper/lower bounds.
Ours (Thm. 3)	BEoS dynamics, input on a mixture of m-**dim** balls ($m \leq d$)	$\tilde{O}(n^{-\frac{1}{2m+4}})$	**Provable adaptation to intrinsic dimension.** Our rate depends on m, not d, improving upon Liang et al. [24] for structured data.
Ours (Thm. 4)	BEoS dynamics, input on the **Sphere** $\mathbb{S}^{d-1}$	$\tilde{\Omega}(1)$	**Flat Interpolation.** For any dataset with inputs on $\mathbb{S}^{d-1}$, we construct interpolating network satisfying the BEoS condition.

Technical Novelty. A major technical hurdle is that standard uniform convergence bounds (e.g., global Rademacher complexity) are inapplicable in the EoS regime. Indeed, our result on spherical interpolation demonstrates that nontrivial **distribution-agnostic generalization bounds** are impossible, as the function class allows for arbitrary complexity on the sphere. To resolve this, we adopt a **divide-and-conquer** strategy that enforces **local complexity control** over the input domain. By decoupling the analysis into local subspaces and exploiting the projection properties of ReLU networks, we derive tight bounds that adapt to the local geometry where the data actually concentrates, bypassing the infinite capacity of the "bad" regions.

Scope of analysis. Our theoretical framework is situated in the feature-learning (or "rich") regime of overparameterized networks. Rather than tracking the full gradient dynamics[1], we focus on the regime in which gradient descent with a large step size operates for long stretches below around the edge-of-stability boundary. We therefore analyze the generalization behavior of all parameter vectors satisfying a Below-Edge-of-Stability (BEoS, see Definition 1), without assuming optimality or stationarity. Our bounds hold uniformly over this BEoS region and characterize a baseline form of implicit regularization that is enforced whenever training remains near the edge of stability.

2. Preliminaries and Notations

Neural network, data, and loss. We consider two-layer ReLU networks

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k\, \phi(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k) + \beta, \quad \phi(z) = \max\{z, 0\}, \tag{1}$$

with parameters $\boldsymbol{\theta} = \{(v_k, \boldsymbol{w}_k, b_k)\}_{k=1}^K \cup \{\beta\} \in \mathbb{R}^{(d+2)K+1}$. Let $\boldsymbol{\Theta}$ be the parameter set of such $\boldsymbol{\theta}$ for arbitrary $K \in \mathbb{N}$. We also assume $\boldsymbol{w}_k \neq \mathbf{0}$ for all k in this form, otherwise we may absorb it into the output bias β. Given data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i$ in a bounded domain $\Omega \subset \mathbb{R}^d$ with $d > 1$, the training loss is $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^n \big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\big)^2$. We assume $\|\boldsymbol{x}_i\| \le R$ and $|y_i| \le D, \forall i$.

"Edge of Stability" regime. Empirical and theoretical research [7, 9] has established the critical role of the linear stability threshold in the dynamics of gradient descent. In GD's trajectory, there is an initial phase of "progressive sharpening" where $\lambda_{\max}(\nabla^2\mathcal{L}(\boldsymbol{\theta}_t))$ increases. This continues until the GD process approaches the "Edge of Stability", a state where $\lambda_{\max}(\nabla^2\mathcal{L}(\theta_t)) \approx 2/\eta$, where η is the learning rate. In this paper, all the GD refers to vanilla GD with learning rate η.

Definition 1 (Below Edge of Stability [34, Definition 2.3]) *We define the trajectory of parameters $\{\boldsymbol{\theta}_t\}_{t=1,2,\cdots}$ generated by gradient descent with a learning rate η as* Below-Edge-of-Stability (BEoS) *if there exists a time $t^* > 0$ such that for all $t \ge t^*$, $\lambda_{\max}(\nabla^2\mathcal{L}(\boldsymbol{\theta}_t)) \le \frac{2}{\eta}$. Any parameter state $\boldsymbol{\theta}_t$ with $t \ge t^*$ is thereby referred to as a BEoS solution.*

This condition applies to any twice-differentiable solution found by GD, even when the optimization process does not converge to a local or global minimum. Moreover, BEoS is empirically verified to hold during both the "progressive sharpening" phase and the oscillatory phase.

Our work aims to analyze the generalization properties of any solutions that satisfy the BEoS condition (Definition 1). The set of solutions defined as:

$$\Theta_{\mathrm{BEoS}}(\eta, \mathcal{D}) := \left\{\boldsymbol{\theta} \;\middle|\; \lambda_{\max}(\nabla^2\mathcal{L}(\boldsymbol{\theta})) \le \frac{2}{\eta}\right\}. \tag{2}$$

Data-dependent weighted path norm. Given a weight function $g : \mathbb{S}^{d-1} \times \mathbb{R} \to \mathbb{R}$, where $\mathbb{S}^{d-1} := \{\boldsymbol{u} \in \mathbb{R}^d \;:\; \|\boldsymbol{u}\| = 1\}$, the *g-weighted path norm* of a neural network $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^K v_k\phi(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k) + \beta$ is defined to be

$$\|f_{\boldsymbol{\theta}}\|_{\mathrm{path},g} = \sum_{k=1}^{K} |v_k|\|\boldsymbol{w}_k\|_2 \cdot g\left(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|_2}, \frac{b_k}{\|\boldsymbol{w}_k\|_2}\right). \tag{3}$$

[1] Nevertheless, there is informal and heuristic discussion from the perspective of gradient dynamics in Appendix B.

The link between the EoS regime and weighted path norm constraint is presented in the following data-dependent weight function [24, 28, 29]. Fix a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$, we consider a weight function $g_{\mathcal{D}} : \mathbb{S}^{d-1} \times \mathbb{R} \to \mathbb{R}$ defined by $g_{\mathcal{D}}(\boldsymbol{u}, t) := \min\{\tilde{g}_{\mathcal{D}}(\boldsymbol{u}, t), \tilde{g}_{\mathcal{D}}(-\boldsymbol{u}, -t)\}$, where

$$\tilde{g}_{\mathcal{D}}(\boldsymbol{u}, t) := \mathbb{P}(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} > t)^2 \cdot \mathbb{E}[\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t \mid \boldsymbol{X}^\mathsf{T}\boldsymbol{u} > t] \cdot \sqrt{1 + \|\mathbb{E}[\boldsymbol{X} \mid \boldsymbol{X}^\mathsf{T}\boldsymbol{u} > t]\|^2}. \tag{4}$$

Here, $\boldsymbol{X}$ is a random vector drawn uniformly at random from the training examples $\{\boldsymbol{x}_i\}_{i=1}^n$. Specifically, we may also consider its population level $g_{\mathcal{P}}$ by viewing $\boldsymbol{X}$ as a random variable.

Proposition 2 *For any $\boldsymbol{\theta} \in \Theta_{\mathrm{BEoS}}(\eta, \mathcal{D})$, $\|f_{\boldsymbol{\theta}}\|_{\mathrm{path}, g_{\mathcal{D}}} \le \frac{1}{\eta} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}$.*

The proof of this proposition refers to [24, Corollary 3.3]. The non-parametric version of the weighted path norm constrain can be found in [24, 29].

Supervised statistical learning and generalization gap. We consider a supervised learning problem where i.i.d. samples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are drawn from an unknown distribution $\mathcal{P}$. In this paper, we assume the feature space is a compact subset of Euclidean space, $\Omega \subset \mathbb{R}^d$, the label space is $\mathbb{R}$, and the data is supported on $\Omega \times [-D, D]$. We use the squared loss, defined as $\ell(f, \boldsymbol{x}, y) = \frac{1}{2}(f(\boldsymbol{x}) - y)^2$. The performance of a predictor f is measured by its population risk $R_{\mathcal{P}}(f) = \mathbb{E}_{(\boldsymbol{X}, Y)\sim\mathcal{P}}\, \ell(f, \boldsymbol{X}, Y)$, while we optimize the empirical risk $\widehat{R}_{\mathcal{D}}(f) = \frac{1}{|\mathcal{D}|}\sum_{(\boldsymbol{x}_i, y_i)\in\mathcal{D}} \ell(f, \boldsymbol{x}_i, y_i)$. The difference between these two quantities is the generalization gap $\mathrm{Gap}_{\mathcal{P}}(f; \mathcal{D}) = |R_{\mathcal{P}}(f) - \widehat{R}_{\mathcal{D}}(f)|$. Our work focuses on the hypothesis classes the BEoS class $\Theta_{\mathrm{BEoS}}(\eta, \mathcal{D})$ and the bounded weighted-path norm class $\boldsymbol{\Theta}_g(\Omega; M, C)$,

$$\boldsymbol{\Theta}_g(\Omega; M, C) = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid \|f_{\boldsymbol{\theta}}\mid_{\Omega}\|_{L^\infty} \le M,\ \|f_{\boldsymbol{\theta}}\|_{\mathrm{path}, g} \le C\}\,. \tag{5}$$

where g can be the weight function $g_{\mathcal{D}}$ associated to the empirical distribution $\mathcal{D}$ or the weight function $g_{\mathcal{P}}$ associated to the population distribution $\mathcal{P}$, see Section E for more details.

3. Main Results

This section instantiates the principle of inhomogeneous regularization established in the previous section. We examine how the interplay between the stability-induced constraint and data geometry dictates generalization performance within two contrasting regimes: one where the geometry forces the model to adapt to intrinsic low-dimensional structures (Section 3.1), and another where the degenerate geometry allows the model to bypass regularization and memorize noise (Section 3.2).

3.1. Provable Adaptation to Intrinsic Low-Dimensionality

We first consider the scenario where data possesses favorable geometry, modeled as a union of low-dimensional subspaces. This setting is crucial for modeling multi-modal data where distinct clusters can be approximated by low-dimensional linear structures.

Theorem 3 (Generalization Bound for Mixture Models) *Given a data distribution $\mathcal{P}$ on $\mathbb{R}^d \times \mathbb{R}$ whose feature marginal satisfies $\mathcal{P}_X = \sum_{j=1}^J \pi_j \mathcal{P}_{X,j}$, where each $\mathcal{P}_{X,j}$ is the uniform distribution on the unit ball in an m-dimensional affine subspace $V_j \subset \mathbb{R}^d$, let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset of n i.i.d. samples drawn from $\mathcal{P}$. Then, with probability at least $1 - \delta$,*

$$\sup_{\boldsymbol{\theta}\in\Theta_{\mathrm{BEoS}}(\eta, \mathcal{D})} \mathrm{Gap}_{\mathcal{P}}(f_{\boldsymbol{\theta}}; \mathcal{D}) \ \lesssim_d\ \tilde{O}\left(n^{-\frac{1}{2m+4}}\right). \tag{6}$$

The notation $\lesssim_d$ hides constants depending on d, but the exponent depends solely on m.

Sketch of Proof. The proof (detailed in Appendix F) relies on a divide-and-conquer strategy that exploits the geometric structure of the stability constraint. We first define local weight functions g_j associated with each component distribution $\mathcal{P}_{X,j}$. We prove that the global weight function g (induced by the full mixture) uniformly dominates these local weights, i.e., $g(\boldsymbol{u}, t) \gtrsim g_j(\boldsymbol{u}, t)$. This ensures that any solution satisfying the global stability constraint automatically inherits strict regularity conditions on each local subspace. On any specific m-dimensional subspace V_j, the activation of a neuron with parameters $(\boldsymbol{w}_k, b_k)$ is determined solely by the projection of $\boldsymbol{w}_k$ onto V_j. The component of the weight vector orthogonal to V_j acts merely as a bias shift and is "invisible" to the data distribution on that component. Consequently, the weighted path norm constraint effectively penalizes only the projected weights. This reduces the effective capacity analysis from the ambient dimension d to the intrinsic dimension m, yielding the improved rate.

3.2. The Limits of Stability: Flat Interpolation on the Sphere

To demonstrate that EoS stability is not a sufficient condition for generalization, we examine a geometry that lacks the "dense interior" property utilized in the previous section: the uniform distribution on the high-dimensional sphere.

Theorem 4 (Flat interpolation with width $\leq n$) *Assume that $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is a dataset with $\boldsymbol{x}_i \in \mathbb{S}^{d-1}$ and pairwise distinct inputs. Then there exists a width $K \leq n$ network of the form (1) that interpolates the dataset and whose Hessian operator norm satisfies*

$$\lambda_{\max}\left(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}\right) \;\leq\; 1 + \frac{D^2+2}{n}. \tag{7}$$

If we remove the output bias parameter β in (1), then $\lambda_{\max}\left(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}\right) \leq \frac{D^2+2}{n}$.

The construction of this kinds of networks can be found in Appendix G.

