Simplicity Bias of Two-Layer Networks beyond Linearly Separable Data

Nikita Tsoy [1] **Nikola Konstantinov** [1]

Abstract

Simplicity bias, the propensity of deep models to over-rely on simple features, has been identified as a potential reason for limited out-of-distribution generalization of neural networks (Shah et al., 2020). Despite the important implications, this phenomenon has been theoretically confirmed and characterized only under strong dataset assumptions, such as linear separability (Lyu et al., 2021). In this work, we characterize simplicity bias for general datasets in the context of two-layer neural networks initialized with small weights and trained with gradient flow. Specifically, we prove that in the early training phases, network features cluster around a few directions that do not depend on the size of the hidden layer. Furthermore, for datasets with an XOR-like pattern, we precisely identify the learned features and demonstrate that simplicity bias intensifies during later training stages. These results indicate that features learned in the middle stages of training may be more useful for OOD transfer. We support this hypothesis with experiments on image data.

1. Introduction

Out-of-distribution (OOD) generalization is a key challenge towards the widespread adoption of machine learning. Specifically, since training data may not always cover all possible test scenarios, networks often rely on shortcuts: spurious rules that hold on the training distribution but not in more complicated real-world situations (Geirhos et al., 2020). For example, convolutional networks often prioritize texture over shape (Geirhos et al., 2019), or transformers might rely on simplistic heuristics in natural language inference (McCoy et al., 2019).

One possible mechanism behind shortcuts is *simplicity bias*, the propensity of neural networks to rely only on "simple" features. As Shah et al. (2020) demonstrated, this bias might be persistent and hurt OOD generalization in image classification tasks. Simplicity bias is also a peculiar phenomenon from a theoretical perspective. Since many neural architectures are universal function approximators (Cybenko, 1989; Hornik et al., 1989), one might hope that models will learn other, more sophisticated patterns within the data.

Despite the importance of simplicity bias, a thorough theoretical understanding of this phenomenon is still lacking. To the best of our knowledge, existing works on simplicity bias (Lyu et al., 2021; Safran et al., 2022; Morwani et al., 2023) only demonstrate its emergence by employing stringent assumptions, which restrict training data to be linearly separable or one-dimensional.

Contributions We give the first proof of the existence of simplicity bias and a precise mathematical characterization of the features learned during training beyond linearly separable data. We do so for two-layer neural networks trained with gradient flow from a small initialization, a model popular in