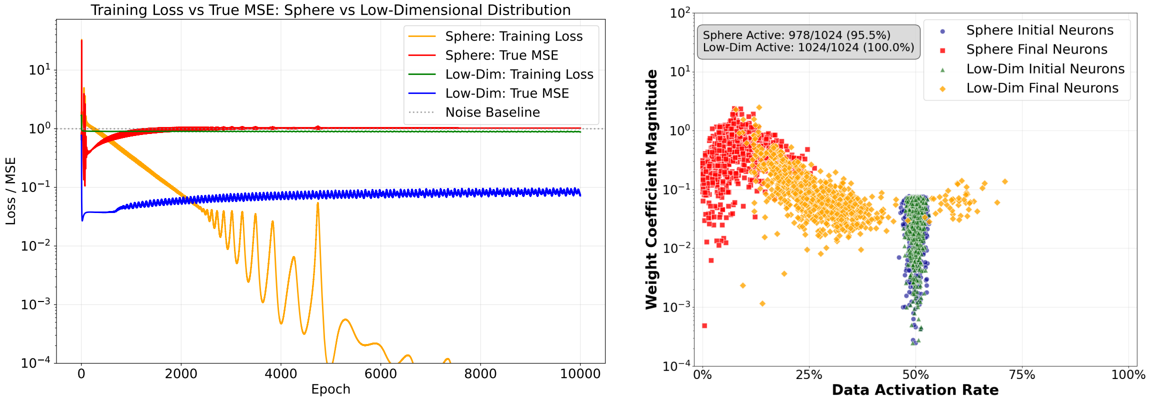
Remark 5 (Comparison to Wu and Su [42]) *Wu and Su [42] established that for **bias-free** networks on the sphere, stable global minima enjoy an $O(1/n)$ generalization rate. Crucially, this result relies on a specific alignment between architecture and data geometry: the bias-free constraint forces every activation boundary to pass through the origin. Due to the point-symmetry of the spherical distribution, this geometric rigidity ensures that every neuron activates approximately half the probability mass, causing the EoS constraint to degenerate into a uniform path-norm bound. Our work explicitly models this dependency by including hidden biases, which relax this geometric constraint. This allows activation boundaries to form arbitrary hyperplanes that can easily isolate individual data points within small spherical caps, thereby bypassing the stability penalty to enable memorization.*

4. Experiments

We provide empirical verification that data geometry fundamentally alters the representations learned by GD. More detials can be found in Appendix C.

Setup. We train two-layer ReLU networks ($K = 1000$) with MSE loss and vanilla GD ($\eta = 0.4$) for 20,000 epochs. We use two synthetic data distributions in $\mathbb{R}^{50}$: (1) **Sphere:** Uniform distribution on the unit sphere. (2) **Low-Dimensional Mixture:** A union of 20 random 1D lines ($m = 1$). Labels are generated by a fixed teacher network with added Gaussian noise to test resistance to overfitting.

Analysis of Activation Rates. We monitor the *data activation rate* of neurons, defined as the fraction of the training set a neuron activates on: $\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}\{\boldsymbol{w}_k^\top \boldsymbol{x}_i > b_k\}$.



(a) Training curves on different geometries.

(b) Weight magnitude vs. activation rate

Figure 1: **Generalization and Learned Representations under Different Geometries.** (a) Training loss and clean MSE on Sphere vs. Low-dimensional mixture. We can see GD on sphere interpolate very quickly (before the 2000-th epoch) while the mixed low-dimensional data resist to overfitting. (b) Scatter plot of weight magnitude vs. activation rate. On the sphere, we observe high-magnitude weights associated with low activation rates, consistent with the memorization construction in Theorem 4.

The results in Figure 1 corroborate our theoretical dichotomy. On the low-dimensional mixture, neurons are forced to learn shared features (high activation rate) to satisfy stability. On the sphere, the network utilizes "lazy" neurons (low activation rate) to memorize specific data points, as the stability penalty for such neurons is minimal.

5. Discussion and Conclusion

This work demonstrates that generalization at the Edge of Stability depends on data geometry. The stability constraint creates a regularization that is spatially inhomogeneous. It effectively penalizes a neuron based on the probability mass it activates. We prove that for data on low-dimensional subspaces, the geometry forces neurons to activate significant mass. This enforces strict complexity control and leads to adaptation to the intrinsic dimension. However, on the high-dimensional sphere, neurons can isolate individual data points while activating negligible mass. This allows the network to bypass the regularization and stably memorize noise. Therefore, dynamical stability alone does not guarantee generalization.

References

[1] Kwangjun Ahn, Jingzhao Zhang, and Suvrit Sra. Understanding the unstable convergence of gradient descent. In *International conference on machine learning*, pages 247–257. PMLR, 2022.

[2] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

[3] Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024. PMLR, 2022.

[4] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

[5] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[6] Alexander Cloninger and Timo Klock. A deep network construction that adapts to intrinsic dimensionality beyond the domain. *Neural Networks*, 141:404–419, 2021.

[7] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2020.

[8] Jeremy M. Cohen, Alex Damian, Ameet Talwalkar, J. Zico Kolter, and Jason D. Lee. Understanding optimization in deep learning with central flows. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sIE2rI3ZPs.

[9] Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *International Conference on Learning Representations*, 2024.

[10] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[11] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[12] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[13] Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:59291990.

[14] Boris Hanin and David Rolnick. Deep ReLU networks have surprisingly few activation patterns. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[15] Boris Hanin, Ryan Jeong, and David Rolnick. Deep relu networks preserve expected length. *ArXiv*, abs/2102.10492, 2021. URL https://api.semanticscholar.org/CorpusID:231986303.

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

[17] Hui Jin and Guido Montúfar. Implicit bias of gradient descent for mean squared error regression with two-layer wide neural networks. *Journal of Machine Learning Research*, 24(137):1–97, 2023.

[18] Nirmit Joshi, Gal Vardi, and Nathan Srebro. Noisy interpolation learning with shallow univariate ReLU networks. In *International Conference on Learning Representations*, 2024.

[19] Michael Kohler and Sophie Langer. On the rate of convergence of fully connected deep neural network regression estimates. *Annals of Statistics*, 49(4):2231–2249, 2021.

[20] Michael Kohler, Adam Krzyżak, and Sophie Langer. Estimation of a function of low local dimensionality by deep neural networks. *IEEE transactions on information theory*, 68(6): 4032–4042, 2022.

[21] Lingkai Kong and Molei Tao. Stochasticity of deterministic gradient descent: Large learning rate for multiscale objective function. *Advances in neural information processing systems*, 33: 2625–2638, 2020.

[22] Guy Kornowski, Gilad Yehudai, and Ohad Shamir. From tempered to benign overfitting in relu neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 888–896, Naha, Okinawa, Japan, 2019. PMLR. URL https://proceedings.mlr.press/v89/liang19a.html.

[24] Tongtong Liang, Dan Qiao, Yu-Xiang Wang, and Rahul Parhi. Stable minima of relu neural networks suffer from the curse of dimensionality: The neural shattering phenomenon, 2025. URL https://arxiv.org/abs/2506.20779.

[25] Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. *Advances in Neural Information Processing Systems*, 34:16805–16817, 2021.

[26] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pages 2388–2464. PMLR, 2019.

[27] Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Neural Information Processing Systems*, 2014. URL https://api.semanticscholar.org/CorpusID:5941770.

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

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

[30] Rahul Parhi and Robert D. Nowak. Banach space representer theorems for neural networks and ridge splines. *Journal of Machine Learning Research*, 22(43):1–40, 2021.

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

[32] Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *Advances in Neural Information Processing Systems 34*, volume 34. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/995f5e03890b029865f402e83a81c29d-Abstract.html.

[33] Tomaso Poggio and Qianli Liao. Theory ii: Landscape of the empirical risk in deep learning. *arXiv preprint arXiv:1703.09833*, 2017.

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

[35] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, 48(4):1875–1897, 2020.

[36] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. *ArXiv*, abs/1711.02114, 2017. URL https://api.semanticscholar.org/CorpusID:34019680.

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

[38] Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. *arXiv preprint arXiv:1810.08033*, 2018.

[39] Saket Tiwari and George Konidaris. Effects of data geometry in early deep learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 25010–25023, 2022.

[40] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[41] René Vidal, Yi Ma, and Shankar Sastry. *Generalized Principal Component Analysis*. Springer, 2016.

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

[43] Yunfei Yang and Ding-Xuan Zhou. Optimal rates of approximation by shallow ReLU^k neural networks and applications to nonparametric regression. *Constructive Approximation*, pages 1–32, 2024.

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

[45] Kaiqi Zhang and Yu-Xiang Wang. Deep learning meets nonparametric regression: Are weight-decayed DNNs locally adaptive? In *International Conference on Learning Representations*, 2023.

[46] Xiao Zhang and Dongrui Wu. Empirical studies on the properties of linear regions in deep neural networks. *ArXiv*, abs/2001.01072, 2020. URL https://api.semanticscholar.org/CorpusID:209862298.

Appendix A. More Related Works

Subspace and manifold hypothesis. A common modeling assumption in high-dimensional learning is that data lies on or near one or several low-dimensional subspaces embedded in the ambient space, especially in image datasets where pixel values are constrained by geometric structure and are well-approximated by local subspaces or unions of subspaces [41]. In particular, results in sparse representation and subspace clustering demonstrate that such structures enable efficient recovery and segmentation of high-dimensional data into their intrinsic subspaces [11]. This also extends to a more general framework of the manifold hypothesis [12].

Capacity of neural networks. The subspace and manifold hypotheses have important implications for the capacity and generalization of neural networks. When data lies near low-dimensional subspaces and manifolds, networks can achieve expressive power with significantly fewer parameters, as the complexity of the function to be learned is effectively constrained by the subspace dimension rather than the ambient dimension [6, 20, 33]. However, these results focus only on expressivity and the existence of neural networks to learn efficiently on this data.

Interpolation, Benign overfitting and data geometry. Benign-overfitting [5] studies the curious phenomenon that one can interpolate noisy labels (i.e., 0 training loss) while consistently learn (excess risk → 0 as n gets larger). Joshi et al. [18] establishes that overfitting in ReLU Networks is not benign in general, but it could become more benign as the input dimension grows [22] in the isotropic Gaussian data case. Our results suggest that such conclusion may be fragile under *low-dimensional or structured* input distributions. On a positive note, our results suggest that in these cases, generalization may follow from edge-of-stability, which applies without requiring interpolation.

Implicit bias of gradient descent. A rich line of work analyzes the implicit bias of (stochastic) gradient descent (GD), typically through optimization dynamics or limiting kernels [2, 17, 26]. In contrast, we do not analyze the time evolution per se; we characterize the *function spaces* that GD tends to realize at solutions. Our results highlight a strong dependence on the *input distribution*: even for the same architecture and loss, the induced hypothesis class (and thus generalization) changes as the data geometry changes, complementing prior dynamics-centric views.

Edge of Stability (EoS) and minima stability. The EoS literature primarily seeks to explain when and why training operates near instability and how optimization proceeds there [1, 3, 7, 9, 21]. Central flows offer an alternative viewpoint on optimization trajectories that also emphasizes near-instability behavior [8]. Closest to our work is the line on *minima stability* [25, 28, 29], which links Hessian spectra and training noise to the geometry of solutions but largely leaves generalization out of scope. We leverage the EoS/minima-stability phenomena to *define* and analyze a data-distribution-aware notion of stability, showing adaptivity to low-dimensional structure and making explicit how distributional geometry shapes which stable minima GD selects.

Nonparametric function estimation with neural networks. The notion of generalization gap is closely related to the estimation error. It is well known that neural networks are minimax optimal estimators for a wide variety of functions [19, 31, 34, 35, 38, 42, 43, 45]. Outside of the univariate work of Qiao et al. [34], all prior works construct their estimators via empirical risk minimization problems. Thus, they do not incorporate the training dynamics that arise when training neural networks in practice. In contrast, Liang et al. [24], Qiao et al. [34] derive nonparametric guarantees for solutions selected directly by gradient descent dynamics, without assuming even stationary condition. Our work further develops in this direction by modeling how data geometry affect generalization for the gradient trajectories below the edga-of-stability.

Flatness vs. generalization. Whether (and which notion of) flatness predicts generalization remains debated. Several works argue sharp minima can still generalize [10], propose information-geometric or Fisher-Rao-based notions [23], or develop relative/scale-invariant flatness measures [32]. We focus on the *largest* curvature direction (i.e., $\lambda_{\max}$) motivated by EoS/minima-stability. Our results rigorously prove that flatness in this notion does imply generalization (note that there is no contradiction with Dinh et al. [10]), but it depends on data distribution.

Linear regions of neural networks. Our research connects to a significant body of work that investigates the shattering capability of neural networks by quantifying their linear activation regions [13–15, 27, 36]. Other empirical work has meticulously characterized the geometric properties of linear regions shaped by different optimizers [46]. Particularly, [39] consider the how these linear regions intersect with data manifolds. These analyses primarily leverage the number of regions to characterize the expressive power of deep networks, while our work shifts the focus on the generalization performance of shallow networks at the EoS regime.

Appendix B. A Gradient-dynamics Perspective on Stability and Data Geometry

We now discuss our results from the viewpoint of *gradient dynamics*.

Why Shattering Neural Networks May Be Dynamically Stable. Here we paraphrase and adapt the discussion from [24, Appendix A.4]. The BEoS condition defines a set of dynamically stable parameter states, and the data-dependent weight function $g_{\mathcal{D}}(\boldsymbol{u}, t)$ provides a static summary of how expensive it is to place a ReLU ridge at orientation $\boldsymbol{u}$ and threshold t. Small values of $g_{\mathcal{D}}(\boldsymbol{u}, t)$ indicate weak stability constraints in that region of parameter space, so a neuron aligned with $(\boldsymbol{u}, t)$ can carry a large coefficient while still satisfying the BEoS curvature bound. In highly shatterable geometries, the shallow shell contains many such directions with tiny $g_{\mathcal{D}}$, creating ample room inside the stable set for high-magnitude, sparsely activating neurons supported on disjoint caps.

This static picture is closely tied to the actual gradient dynamics. If the dataset is highly shatterable in the sense of the half-space-depth concentration index, a neuron's activation boundary can easily drift toward regions where it fires on only a few data points. Once those few points are already well-fitted, the gradient contributions in (8) become small and localized, so the neuron experiences almost no force pulling it back toward more central regions of the data cloud. Its parameters become effectively "stuck" near the boundary, and the corresponding directions in the loss landscape remain flat enough to satisfy the BEoS condition. Our lower bound constructions exploit exactly this mechanism: by arranging many such trapped, boundary-supported neurons on disjoint caps, we obtain shattering networks that interpolate, remain dynamically stable, and yet are statistically hard to learn. Although our analysis is carried out for ReLU, where hard sparsity makes this effect particularly transparent, the underlying "weak-gradient trapping" mechanism suggests that similar phenomena may persist for other activations with rapidly decaying gradients away from their transition region.

How GD Adapts to Low-dimensionality Dynamically. Recall our notations: for a two-layer ReLU network

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \, \phi(\boldsymbol{w}_k^{\mathsf{T}} \boldsymbol{x} - b_k) + \beta$$

trained on data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with empirical loss $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^n \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$, the gradient with respect to a hidden weight $\boldsymbol{w}_k$ has the form

$$\nabla_{\boldsymbol{w}_k}\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell'\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i\big)\, v_k\, \phi'(\boldsymbol{w}_k^{\mathsf{T}}\boldsymbol{x}_i - b_k)\, \boldsymbol{x}_i = \sum_{i=1}^n \alpha_{k,i}(\boldsymbol{\theta})\, \boldsymbol{x}_i. \tag{8}$$

Thus, at every step of gradient descent, the update of $\boldsymbol{w}_k$ is a linear combination of input vectors. In informal terms, the *shape* of the gradient field is inherited from the shape of the data cloud: the dynamics cannot move in arbitrary directions in parameter space, but only along directions induced by the training inputs and the neuron-specific activation pattern.

In the idealized case where all $\boldsymbol{x}_i$ lie in a linear subspace $V \subset \mathbb{R}^d$, (8) already suggests a form of intrinsic-dimension adaptation: the trajectory of each hidden weight lives in an affine translate of V, so any meaningful notion of effective complexity should depend on $\dim V$ rather than on the ambient dimension. However, as soon as we move to more realistic geometries—affine subspaces $a + V$ or mixtures of m-dimensional components—the global span of the data can easily be full-dimensional. In those settings, the observation that gradients lie in $\operatorname{span}\{\boldsymbol{x}_i\}$ is nearly vacuous: it no longer encodes the *structured* way in which the data constrain the dynamics, and by itself it does not yield intrinsic-dimension generalization bounds.

Our contribution is to make this stability-based picture interact explicitly with the *geometry of the data*. For structured distributions such as mixtures of m-dimensional balls, the data-dependent weight function $g_{\mathcal{D}}(\boldsymbol{u}, t)$ inherits a corresponding structure. A hyperplane that cuts through the thick interior of an m-dimensional component activates on many of its points. Neurons aligned with such hyperplanes receive large gradients and are strongly constrained by the stability condition, which is reflected in large values of $g_{\mathcal{D}}(\boldsymbol{u}, t)$. In contrast, hyperplanes that only touch shallow caps or skim the boundary fire on very few points; neurons aligned with these directions see much weaker "gradient pressure" and correspondingly small values of $g_{\mathcal{D}}(\boldsymbol{u}, t)$, allowing them to carry high norm and implement localized features.

In this way, the gradient-dynamics perspective can be summarized as

$$\text{data geometry} \implies \text{gradient geometry} \implies \text{stability-induced regularization},$$

with $g_{\mathcal{D}}$ acting as the bridge between dynamics and capacity control. Stability does not merely say that gradients live in the span of the data; through $g_{\mathcal{D}}$, it tells us *which parts* of the data geometry are expensive to fit and *which parts* can host high-norm, sparsely supported neurons. This refinement allows our analysis to turn the qualitative statement that "gradient trajectories are shaped by the data geometry" into quantitative, distribution-dependent generalization bounds that scale with the intrinsic dimension m rather than the ambient dimension d.

Appendix C. Details of Experiments

Here we provide the full experimental details of the discussion in Section 4.

We worked in ambient dimension $d = 50$ with $n = 2000$ training examples. For the *Sphere* condition, samples were drawn uniformly from the unit sphere. For the *Low-dimensional mixture*, we generated data from a mixture of 20 randomly oriented 1-dimensional subspaces uniformly. Labels were produced by a fixed quadratic teacher function with added Gaussian noise of variance 1.

We trained a two-layer ReLU network with hidden width 1024. All models were trained with GD for 10000 epochs using learning rate 0.4 and gradient clipping at 50. The loss function was the squared error against noisy labels, while generalization performance was evaluated by the *true MSE* against the noiseless teacher. For comparability, both datasets shared the same initialization of parameters.

We monitored (i) training loss and true MSE, (ii) Hessian spectral norm estimated by power iteration on random minibatches, and (iii) neuron-level statistics such as activation rate and coefficient magnitude. The training curves are shown in Figure 2 and $\lambda_{\max}(\nabla_{\boldsymbol{\theta}}\mathcal{L})$-curves are shown in Figure 4.

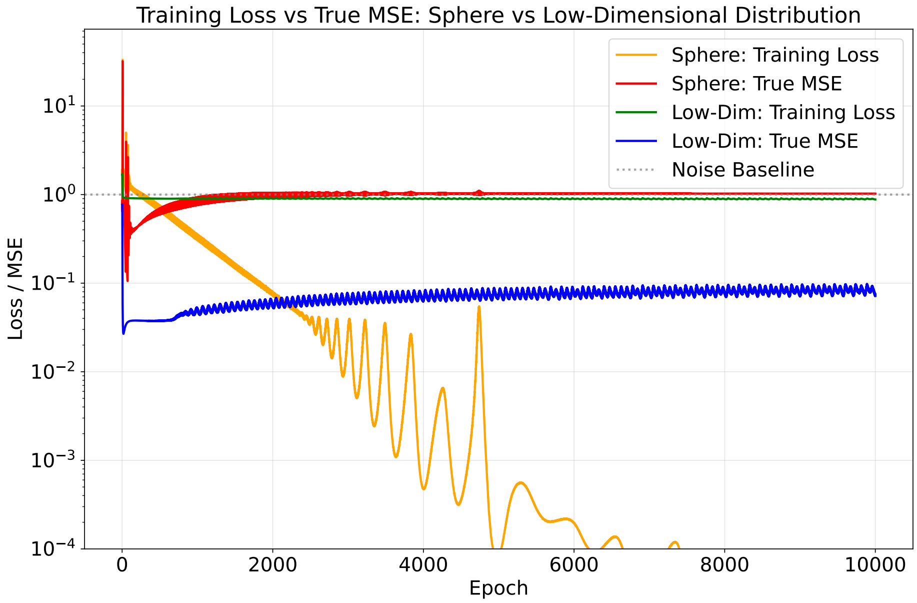


Figure 2: **Training curves on different geometries.** Training loss and clean MSE on Sphere vs. Low-dimensional mixture. We can see GD on sphere interpolate very quickly (before the 2000-th epoch) while the mixed low-dimensional data resist to overfitting.

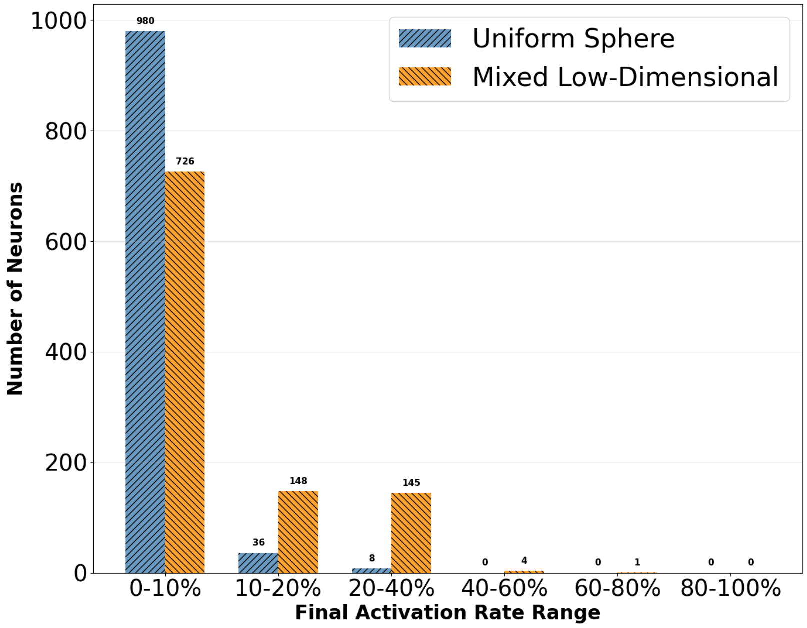


Figure 3: **Neuron activation statistics under different geometries.** On the uniform sphere, most neurons fire on less than 10% of the data, indicating highly specialized ReLUs as we predict in Theorem 4. On the low-dimensional mixture, many neurons fire on 10-40% of the data.

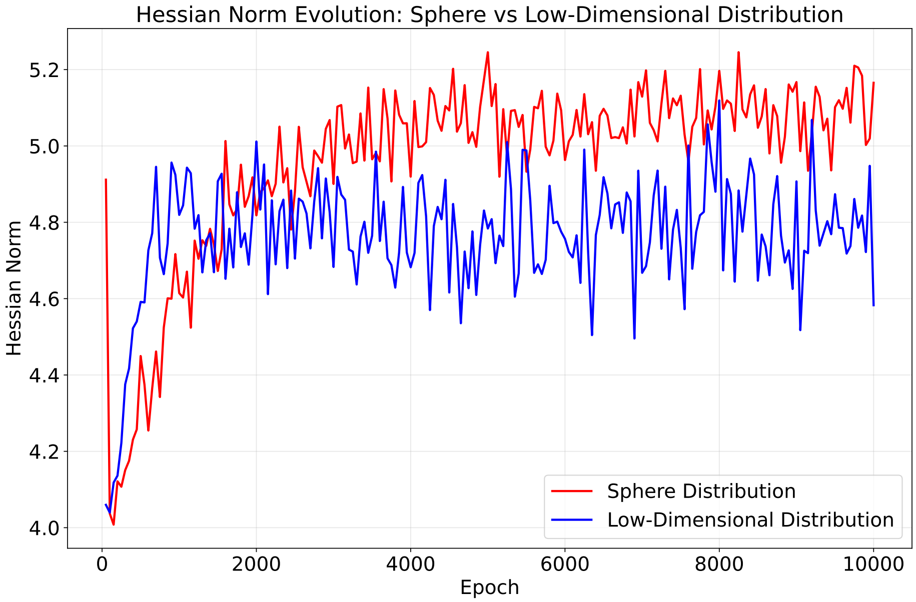


Figure 4: $\lambda_{\max}(\nabla_{\boldsymbol{\theta}}\mathcal{L})$**-curves.** Both of the curves oscillates around $2/\eta = 5$, signaling the edge of stability regime.

Appendix D. Functional Analysis of Shallow ReLU Networks

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

In this section, we summarize some result in [31] and [37].

Definition 6 *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 two-layer neural network. The (unweighted) path-norm of $f_{\boldsymbol{\theta}}$ is defined to be*

$$\|f_{\boldsymbol{\theta}}\|_{\text{path}} := \sum_{k=1}^{K} |v_k|\, \|\boldsymbol{w}_k\|_2 \,. \tag{9}$$

Dictionary representation of ReLU networks. By the positive 1-homogeneity of ReLU, each neuron can be rescaled without changing the realized function:

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

Hence f_θ admits the normalized finite-sum form

$$f(\boldsymbol{x}) = \sum_{k=1}^{K'} a_k\, \phi(\boldsymbol{u}_k^{\mathsf{T}} \boldsymbol{x} - t_k) + \boldsymbol{c}^{\mathsf{T}} \boldsymbol{x} + c_0. \tag{10}$$

Let the (ReLU) ridge dictionary be $\mathscr{D}_\phi := \left\{ \phi(\boldsymbol{u}^{\mathsf{T}} \cdot - t) : \ \boldsymbol{u} \in \mathbb{S}^{d-1},\ t \in \mathbb{R} \right\}$. We study the *overparameterized, width-agnostic* class given by the *union over all finite widths*

$$\mathcal{F}_{\text{fin}} := \bigcup_{K \geq 1} \left\{ \sum_{k=1}^{K} a_k\, \phi(\boldsymbol{u}_k^{\mathsf{T}} \cdot - t_k) + \boldsymbol{c}^{\mathsf{T}}(\cdot) + c_0 \right\}, \tag{11}$$

and measure complexity by the minimal path-norm needed to realize f:

$$\|f\|_{\text{path,min}} := \inf \left\{ \|f_{\boldsymbol{\theta}}\|_{\text{path}} : \ f_{\boldsymbol{\theta}} \equiv f \text{ of the form (10)} \right\}.$$

From finite sums to a *width-agnostic* integral representation. To analyze $\mathcal{F}_{\text{fin}}$ without committing to a fixed width K, we pass to a convex, measure-based description that *represents the closure/convex hull of* (11). Specifically, let ν be a finite signed Radon measure on $\mathbb{S}^{d-1} \times [-R, R]$ and consider

$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1} \times [-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}} \boldsymbol{x} - t)\ \mathrm{d}\nu(\boldsymbol{u}, t) + c^{\mathsf{T}} \boldsymbol{x} + c_0. \tag{12}$$

Any finite network (10) corresponds to the *sparse* measure $\nu = \sum_{k=1}^{K} a_k\, \delta_{(\boldsymbol{u}_k, t_k)}$, and conversely sparse measures yield finite networks. Thus, (12) is a *width-agnostic relaxation* of (9), not an assumption of an infinite-width limit.

Definition 7 *The (unweighted) variation (semi)norm*

$$|f|_{\mathrm{V}} := \inf \left\{ \|\nu\|_{\mathcal{M}} : \ f \text{ admits (10) for some } (\nu, c, c_0) \right\}, \tag{13}$$

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

$$\mathrm{V}_C(\Omega) := \left\{ f : \Omega \to \mathbb{R} \mid f = \int_{\mathbb{S}^{d-1} \times [-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}} \boldsymbol{x} - t)\, \mathrm{d}\nu(\boldsymbol{u}, t) + \boldsymbol{c}^{\mathsf{T}} \boldsymbol{x} + b,\ |f|_{\mathrm{V}} \leq C \right\}. \tag{14}$$

Specifically, by identifying (10) with the atomic measure $\nu = \sum_k a_k \delta_{(\boldsymbol{u}_k, t_k)}$, we have

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

Conversely, the smallest variation needed to represent f equals the smallest path-norm across all finite decompositions,

$$\|f\|_{\mathrm{path,min}} = |f|_{\mathrm{V}}. \tag{15}$$

Thus, the variation seminorm (13) is the *nonparametric* counterpart of the path-norm, which captures the same notion of complexity but *without fixing the width* K.

Remark 8 ("Arbitrary width" $\neq$ "infinite width") *Our analysis concerns $\mathcal{F}_{\mathrm{fin}}$ in* (11)*, i.e., the union over all* finite *widths. The integral model* (12) *is a convexification/closure of this union that facilitates analysis and regularization; it* does not *assume an infinite-width limit. In variational training with a total-variation penalty on ν, first-order optimality ensures sparse solutions (finite support of ν), which correspond to* finite-width *networks. Thus, all results in this paper apply to* arbitrary (but finite) width*, and the continuum measure is only a device to characterize and control $\|f\|_{\mathrm{path,min}}$.*

D.2. Total Variation Semi-norm on Radon Domain

We now connect the (unweighted) variation semi-norm of shallow ReLU networks to an analytic description on the *Radon domain*. Our presentation follows [30, 31].

Definition 9 *For a function $f : \mathbb{R}^d \to \mathbb{R}$ and $(\boldsymbol{u}, t) \in \mathbb{S}^{d-1} \times \mathbb{R} := \mathbb{S}^{d-1} \times \mathbb{R}$, the Radon transform and its dual are defined by*

$$\mathscr{R} f(\boldsymbol{u}, t) = \int_{\{\boldsymbol{x}:\, \boldsymbol{u}^\top \boldsymbol{x} = t\}} f(\boldsymbol{x})\, \mathrm{d}s(\boldsymbol{x})$$

$$\mathscr{R}^* \{\Phi\}(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}} \Phi\big(\boldsymbol{u},\, \boldsymbol{u}^\top \boldsymbol{x}\big)\, \mathrm{d}\sigma(\boldsymbol{u}).$$

The Radon framework encodes a function f by its integrals over affine hyperplanes faithfully in the senses that the Radon transform is invertible up to a known dimension-dependent constant via a one-dimensional "ramp" filter in t.

Proposition 10 (Filtered backprojection (Radon inversion)) *There exists $c_d > 0$ such that*

$$c_d\, f \;=\; \mathscr{R}^* \{\, \Lambda_{d-1} \mathscr{R} f \,\},$$

where Λ_{d-1} acts in the t-variable with Fourier symbol $\widehat{\Lambda_{d-1}\Phi}(\boldsymbol{u}, \omega) = i^{\,d-1}\, |\omega|^{\,d-1}\, \hat{\Phi}(\boldsymbol{u}, \omega)$.

The inversion formula motivates measuring the "ridge-curvature" of f by differentiating in the Radon offset t after filtering, and aggregating its magnitude over all orientations and offsets.

The next definition is the sole norm we need on the Radon domain; it specializes all higher-order variants to the ReLU case.

Definition 11 (Second-order Radon total variation (ReLU case)) *The (second-order) Radon total-variation seminorm is*

$$\mathscr{R}\mathrm{TV}^2(f) := \left\| \mathscr{R}\left\{(-\mathrm{Lap})^{\frac{d+1}{2}} f\right\} \right\|_{\mathcal{M}(\mathbb{S}^{d-1}\times\mathbb{R})},$$

where the fractional power is understood in the tempered-distribution sense. The null space of $\mathscr{R}\mathrm{TV}^2(\cdot)$ *is the set of affine functions on* $\mathbb{R}^d$.

Proposition 12 (Equivalence of seminorms on bounded domains [30]) *Let* $\mathcal{B} = \mathbb{B}^d_R$. *For any* $f : \mathcal{B} \to \mathbb{R}$ *with finite variation seminorm, its canonical extension* f_{ext} *to* $\mathbb{R}^d$ *satisfies*

$$|f|_{\mathrm{V}} \;=\; \mathscr{R}\mathrm{TV}^2(f_{\mathrm{ext}}),$$

and, in particular, for any finite two-layer ReLU network in reduced form $f_\theta(\boldsymbol{x}) = \sum_{k=1}^K v_k\, \phi(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k) + \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + c_0$,

$$\mathscr{R}\mathrm{TV}^2(f_{\boldsymbol{\theta}}) \;=\; \sum_{k=1}^K |v_k|\; \|\boldsymbol{w}_k\|_2\,,$$

which equals the minimal (unweighted) path-norm needed to realize f_θ *on* $\mathbb{B}^d_R$.

The key structural reason is simple: $\partial_t^2 \Lambda_{d-1}\mathscr{R}$ turns each ReLU ridge $\phi(\boldsymbol{u}^\mathsf{T}\boldsymbol{x} - t)$ into a Dirac mass at $(\boldsymbol{u}, t)$ on $\mathbb{S}^{d-1} \times \mathbb{R}$, so superpositions of ridges correspond exactly to finite signed measures on $\mathbb{S}^{d-1} \times \mathbb{R}$, and the total-variation of that measure coincides with both the variation seminorm and $\mathscr{R}\mathrm{TV}^2(\cdot)$ after fixing the affine null space.

Remark 13 (Takeaway) *For ReLU networks on bounded domains, the three viewpoints*

$$\textit{path-norm } \|f\|_{\mathrm{path}} \;\longleftrightarrow\; \textit{unweighted variation } |f|_{\mathrm{V}} \;\longleftrightarrow\; \textit{Radon-TV } \mathscr{R}\mathrm{TV}(f)$$

are equivalent up to the affine null space. We will freely switch between them in the sequel.

D.3. The Metric Entropy of Variation Spaces

Metric entropy quantifies the compactness of a set A in a metric space (X, ρ_X). Below we introduce the definition of covering numbers and metric entropy.

Definition 14 (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 minimum number of closed balls of radius t needed to cover* A*:*

$$N(t, A, \rho_X) := \min\left\{ N \in \mathbb{N} : \exists\, x_1, \ldots, x_N \in X \text{ s.t. } A \subset \bigcup_{i=1}^N \mathbb{B}(x_i, t) \right\}, \tag{16}$$

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 defined as:*

$$H_t(A)_X := \log N(t, A, \rho_X). \tag{17}$$

The metric entropy of the bounded variation function class has been studied in previous works. More specifically, we will directly use the one below in future analysis.

Proposition 15 (Parhi and Nowak 31, Appendix D) *The metric entropy of* $\mathrm{V}_C(\mathbb{B}_R^d)$ *(see Definition 7) with respect to the* $L^\infty(\mathbb{B}_R^d)$*-distance* $\|\cdot\|_\infty$ *satisfies*

$$\log N(t, \mathrm{V}_C(\mathbb{B}_R^d), \|\cdot\|_\infty) \lessapprox_d \left(\frac{C}{t}\right)^{\frac{2d}{d+3}}. \tag{18}$$

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

D.4. Generalization Gap of Unweighted Variation Function Class

As a middle step towards bounding the generalization gap of the weighted variation function class, we first bound the generalization gap of the unweighted variation function class according to a metric entropy analysis.

Lemma 16 *Let* $\mathcal{F}_{M,C} = \{f \in \mathrm{V}_C(\mathbb{B}_R^d) \mid \|f\|_\infty \le M\}$ *with* $M \ge D$*. Then let* $\mathcal{D} \sim \mathcal{P}^{\otimes n}$ *be a sampled data set of size* n*, with probability at least* $1-\delta$*,*

$$\sup_{f \in \mathcal{F}_{M,C}} \left|R(f) - \widehat{R}_{\mathcal{D}}(f)\right| \lesssim_d C^{\frac{d}{2d+3}} M^{\frac{3(d+2)}{2d+3}} n^{-\frac{d+3}{4d+6}} + M^2 \left(\frac{\log(4/\delta)}{n}\right)^{-\frac{1}{2}}. \tag{19}$$

Proof According to Proposition 15, one just needs $N(t)$ balls to cover $\mathcal{F}$ in $\|\cdot\|_\infty$ with radius $t > 0$ such that where

$$\log N(t) \lessapprox_d \left(\frac{C}{t}\right)^{\frac{2d}{d+3}}.$$

Then for any $f, g \in \mathcal{F}_{M,C}$ and any $(\boldsymbol{x}, y)$,

$$\left|(f(\boldsymbol{x})-y)^2 - (g(\boldsymbol{x})-y)^2\right| = |f(\boldsymbol{x}) - g(\boldsymbol{x})|\,|f(\boldsymbol{x}) + g(\boldsymbol{x}) - 2y| \le 4M\,\|f-g\|_\infty.$$

Hence replacing f by a centre f_i within t changes both the empirical and true risks by at most $4Mt$.

For any fixed centre $\bar{f}$ in the covering, Hoeffding's inequality implies that with probability at least $\ge 1-\delta$, we have

$$|R(\bar{f}) - \widehat{R}_{\mathcal{D}}(\bar{f})| \le 4M^2\sqrt{\frac{\log(2/\delta)}{n}} \tag{20}$$

because each squared error lies in $[0, 4M^2]$. Then we take all the centers with union bound to deduce that with probability at least $1-\delta/2$, for any center $\bar{f}$ in the set of covering index, we have

$$\begin{aligned} |R(\bar{f}) - \widehat{R}_{\mathcal{D}}(\bar{f})| &\le 4M^2\sqrt{\frac{\log(4N(t)/\delta)}{n}} \\ &\lesssim M^2 \cdot \left(\frac{C}{t}\right)^{\frac{d}{d+3}} \left(\frac{1}{n}\right)^{-\frac{1}{2}} + M^2\left(\frac{\log(4/\delta)}{n}\right)^{-\frac{1}{2}} \\ &\lessapprox_d M^2 \cdot \left(\frac{C}{t}\right)^{\frac{d}{d+3}} \left(\frac{1}{n}\right)^{-\frac{1}{2}}, \end{aligned} \tag{21}$$

where $\lessapprox_d$ hides the logarithmic factors about $1/\delta$ and constants.

According to the definition of covering sets, for any $f \in \mathcal{F}_{M,C}$, we have that $\|f - \bar{f}\|_\infty \le t$ for some center $\bar{f}$. Then we have

$$\begin{aligned} & |R(f) - \widehat{R}_{\mathcal{D}}(f)| \\ \lessapprox_d \ & |R(\bar{f}) - \widehat{R}_{\mathcal{D}}(\bar{f})| + O(Mt) \\ \lesssim_d \ & M^2 \cdot \left(\frac{C}{t}\right)^{\frac{d}{d+3}} n^{-\frac{1}{2}} + O(Mt). \end{aligned} \tag{22}$$

After tuning t to be the optimal choice, we deduce that (19). ■

Appendix E. Data-Dependent Regularity from Edge-of-Stability

This section summarizes the *data-dependent regularity* induced by minima stability for two-layer ReLU networks.

E.1. Function Space Viewpoint of Neural Networks Below the Edge of Stability

Recall the notations: given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$, we define the data-dependent weight function $g_{\mathcal{D}} : \mathbb{S}^{d-1} \times \mathbb{R} \to \mathbb{R}$ by

$$g_{\mathcal{D}}(\boldsymbol{u}, t) := \min\{\tilde{g}_{\mathcal{D}}(\boldsymbol{u}, t), \tilde{g}_{\mathcal{D}}(-\boldsymbol{u}, -t)\},$$

where

$$\tilde{g}_{\mathcal{D}}(\boldsymbol{u}, t) := \mathbb{P}_{\mathcal{D}}(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} > t)^2 \cdot \mathbb{E}_{\mathcal{D}}[\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t \mid \boldsymbol{X}^\mathsf{T}\boldsymbol{u} > t] \cdot \sqrt{1 + \|\mathbb{E}_{\mathcal{D}}[\boldsymbol{X} \mid \boldsymbol{X}^\mathsf{T}\boldsymbol{u} > t]\|^2}. \tag{23}$$

Here, $\boldsymbol{X}$ denotes a random draw uniformly sampled from $\{\boldsymbol{x}_i\}_{i=1}^n$, so that $\mathbb{P}_{\mathcal{D}}, \mathbb{E}_{\mathcal{D}}$ refer to probability and expectation under the empirical distribution $\frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{x}_i}$. When the dataset $\mathcal{D}$ is fixed and clear from context, we will simply write g in place of $g_{\mathcal{D}}$.

Then the curvature constrain on the loss landscape of $\mathcal{L}$ is converted into a weighted path norm constrain in the following sense.

Proposition 17 (Finite-sum version of Theorem 3.2 in [24]) *Suppose that $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^K v_k\, \phi(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k) + \beta$ is two-layer neural network such that the loss $\mathcal{L}$ is twice differentiable at $\boldsymbol{\theta}$. Then*

$$\sum_{k=1}^K |v_k|\, \|\boldsymbol{w}_k\| \cdot g\Big(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\Big) \le \frac{\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}))}{2} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{24}$$

If we write $f_{\boldsymbol{\theta}}$ into a reduced form in (10), then we have

$$\sum_{k=1}^{K'} a_k \cdot g\Big(\boldsymbol{u}_k, t_k\Big) \le \frac{\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}))}{2} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}. \tag{25}$$

Therefore, we bring up the definition the g-weighted path norm and variation norm are introduced as prior work introduced [24, 29].

Definition 18 *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 two-layer neural network. The (g-)weighted path-norm of $f_{\boldsymbol{\theta}}$ is defined to be*

$$\|f_{\boldsymbol{\theta}}\|_{\mathrm{path},g} := \sum_{k=1}^{K} |v_k|\, \|\boldsymbol{w}_k\|_2 \cdot g\Big(\frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|}, \frac{b_k}{\|\boldsymbol{w}_k\|}\Big). \tag{26}$$

Similarly, for functions of the form

$$f_{\nu,\boldsymbol{c},c_0}(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}\times[-R,R]} \phi(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x} - t)\, \mathrm{d}\nu(\boldsymbol{u},t) + \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + c_0, \quad \boldsymbol{x}\in\mathbb{R}^d, \tag{27}$$

*where $R > 0$, $\boldsymbol{c}\in\mathbb{R}^d$, and $c_0\in\mathbb{R}$, we define the g-*weighted variation (semi)norm *as*

$$|f|_{\mathrm{V}_g} := \inf_{\substack{\nu\in\mathcal{M}(\mathbb{S}^{d-1}\times[-R,R])\\ \boldsymbol{c}\in\mathbb{R}^d, c_0\in\mathbb{R}}} \|g\cdot\nu\|_{\mathcal{M}} \quad \text{s.t.} \quad f = f_{\nu,\boldsymbol{c},c_0}, \tag{28}$$

where, if there does not exist a representation of f in the form of (27), then the seminorm is understood to take the value $+\infty$. Here, $\mathcal{M}(\mathbb{S}^{d-1}\times[-R,R])$ denotes the Banach space of (Radon) measures and, for $\mu\in\mathcal{M}(\mathbb{S}^{d-1}\times[-R,R])$, $\|\mu\|_{\mathcal{M}} := \int_{\mathbb{S}^{d-1}\times[-R,R]} \mathrm{d}|\mu|(\boldsymbol{u},t)$ is the measure-theoretic total-variation norm.

With this seminorm, we define the Banach space of functions $\mathrm{V}_g(\mathbb{B}_R^d)$ on the ball $\mathbb{B}_R^d := \{\boldsymbol{x}\in\mathbb{R}^d : \|\boldsymbol{x}\|_2 \le R\}$ as the set of all functions f such that $|f|_{\mathrm{V}_g}$ is finite. When $g\equiv 1$, $|\cdot|_{\mathrm{V}_g}$ and $\mathrm{V}_g(\mathbb{B}_R^d)$ coincide with the variation (semi)norm and variation norm space of Bach [4].

For convenience, we introduce the notation of bounded weighted variation class

$$\mathcal{F}_g(\Omega; M, C) := \left\{ f:\Omega\to\mathbb{R} \,\middle|\, |f|_{\mathrm{V}_g} \le C,\ \|f|_{\Omega}\|_{L^\infty} \le M \right\}. \tag{29}$$

In particular, for any $\boldsymbol{\theta}\in\boldsymbol{\Theta}_g(\Omega; M, C)$, we have $f_{\boldsymbol{\theta}}\in\mathcal{F}_g(\Omega; M, C)$.

Within this framework together with the connection between $|\cdot|_{\mathrm{V}}$ and $\mathscr{R}\mathrm{TV}^2(\cdot)$ as summarized in Section D.2, we show the functional characterization of stable minima.

Theorem 19 *For any $f_{\boldsymbol{\theta}} \in \Theta_{\mathrm{BEoS}}(\eta,\mathcal{D})$, $|f_{\boldsymbol{\theta}}|_{\mathrm{V}_g} = \|g\cdot\mathscr{R}(-\Delta)^{\frac{d+1}{2}} f_{\boldsymbol{\theta}}\|_{\mathcal{M}} \le \frac{1}{\eta} - \frac{1}{2} + (R+1)\sqrt{2\mathcal{L}(\boldsymbol{\theta})}$.*

The detailed explanation and proof can be found in [24, Theorem 3.2, Corollary 3.3, Theorem 3.4, Appendix C, D].

E.2. Empirical Process for the Weight Function g

The implicit regularization of Edge-of-Stability induces a *data-dependent* regularity weight on the cylinder $\mathbb{S}^{d-1}\times\mathbb{R} := \mathbb{S}^{d-1}\times[-1,1]$. Denote this empirical weight by $g_{\mathcal{D}}$ for a dataset $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n$. Directly analyzing generalization through the random, data-dependent class weighted by $g_{\mathcal{D}}$ is conceptually delicate, since the hypothesis class itself depends on the sample. To separate *statistical* from *algorithmic* randomness, we adopt the following paradigm.

(1) Fix an underlying distribution $\mathcal{P}$ for X with only the support assumption $\mathrm{supp}(\mathcal{P}) \subseteq \mathbb{B}_R^d := \{\boldsymbol{x}\in\mathbb{R}^d : \|\boldsymbol{x}\| \le R\}$. Define a *population* reference weight $g_{\mathcal{P}}$ on $\mathbb{S}^{d-1}\times\mathbb{R}$ (see below). This anchors a distribution-level notion of regularity independent of the particular sample.

(ii) For a realized dataset $\mathcal{D} \sim \mathcal{P}^{\otimes n}$, form the empirical plug-ins that define $g_{\mathcal{D}}$ on the same index set $\mathbb{S}^{d-1} \times \mathbb{R}$.

(iii) Use empirical-process theory to control the uniform deviation $\|g_{\mathcal{D}} - g_{\mathcal{P}}\|_\infty$ with high probability over the draw of $\mathcal{D}$. After this step, we can *condition on the high-probability event* and regard $\mathcal{D}$ as fixed in any subsequent analysis.

Let $\boldsymbol{X} \sim \mathcal{P}$ with $\operatorname{supp}(\mathcal{P}) \subseteq \mathbb{B}_R^d$. For $(\boldsymbol{u}, t) \in \mathbb{S}^{d-1} \times \mathbb{R}$ define

$$p_{\mathcal{P}}(\boldsymbol{u}, t) := \mathcal{P}\big(\boldsymbol{X}^\top \boldsymbol{u} > t\big), \qquad s_{\mathcal{P}}(\boldsymbol{u}, t) := \mathbb{E}_{\boldsymbol{X}\sim\mathcal{P}}\big[(\boldsymbol{X}^\top \boldsymbol{u} - t)_+\big].$$

On the unit ball we have $0 \le (\boldsymbol{X}^\top \boldsymbol{u} - t)_+ \le 2$ and $\|\mathbb{E}_{\mathcal{P}}[X \mid \boldsymbol{X}^\top \boldsymbol{u} > t]\| \le 1$, which yields the *pointwise equivalence*

$$g_{\mathcal{P}}(\boldsymbol{u}, t) \; \asymp \; p_{\mathcal{P}}(\boldsymbol{u}, t)\, s_{\mathcal{P}}(\boldsymbol{u}, t) \quad \text{(with absolute constants).} \tag{30}$$

Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$, let $\mathbb{P}_{\mathcal{D}}, \mathbb{E}_{\mathcal{D}}$ denote probability and expectation under the empirical distribution $\frac{1}{n}\sum_{i=1}^n \delta_{x_i}$. Define

$$p_{\mathcal{D}}(\boldsymbol{u}, t) := \mathbb{P}_{\mathcal{D}}(\boldsymbol{X}^\top \boldsymbol{u} > t) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}\{\boldsymbol{x}_i^\top \boldsymbol{u} > t\}, \quad s_{\mathcal{D}}(\boldsymbol{u}, t) := \mathbb{E}_{\mathcal{D}}\big[(\boldsymbol{X}^\top \boldsymbol{u} - t)_+\big] = \frac{1}{n}\sum_{i=1}^n (\boldsymbol{x}_i^\top \boldsymbol{u} - t)_+,$$

and the empirical weight

$$g_{\mathcal{D}}(\boldsymbol{u}, t) \asymp p_{\mathcal{D}}(\boldsymbol{u}, t)\, s_{\mathcal{D}}(\boldsymbol{u}, t). \tag{31}$$

Lemma 20 (Uniform deviation for halfspaces) *There exists a universal constant $C > 0$ such that, for every $\delta \in (0, 1)$,*

$$\mathbb{P}\left(\sup_{u \in \mathbb{S}^{d-1},\, t \in [-1,1]} \big|p_{\mathcal{D}}(\boldsymbol{u}, t) - p_{\mathcal{P}}(\boldsymbol{u}, t)\big| > C\sqrt{\frac{d + \log(1/\delta)}{n}}\right) \le \delta.$$

Proof The class $\{(\boldsymbol{x} \mapsto \mathbb{1}\{\boldsymbol{x}^\top \boldsymbol{u} > t\}) : \boldsymbol{u} \in \mathbb{S}^{d-1},\, t \in \mathbb{R}\}$ has VC-dimension $d+1$. Apply the VC-uniform convergence inequality for $\{0,1\}$-valued classes (e.g., [40]) to the index set $\mathbb{S}^{d-1} \times [-1, 1]$ to obtain the stated bound. ■

Lemma 21 (Uniform deviation for ReLU) *There exists a universal constant $C > 0$ such that, for every $\delta \in (0, 1)$,*

$$\mathbb{P}\left(\sup_{u \in \mathbb{S}^{d-1},\, t \in [-1,1]} \big|s_{\mathcal{D}}(\boldsymbol{u}, t) - s_{\mathcal{P}}(\boldsymbol{u}, t)\big| > C\sqrt{\frac{d + \log(1/\delta)}{n}}\right) \le \delta.$$

Proof Let $\mathcal{F} := \{f_{\boldsymbol{u},t}(\boldsymbol{x}) = (\boldsymbol{u}^\top \boldsymbol{x} - t)_+ : \; \boldsymbol{u} \in \mathbb{S}^{d-1},\; t \in [-1, 1]\}$. Since $\|\boldsymbol{x}\| \le 1$ and $t \in [-1, 1]$, every $f \in \mathcal{F}$ takes values in $[0, 2]$. Consider the subgraph class

$$\mathsf{subG}(\mathcal{F}) = \big\{\, (\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R} : \; y \le (\boldsymbol{u}^\top \boldsymbol{x} - t)_+ \,\big\}.$$

For any $(\boldsymbol{x}, y)$ with $y \le 0$, membership in $\mathsf{subG}(\mathcal{F})$ holds for all parameters, hence such points do not contribute to shattering. For points with $y > 0$, the condition $y \le (\boldsymbol{u}^\top \boldsymbol{x} - t)_+$ is equivalent to $\boldsymbol{u}^\top \boldsymbol{x} - t - y \ge 0$, i.e., an affine halfspace in $\mathbb{R}^{d+1}$ with variables $(\boldsymbol{x}, y)$. Therefore the family $\mathsf{subG}(\mathcal{F})$ is (up to the immaterial fixed set $\{y \le 0\}$) parametrized by affine halfspaces in $\mathbb{R}^{d+1}$, whose VC-dimension is at most $d+2$. By the standard equivalence $\mathrm{Pdim}(\mathcal{F}) = \mathrm{VCdim}(\mathsf{subG}(\mathcal{F}))$, we obtain

$$\mathrm{Pdim}(\mathcal{F}) \;\le\; d+2.$$

Then by [16, Theorem 3, Theorem 6, Theorem 7], we

$$\sup_{(\boldsymbol{u},t)} \left| s_{\mathcal{D}}(\boldsymbol{u}, t) - s_{\mathcal{P}}(\boldsymbol{u}, t) \right| \;\le\; C\sqrt{\frac{d + \log(1/\delta)}{n}}$$

with probability at least $1-\delta$ for some universal constant C, which is the claimed bound. ■

Theorem 22 (Distribution-free uniform deviation for $\hat{g}_n$) *There exists a universal constant $C > 0$ such that, for every $\delta \in (0,1)$,*

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

Proof *By (30) and (31), it suffices (up to a universal factor) to control $\left| p_{\mathcal{D}} s_{\mathcal{D}} - p_{\mathcal{P}} s_{\mathcal{P}} \right|$. Using $0 \le s_{\mathcal{D}}, s_{\mathcal{P}} \le 2$ and $0 \le p_{\mathcal{D}}, p_{\mathcal{P}} \le 1$,*

$$\left| p_{\mathcal{D}} s_{\mathcal{D}} - p_{\mathcal{P}} s_{\mathcal{P}} \right| \le \; |p_{\mathcal{D}} - p_{\mathcal{P}}|\, s_{\mathcal{P}} + |s_{\mathcal{D}} - s_{\mathcal{P}}|\, p_{\mathcal{P}} + |p_{\mathcal{D}} - p_{\mathcal{P}}|\, |s_{\mathcal{D}} - s_{\mathcal{P}}|$$

Taking the supremum over $(\boldsymbol{u}, t) \in \mathbb{S}^{d-1} \times [-1,1]$ and applying Lemmas 20 and 21 with a union bound yields

$$\mathbb{P}\left(\sup_{\boldsymbol{u},t} \left| p_{\mathcal{D}} s_{\mathcal{D}} - p_{\mathcal{P}} s_{\mathcal{P}} \right| \gtrsim \sqrt{\frac{d + \log(1/\delta)}{n}} \right) \le 2\delta.$$

Finally, the equivalence $g \asymp p\, s$ transfers this bound to $\left| g_{\mathcal{D}} - g_{\mathcal{P}} \right|$ at the cost of an absolute multiplicative factor and one more failure event. ■

Appendix F. Generalization Upper Bound: Mixture of Low-Dimensional Balls

In this section, we present the proof of Theorem 3. We formalize the mixture model by the following setting.

Assumption 1 (Mixture of low-dimensional balls) *Let $\{V_j\}_{j=1}^J$ be a finite collection of J distinct m-dimensional (affine) linear subspaces within $\mathbb{R}^d$. Let $\mathcal{P}$ be a joint distribution over $\mathbb{R}^d \times \mathbb{R}$. The marginal distribution of the features $\boldsymbol{x}$ under $\mathcal{P}$, denoted $\mathcal{P}_{\boldsymbol{X}}$, is a mixture distribution given by*

$$\mathcal{P}_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{j=1}^J p_j \mathcal{P}_{\boldsymbol{X},j}(\boldsymbol{x}), \quad \mathcal{P}_{\boldsymbol{X},j}(\boldsymbol{x}) = \mathcal{P}_{\boldsymbol{X}}(\boldsymbol{x} \mid \boldsymbol{x} \in V_j), \tag{32}$$

where $p_j > 0$ are the mixture probabilities $\mathbb{P}(\boldsymbol{x} \in V_j)$ satisfying $\sum_{j=1}^J p_j = 1$. Each component distribution $\mathcal{P}_j$ is the uniform distribution on the unit ball $\mathbb{B}_1^{V_j} := \{\boldsymbol{x} \in V_j : \|\boldsymbol{x}\|_2 \le 1\}$. The corresponding labels y are generated from a conditional distribution $\mathcal{P}(y|\boldsymbol{x})$ and are assumed to be bounded, i.e., $|y| \le D$ for some constant $D > 0$. Similarly, we define $\mathcal{P}_j(\boldsymbol{x}, y) = \mathcal{P}(\boldsymbol{x}, y \mid \boldsymbol{x} \in V_j)$.

First, we prove the simple case of singe-subspace assumption ($J = 1$) via Theorem 24.

F.1. Case: uniform distribution on unit disc of a linear subspace

Fix an m-dimensional subspace $V \subset \mathbb{R}^d$ and write $\mathbb{B}_1^V := \{\boldsymbol{x} \in V : \|\boldsymbol{x}\|_2 \le 1\}$, the canonical linear projection $\mathrm{proj}_V : \mathbb{R}^d \to V$. Recall the notations in (1): the parameters $\boldsymbol{\theta} := \big\{(v_k, \boldsymbol{w}_k, b_k)_{k=1}^K,\ \beta\big\}$ with $\boldsymbol{w}_k \neq \boldsymbol{0}$, define a two-layer neural network

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^K v_k\, \phi(\boldsymbol{w}_k^\mathsf{T} \boldsymbol{x} - b_k) + \beta, \qquad \bar{\boldsymbol{w}}_k := \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|_2}, \quad \bar{b}_k := \frac{b_k}{\|\boldsymbol{w}_k\|_2}.$$

Then we define neuronwise projection operator from neural networks to neural networks

$$\mathrm{proj}_V^*\colon\ f_{\boldsymbol{\theta}}(\boldsymbol{x}) \mapsto \sum_{k=1}^K v_k\, \phi\big((\mathrm{proj}_V\, \boldsymbol{w}_k)^\mathsf{T} \boldsymbol{x} - b_k\big) + \beta. \tag{33}$$

Lemma 23 (Projection reduction) *Fix $\mathcal{F}$ a hyothesis class of two-layer neural networks. Let $\mathcal{P}$ be a joint distribution on $(\boldsymbol{x}, y)$ supported on $\mathbb{R}^d \times [-D, D]$ such that the marginal distribution $\mathcal{P}_{\boldsymbol{X}}$ of $\boldsymbol{x}$ supports on V. For any dataset $\mathcal{D} := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ drawn i.i.d. from $\mathcal{P}$,*

$$\sup_{f \in \mathcal{F}} \mathrm{Gap}_{\mathcal{P}}(f; \mathcal{D}) = \sup_{f \in \mathcal{F}} \mathrm{Gap}_{\mathcal{P}}(\mathrm{proj}_V^*\, f; \mathcal{D}). \tag{34}$$

Proof Because $\boldsymbol{x} \in V$ almost surely and in the sample, we have $f(\boldsymbol{x}) = (f \circ \mathrm{proj}_V)(\boldsymbol{x})$ for every f and every $\boldsymbol{x} \in \mathbb{B}_1^V$. Using the identity $\boldsymbol{w}_k^\mathsf{T}(\mathrm{proj}_V\, \boldsymbol{x}) = (\mathrm{proj}_V\, \boldsymbol{w}_k)^\mathsf{T} \boldsymbol{x}$, we obtain $f \circ \mathrm{proj}_V = \mathrm{proj}_V^*\, f$ pointwise on $\mathbb{B}_1^V$. Hence for any $f \in \mathcal{F}$, $\mathrm{Gap}_{\mathcal{P}}(f; \mathcal{D}) = \mathrm{Gap}_{\mathcal{P}}(\mathrm{proj}_V^*\, f; \mathcal{D})$. ■

Theorem 24 *Let $\mathcal{P}$ denote the joint distribution of $(\boldsymbol{x}, y)$. Assume that $\mathcal{P}$ is supported on $\mathbb{B}_1^d \times [-D, D]$ for some $D > 0$ and that the marginal distribution of $\boldsymbol{x}$ is $\mathrm{Uniform}(\mathbb{B}_1^V)$. Fix a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where each $(\boldsymbol{x}_i, y_i)$ is drawn i.i.d. from $\mathcal{P}$. Then, with probability $\geq 1 - \delta$,*

$$\sup_{f_{\boldsymbol{\theta}} \in \boldsymbol{\Theta}_{g_{\mathcal{D}}}(\mathbb{B}_1^{V_j}; M, C)} \mathrm{Gap}_{\mathcal{P}}(f_{\boldsymbol{\theta}}; \mathcal{D}) \lesssim_d C^{\frac{m}{m^2+4m+3}} M^2 n^{-\frac{1}{2m+4}} + M^2 \left(\frac{\log(4/\delta)}{n}\right)^{-\frac{1}{2}},$$

where $M := \max\{D, \|f_{\boldsymbol{\theta}}\|_{L^\infty(\mathbb{B}_1^V)}, 1\}$ and $\lesssim_d$ hides constants (which could depend on d).

Proof By Lemma 23, it remains to consider the case of $\mathrm{proj}_V^* f_{\boldsymbol{\theta}}$. Similarly, for any $\boldsymbol{u} \in \mathbb{S}^{d-1}$ and any data set $\mathcal{D} \subset V$, we have $g(\boldsymbol{u}, t) = g(\mathrm{proj}_V(\boldsymbol{u}), t)$. Therefore, we just need to consider the generalization gap with respect to the $\boldsymbol{\Theta}^V_{g_{\mathcal{D}}}(\mathbb{B}_1^{V_j}; M, C) = \left\{\mathrm{proj}_V^* f_{\boldsymbol{\theta}} \;:\; f_{\boldsymbol{\theta}} \in \boldsymbol{\Theta}_{g_{\mathcal{D}}}(\mathbb{B}_1^{V_j}; M, C)\right\}$. Therefore, we just need consider the case where the whole algorithm with any dataset sample from V operates in V and we get the result from [24, Theorem F.8] by replacing $\mathbb{R}^d$ with $V \cong \mathbb{R}^m$. ■

F.2. Proof of Theorem 3

In this section, we extend the generalization analysis from a single low-dimensional subspace to a more complex and practical scenario where the data is supported on a finite union of such subspaces. This setting is crucial for modeling multi-modal data, where distinct clusters can each be approximated by a low-dimensional linear structure. Our main result demonstrates that the sample complexity of stable minima adapts to the low intrinsic dimension of the individual subspaces, rather than the high ambient dimension of the data space.

F.2.1. ANALYSIS OF THE GLOBAL WEIGHT FUNCTION

A critical step in our proof is to understand the relationship between the global weight function $g(\boldsymbol{u}, t)$, which is induced by the mixture distribution $\mathcal{P}$, and the local weight functions $g_j(\boldsymbol{u}, t)$, each induced by a single component distribution $\mathcal{P}_j$ defined on V_j, which should be understood as the distribution conditioned to $\boldsymbol{x} \in V_j$. Fix a dataset $\mathcal{D}$, the function class $\Theta_{\mathrm{BEoS}}(\eta; \mathcal{D})$ is defined by the properties of the global function g. To analyze the performance on a specific subspace V_j, we must ensure that the global regularity constraint is sufficiently strong when viewed locally. The following lemma provides this crucial guarantee.

Lemma 25 (Global-to-Local Weight Domination) *For any mixed distribution $\mathcal{P}_X = \sum_{j=1}^J p_j \mathcal{P}_{\boldsymbol{X},j}$ with $\mathrm{supp}(\mathcal{P}_{\boldsymbol{X},j}) = V_j$. Let g be the global weight induced by the mixture $\mathcal{P}_X$, and g_j the weight induced by $\mathcal{P}_{\boldsymbol{X},j}$. For every $j \in \{1, \ldots, J\}$,*

$$g(\boldsymbol{u}, t) \geq \frac{p_j^2}{\sqrt{2}} g_j(\boldsymbol{u}, t), \quad \textit{for all } (\boldsymbol{u}, t) \in \mathbb{S}^{d-1} \times \mathbb{R}. \tag{35}$$

Consequently, for any $M, C > 0$,

$$\mathcal{F}_g(\mathbb{B}_1^{V_j}; M, C) \subseteq \mathcal{F}_{g_j}(\mathbb{B}_1^{V_j}; M, \sqrt{2}\, C/p_j^2)\,. \tag{36}$$

Proof Fix j and the activation event $A := \{\boldsymbol{x} : \boldsymbol{u}^\mathsf{T}\boldsymbol{x} > t\}$. By definition of g (global) and g_j (local) we can write

$$\begin{aligned}
g(\boldsymbol{u}, t) &= \mathcal{P}_X(A)^2 \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_X}[\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t \mid A] \cdot \sqrt{1 + \| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_X}[\boldsymbol{X} \mid A]\|_2^2} \\
g_j(\boldsymbol{u}, t) &= \mathcal{P}_X(A \mid \boldsymbol{x} \in V_j)^2 \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_X}[\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t \mid A, \boldsymbol{x} \in V_j] \cdot \sqrt{1 + \| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_X}[\boldsymbol{X} \mid A, \boldsymbol{x} \in V_j]\|_2^2} \\
&= \mathcal{P}_{\boldsymbol{X},j}(A)^2 \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t \mid A] \cdot \sqrt{1 + \| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[\boldsymbol{X} \mid A]\|_2^2}
\end{aligned}$$

Using the law of total probability and total expectation for the mixture distribution $\mathcal{P}_X = \sum_{i=1}^J p_i \mathcal{P}_{\boldsymbol{X},i}$, and the non-negativity of $(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A$, we get

$$\mathcal{P}_X(A) \;\geq\; p_j\, \mathcal{P}_{\boldsymbol{X},j}(A), \qquad \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_X}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A] \;\geq\; p_j \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A].$$

Hence, by combining the first two terms of $g(\boldsymbol{u}, t)$ as $\mathcal{P}_X(A)\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_X}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A]$, we have:

$$g(\boldsymbol{u}, t) \;\geq\; \big(p_j \mathcal{P}_{\boldsymbol{X},j}(A)\big) \cdot \big(p_j \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A]\big) \cdot 1 \;=\; p_j^2\, \mathcal{P}_{\boldsymbol{X},j}(A) \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A].$$

For the local weight function g_j, the same algebra gives

$$g_j(\boldsymbol{u}, t) = \mathcal{P}_{\boldsymbol{X},j}(A) \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A] \cdot \sqrt{1 + \| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[\boldsymbol{X} \mid A]\|_2^2}\,.$$

Since the support of $\mathcal{P}_{\boldsymbol{X},j}$ is $\mathbb{B}_1^{V_j}$, we have $\|\boldsymbol{X}\|_2 \leq 1$ almost surely under $\mathcal{P}_{\boldsymbol{X},j}$. This implies $\| \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[\boldsymbol{X} \mid A]\|_2 \leq 1$, and therefore $\sqrt{1 + \| \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[\boldsymbol{X} \mid A]\|_2^2} \leq \sqrt{2}$.

Combining these results, we establish the lower bound:

$$g(\boldsymbol{u}, t) \;\geq\; \frac{p_j^2}{\sqrt{2}} \left(\mathcal{P}_{\boldsymbol{X},j}(A) \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{u} - t)\mathbb{1}_A] \cdot \sqrt{1 + \| \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{X},j}}[\boldsymbol{X} \mid A]\|_2^2} \right) \;=\; \frac{p_j^2}{\sqrt{2}}\, g_j(\boldsymbol{u}, t),$$

which proves (35). The class embedding (36) follows directly from the definition of the weighted variation seminorm. ■

Proposition 26 *Let $\mathcal{P}$ be a distribution defined in Assumption 1 and recall that $\mathcal{P}_j$ is $\mathcal{P}$ conditional to $\boldsymbol{x} \in V_j$. Fix $j \in \{1, \ldots, J\}$ and a data set $\mathcal{D} \sim \mathcal{P}^{\otimes n}$. Let $\mathcal{D}_j := \mathcal{D} \cap V_j$ and $n_j := |\mathcal{D}_j|$. Then with probability $1 - \delta$,*

$$\sup_{f_{\boldsymbol{\theta}} \in \Theta_{\mathrm{BEoS}}(\eta, \mathcal{D})} \mathrm{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j) \;\lesssim_d\; \left(\frac{\frac{1}{\eta} - \frac{1}{2} + 4M}{p_j^2} \right)^{\frac{m}{m^2+4m+3}} M^2\, n_j^{-\frac{1}{2m+4}} \;+\; M^2 \left(\frac{\log(4/\delta)}{n} \right)^{-\frac{1}{2}}. \tag{37}$$

where $M := \max\{D, \|f_{\boldsymbol{\theta}}\|_{L^\infty(\mathbb{B}_1^{V_j})}, 1\}$ and $\lesssim_d$ hides constants (which could depend on d).

Proof Note that the notation $\mathrm{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j)$ can be expanded into

$$\begin{aligned}\mathrm{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j) &= \left| R_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}) - \widehat{R}_{\mathcal{D}_j}(f_{\boldsymbol{\theta}}) \right| \\ &= \left| \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{P}_j} \left[(f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y)^2 \right] - \widehat{R}_{\mathcal{D}_j}(f_{\boldsymbol{\theta}}) \right| \\ &= \left| \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{P}} \left[(f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y)^2 \mid \boldsymbol{x} \in V_j \right] - \widehat{R}_{\mathcal{D}_j}(f_{\boldsymbol{\theta}}) \right|\end{aligned}$$

Let $C = \frac{1}{\eta} - \frac{1}{2} + 4M$. According to [24, Corollary 3.3], we have that

$$f_{\boldsymbol{\theta}} \in \boldsymbol{\Theta}_{g_{\mathcal{D}}}(\mathbb{B}_1^{V_j}; M, C), \quad \forall \boldsymbol{\theta} \in \Theta_{\mathrm{BEoS}}(\eta; \mathcal{D}).$$

Then by Lemma 25, we conclude that

$$\boldsymbol{\Theta}_g(\mathbb{B}_1^{V_j}; M, C) \subseteq \boldsymbol{\Theta}_{g_j}(\mathbb{B}_1^{V_j}; M, \sqrt{2}C/p_j^2),$$

where the weight functions g and g_j can be either empirical or population.

Therefore,

$$\sup_{\boldsymbol{\theta}\in\Theta_{\mathrm{BEoS}}(\eta;\mathcal{D})} \mathrm{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j) \le \sup_{f\in\boldsymbol{\Theta}_{g_j}(\mathbb{B}_1^{V_j}; M, \sqrt{2}C/p_j^2)} \mathrm{Gap}_{\mathcal{P}_j}(f; \mathcal{D}_j)$$

Then by Theorem 24, we may conclude that

$$\sup_{f_{\boldsymbol{\theta}}\in\mathcal{F}_{g_j}(\mathbb{B}_1^{V_j}; M, \sqrt{2}C/p_j^2)} \mathrm{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j) \lessapprox_d \left(\frac{\frac{1}{\eta} - \frac{1}{2} + 4M}{p_j^2} \right)^{\frac{m}{m^2+4m+3}} M^2 \, n_j^{-\frac{1}{2m+4}}$$

■

Theorem 27 (Generalization Bound for Mixture Models) *Let the data distribution $\mathcal{P}$ be as defined in Assumption 1. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset of n i.i.d. samples drawn from $\mathcal{P}$. Then, with probability at least $1 - 2\delta$,*

$$\sup_{\boldsymbol{\theta}\in\Theta_{\mathrm{BEoS}}(\eta,\mathcal{D})} \mathrm{Gap}_{\mathcal{P}}(f_{\boldsymbol{\theta}}; \mathcal{D}) \lesssim_d \left(\frac{1}{\eta} - \frac{1}{2} + 4M \right)^{\frac{m}{m^2+4m+3}} M^2 \, J^{\frac{4}{m}} \, n^{-\frac{1}{2m+4}} + M^2 \, J \sqrt{\frac{\log(4J/\delta)}{2n}}. \tag{38}$$

where $M := \max\{D, \|f_{\boldsymbol{\theta}}\mid_{\mathbb{B}_1^V}\|_{L^\infty}, 1\}$ and $\lesssim_d$ hides constants (which could depend on d).

The proof proceeds in several steps. First, we establish a high-probability event where the number of samples drawn from each subspace is close to its expected value. Second, we decompose the total generalization gap into several terms. Finally, we bound each of these terms, showing that the dominant term is determined by the generalization performance on the individual subspaces, which scales with the intrinsic dimension m.

Proof Let $n_j = \sum_{i=1}^{n} \mathbb{1}_{\{x_i \in V_j\}}$ be the number of samples from the dataset $\mathcal{D}$ that fall into the subspace V_j. Each n_j is a random variable following a Binomial distribution, $n_j \sim \text{Bin}(n, p_j)$. We need to ensure that for all subspaces simultaneously, the empirical proportion n_j/n is close to the true probability p_j.

We use Hoeffding's inequality for each $j \in \{1, \ldots, J\}$. For any $\epsilon > 0$, $\mathbb{P}\left(\left|\frac{n_j}{n} - p_j\right| \geq \epsilon\right) \leq 2e^{-2n\epsilon^2}$. To ensure this holds for all J subspaces at once, we apply a union bound. Let δ_j be the failure probability allocated to the j-th subspace. The total failure probability is at most $\sum_{j=1}^{J} \delta_j = \delta$, so we set $\delta_j = \delta/J$ and yields $\epsilon = \sqrt{\frac{\log(2J/\delta)}{2n}}$.

Let $\mathcal{E}$ be the event that $|\frac{n_j}{n} - p_j| \leq \epsilon$ holds for all $j = 1, \ldots, J$. We have shown that $\mathbb{P}(\mathcal{E}) \geq 1 - \delta$. The remainder of our proof is conditioned on this event $\mathcal{E}$. A direct consequence of this event is a lower bound on each n_j

$$n_j \geq np_j - n\epsilon = np_j - \sqrt{\frac{n}{2} \log \frac{2J}{\delta}}. \tag{39}$$

Now we decompose the generalization gap using the law of total expectation for the true risk and by partitioning the empirical sum for the empirical risk.

Let $\mathcal{P}_j$ denote the distribution $\mathcal{P}$ conditioned on $x \in V_j$, and let $\mathcal{D}_j = \mathcal{D} \cap V_j\}$.

$$\begin{aligned}
\text{Gap}_{\mathcal{P}}(f_{\boldsymbol{\theta}}; \mathcal{D}) &= \left| R(f_{\boldsymbol{\theta}}) - \widehat{R}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) \right| \\
&= \left| \sum_{j=1}^{J} p_j \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{P}} [(f(\boldsymbol{x}) - y)^2 \mid \boldsymbol{x} \in V_j] - \sum_{j=1}^{J} \frac{n_j}{n} \frac{1}{n_j} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_j} (f(\boldsymbol{x}_i) - y_i)^2 \right| \\
&\leq \left| \sum_{j=1}^{J} p_j \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{P}_j} [(f_{\boldsymbol{\theta}}(\boldsymbol{x}) - y)^2] - \sum_{j=1}^{J} p_j \frac{1}{n_j} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_j} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 \right| \\
&\quad + \left| \sum_{j=1}^{J} p_j \frac{1}{n_j} \sum_{(x_i, y_i) \in \mathcal{D}_j} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 - \sum_{j=1}^{J} \frac{n_j}{n} \frac{1}{n_j} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_j} (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 \right| \\
&\leq \sum_{j=1}^{J} p_j \left| R_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}) - \widehat{R}_{\mathcal{D}_j}(f_{\boldsymbol{\theta}}) \right| + \sum_{j=1}^{J} \left| p_j - \frac{n_j}{n} \right| \widehat{R}_{\mathcal{D}_j}(f_{\boldsymbol{\theta}}) \\
&= \underbrace{\sum_{j=1}^{J} p_j \text{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j)}_{\text{Term A}} + \underbrace{\sum_{j=1}^{J} \left| p_j - \frac{n_j}{n} \right| \widehat{R}_{\mathcal{D}_j}(f_{\boldsymbol{\theta}})}_{\text{Term B}}
\end{aligned}$$

where $\widehat{R}_{\mathcal{D}_j}(f) = \frac{1}{n_j} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}_j} (f(\boldsymbol{x}_i) - y_i)^2$.

- **Bounding the Weighted Sum of Conditional Gaps (Term A):** According to Proposition 26, with probability at least $1 - \delta$, for each j,

$$\text{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j) \lesssim_d \left(\frac{\frac{1}{\eta} - \frac{1}{2} + 4M}{p_j^2} \right)^{\frac{m}{m^2+4m+3}} M^2 \, n_j^{-\frac{1}{2m+4}} + M^2 \left(\frac{\log(4J/\delta)}{n} \right)^{-\frac{1}{2}}.$$

Conditioned on $\mathcal{E}$, we use the lower bound on n_j from (39) , $n_j \leq np_j(1-\epsilon/p_j)$.

$$\begin{aligned}
\text{Term A} &= \sum_{j=1}^{J} p_j \mathrm{Gap}_{\mathcal{P}_j}(f_{\boldsymbol{\theta}}; \mathcal{D}_j) \\
&\lessapprox_d \sum_{j=1}^{J} p_j \Big(\frac{\frac{1}{\eta} - \frac{1}{2} + 4M}{p_j^2}\Big)^{\frac{m}{m^2+4m+3}} M^2 (np_j(1-\epsilon/p_j))^{-\frac{1}{2m+4}} \\
&= \Big(\frac{1}{\eta} - \frac{1}{2} + 4M\Big)^{\frac{m}{m^2+4m+3}} M^2 n^{-\frac{1}{2m+4}} \sum_{j=1}^{J} p_j \cdot (p_j^{-2})^{\frac{m}{m^2+4m+3}} \cdot \left(p_j - \sqrt{\frac{\log(2J/\delta)}{2n}}\right)^{-\frac{1}{2m+4}} \\
&\lessapprox_d \Big(\frac{1}{\eta} - \frac{1}{2} + 4M\Big)^{\frac{m}{m^2+4m+3}} M^2 n^{-\frac{1}{2m+4}} \sum_{j=1}^{J} p_j^{1-\frac{2m}{m^2+4m+3}-\frac{1}{2m+4}}.
\end{aligned}$$

The exponent of p_j simplifies to

$$1 - \frac{2m}{(m+1)(m+3)} - \frac{1}{2m+4)} = \frac{2m^3+7m^2+10m+9}{2(m+1)(m+2)(m+3)}. \tag{40}$$

For positive integers m, (40) is strictly increasing and bounded above by 1. In particular, when $m=1$, (40) $= \frac{7}{12}$. Therefore, a brute-force upper bound is

$$\sum_{j=1}^{J} p_j^{\frac{2m^3+7m^2+10m+9}{2(m+1)(m+2)(m+3)}} \leq J$$

and thus

$$\begin{aligned}
\text{Term A} &\lessapprox_d \Big(\frac{1}{\eta} - \frac{1}{2} + 4M\Big)^{\frac{m}{m^2+4m+3}} M^2 n^{-\frac{1}{2m+4}} \sum_{j=1}^{J} p_j^{1-\frac{2m}{m^2+4m+3}-\frac{1}{2m+4}} \\
&\lessapprox_d \Big(\frac{1}{\eta} - \frac{1}{2} + 4M\Big)^{\frac{m}{m^2+4m+3}} M^2\, J\, n^{-\frac{1}{2m+4}}.
\end{aligned}$$

Note that the dependence of Term A on J is very mild. Indeed, if we denote

$$\alpha(m) = 1 - \frac{2m}{m^2+4m+3} - \frac{1}{2m+4},$$

then

$$\sum_{j=1}^{J} p_j^{\alpha(m)} \leq J^{1-\alpha(m)} \leq J^{\frac{2m}{m^2+4m+3}+\frac{1}{2m+4}} \leq J^{\frac{4}{m}},$$

since $\sum_j p_j = 1$. For large m, the exponent $\alpha(m)$ is close to 1, hence $\sum_j p_j^{\alpha(m)}$ remains essentially of order one. Consequently, the bound on Term A grows at most linearly with J, and in practice the J-dependence is negligible in high m. Here we use the power $4/m$ upper for clean format.

- **Bounding the Sampling Deviation Error (Term B):** Conditioned on the event $\mathcal{E}$, we have $|p_j - n_j/n| \le \epsilon$ for all j. The empirical risk term is bounded because $\max\{|f(\boldsymbol{x})|, |y|\} \le M$, which implies $|\frac{1}{n_j}\sum_{(\boldsymbol{x}_i,y_i)\in\mathcal{D}_j}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2| \le 4M^2$. Thus, Term B is bounded by:

$$\text{Term B} \le \sum_{j=1}^{J} \epsilon 4M^2 = 4M^2\epsilon = 4JM^2\sqrt{\frac{\log(4J/\delta)}{2n}}. \tag{41}$$

The total generalization gap is bounded by the sum of the bounds for Term A and Term B.

$$\mathrm{Gap}_{\mathcal{P}}(f_{\boldsymbol{\theta}};\mathcal{D}) \lesssim_d \left(\frac{1}{\eta} - \frac{1}{2} + 4M\right)^{\frac{m}{m^2+4m+3}} M^2\, J^{\frac{4}{m}}\, n^{-\frac{1}{2m+4}} + M^2\, J\sqrt{\frac{\log(4J/\delta)}{2n}}.$$

This completes the proof. ■

Appendix G. Flat Interpolating Two-Layer ReLU Networks on the Unit Sphere

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset with $\boldsymbol{x}_i \in \mathbb{S}^{d-1}$, $d > 1$, and pairwise distinct inputs. Assume labels are uniformly bounded, i.e., $|y_i| \le D$ for all i. Consider width-K two-layer ReLU models

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k\,\phi(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k) + \beta. \tag{42}$$

Theorem 28 (Flat interpolation with width $\le n$) *Under the set-up above, there exists a width $K \le n$ network of the form (42) that interpolates the dataset and whose Hessian operator norm satisfies*

$$\lambda_{\max}\left(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}\right) \;\le\; 1 + \frac{D^2+2}{n}. \tag{43}$$

Construction 29 (Flat interpolation ReLU network) *Let $I_{\neq 0} := \{i : \; y_i \neq 0\}$ and set the width $K := |I_{\neq 0}| \le n$. For each $k \in I_{\neq 0}$ define*

$$\rho_k \;:=\; \max_{k\neq i} \boldsymbol{x}_i^\mathsf{T}\boldsymbol{x}_k \;<\; 1, \qquad b_k \in (\rho_k,\, 1)\;\; (\text{e.g., } b_k = \frac{1+\rho_k}{2}), \qquad \boldsymbol{w}_k := \boldsymbol{x}_k. \tag{44}$$

Then for any sample index i,

$$\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x}_i - b_k = \begin{cases} 1 - b_k \;>\; 0, & i = k, \\ \le \rho_k - b_k \;<\; 0, & i \neq k, \end{cases} \tag{45}$$

so the k-th unit activates on $\boldsymbol{x}_k$ and is inactive on all $\boldsymbol{x}_i$ with $i \neq k$. Set the output weight

$$v_k \;:=\; \frac{y_k}{1-b_k}. \tag{46}$$

By (45) and (46), the model interpolates on nonzero labels because $f(\boldsymbol{x}_k) = a_k(1-b_k) = y_k$ for $k \in I_{\neq 0}$, and it also interpolates zero labels since all constructed units are inactive on $\boldsymbol{x}_i$ when $i \notin J_{\neq 0}$, hence $f(\boldsymbol{x}_i) = 0 = y_i$.

For each constructed unit, define

$$\tilde{v}_k := \operatorname{sign}(v_k) \in \{\pm 1\}, \qquad \tilde{\boldsymbol{w}}_k := |v_k|\, \boldsymbol{w}_k, \qquad \tilde{b}_k := |a_k|\, b_k. \tag{47}$$

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

$$\tilde{v}_k\, \phi(\tilde{\boldsymbol{w}}_k^\top \boldsymbol{x} - \tilde{b}_k) = \operatorname{sign}(v_k)\, \phi\big(|v_k|(\boldsymbol{w}_k^\top \boldsymbol{x} - b_k)\big) = v_k\, \phi(\boldsymbol{w}_k^\top \boldsymbol{x} - b_k), \tag{48}$$

so interpolation is preserved. Moreover, the activation pattern on the dataset is unchanged because (45) *has strict inequalities and $|a_i| > 0$. At $\boldsymbol{x}_i$ we have the (post-rescaling) pre-activation*

$$\tilde{z}_k \ := \ \tilde{\boldsymbol{w}}_k^\top \boldsymbol{x}_k - \tilde{b}_k = |a_k|\,(1 - b_k) = |y_k| \ > \ 0, \qquad |\tilde{v}_i| = 1. \tag{49}$$

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 $z_k := \boldsymbol{w}_k^\top \boldsymbol{x}_k - b_k = |y_k|$.

Proposition 30 *Let $\boldsymbol{\theta}$ be the model in Construction 29. Then*

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

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

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L} = \frac{1}{n} \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i) \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)^\top + \frac{1}{n} \sum_{i=1}^n (f(\boldsymbol{x}_i) - y_i) \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x}_i). \tag{50}$$

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

$$\nabla^2_{\boldsymbol{\theta}} \mathcal{L} = \frac{1}{n} \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i) \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_i)^\top. \tag{51}$$

Denote the tangent features matrix by

$$\boldsymbol{\Phi} = \left[\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_1), \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_2), \cdots, \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_n)\right]. \tag{52}$$

Then $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}$ in (51) can be expressed by $\nabla^2_{\boldsymbol{\theta}} \mathcal{L} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top / n$, and the operator norm is computed by

$$\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}) = \max_{\boldsymbol{\gamma} \in \mathbb{S}^{(d+2)K}} \frac{1}{n} \|\boldsymbol{\Phi}^\top \boldsymbol{\gamma}\|^2 = \max_{\boldsymbol{u} \in \mathbb{S}^{n-1}} \frac{1}{n} \|\boldsymbol{\Phi} \boldsymbol{u}\|^2 \tag{53}$$

From direct computation we obtain

$$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}) = \begin{pmatrix} \nabla_{\boldsymbol{W}}(f) \\ \nabla_{\boldsymbol{b}}(f) \\ \nabla_{\boldsymbol{\omega}}(f) \\ \nabla_{\beta}(f) \end{pmatrix} \tag{54}$$

For the parameters $[\boldsymbol{w}_k, b_k, v_k]$ associated to the neuron of index j,

$$\frac{\partial f(\boldsymbol{x})}{\partial v_k} = \mathbb{1}\{\boldsymbol{w}_k^\top \boldsymbol{x} > b_k\} \left(\boldsymbol{w}_k^\top \boldsymbol{x} - b_k\right), \qquad \frac{\partial f(\boldsymbol{x}_i)}{\partial \boldsymbol{w}_k} = \mathbb{1}\{\boldsymbol{w}_k^\top \boldsymbol{x} > b_k\}\, v_k\, \boldsymbol{x},$$

$$\frac{\partial f(\boldsymbol{x}_i)}{\partial b_k} = \mathbb{1}\{\boldsymbol{w}_k^\top \boldsymbol{x} > b_k\}\, v_k, \qquad \frac{\partial f(\boldsymbol{x}_i)}{\partial \beta} = 1.$$

By the one-to-one activation property (45), each sample $\boldsymbol{x}_i$ activates exactly one unit (the unit with the same index k when $k \in I_{\neq 0}$), and activates none when $i \notin I_{\neq 0}$. Hence the sample-wise gradient $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}_k)$ has support only on the parameter triplet $(\boldsymbol{w}_k, b_k, v_k, \beta)$ for $k \in I_{\neq 0}$, and is zero for other parameters. Writing the nonzero gradient block explicitly (recall $|v_k| = 1$),

$$\begin{aligned} \nabla_{(\boldsymbol{w}_k, b_k, v_k, \beta)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_k) &= \begin{pmatrix} \nabla_{(\boldsymbol{w}_k, b_k, v_k)} f_{\boldsymbol{\theta}} \\ 1 \end{pmatrix}, \\ \nabla_{(\boldsymbol{w}_k, b_k, v_k)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_k) &= \begin{cases} \begin{pmatrix} v_k \, \boldsymbol{x}_k \\ v_k \\ y_k \end{pmatrix}, & (k \in I_{\neq 0}), \\ \mathbf{0}, & (k \notin I_{\neq 0}), \end{cases} \end{aligned} \tag{55}$$

After row permutation and subsistion by (55), (53) is of the form

$$\boldsymbol{\Phi} = \begin{pmatrix} \nabla_{(\boldsymbol{w}_1, b_1, v_1)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \nabla_{(\boldsymbol{w}_2, b_2, v_2)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_2) & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \vdots \\ \vdots & \vdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \nabla_{(\boldsymbol{w}_n, b_n, v_n)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \\ 1 & 1 & \cdots & 1 \end{pmatrix} \tag{56}$$

$$= \begin{pmatrix} \begin{pmatrix} v_1 \, \boldsymbol{x}_1 \\ v_1 \\ y_1 \end{pmatrix} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \begin{pmatrix} v_2 \, \boldsymbol{x}_2 \\ v_2 \\ y_2 \end{pmatrix} & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \vdots \\ \vdots & \vdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \begin{pmatrix} v_n \, \boldsymbol{x}_n \\ v_n \\ y_n \end{pmatrix} \\ 1 & 1 & \cdots & 1 \end{pmatrix}. \tag{57}$$

Let $\boldsymbol{u} = (u_1, \cdots, u_n) \in \mathbb{S}^{n-1}$ and plug (57) in (53) to have

$$\begin{aligned}
\lambda_{\max}(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}) &= \max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \frac{1}{n}\|\boldsymbol{\Phi u}\|^2 \qquad (58)\\
&= \frac{1}{n}\max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \left\| \begin{pmatrix} u_1\nabla_{(\boldsymbol{w}_1,b_1,v_1)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_1) \\ u_2\nabla_{(\boldsymbol{w}_2,b_2,v_2)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_2) \\ \vdots \\ u_n\nabla_{(\boldsymbol{w}_n,b_n,v_n)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \\ \sum_{i=1}^n u_i \end{pmatrix} \right\|_2^2 \\
&= \frac{1}{n}\max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \sum_{i=1}^n u_i^2 \left\|\nabla_{(\boldsymbol{w}_i,b_i,v_i)} f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right\|_2^2 + \left(\sum_{i=1}^n u_i\right)^2 \qquad (59)\\
&= \frac{1}{n}\max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \sum_{i=1}^n u_i^2 \left(\|\boldsymbol{x}_i\|_2^2 + 1 + y_i^2\right) + \left(\sum_{i=1}^n u_i\right)^2 \qquad (60)\\
&\le \frac{1}{n}\left(\max_{i\in[n]} \left(\|\boldsymbol{x}_i\|_2^2 + 1 + y_i^2\right) + \max_{\boldsymbol{u}\in\mathbb{S}^{n-1}} \left(\sum_{i=1}^n u_i\right)^2\right) \qquad (61)\\
&\le \frac{1}{n}\left(D^2 + 2 + n\right) = 1 + \frac{D^2+2}{n}
\end{aligned}$$

If we remove the output bias term β from the parameters, then the bottom row of 57 will be remove and thus term $\sum_i u_i$ in (59) will be removed. ■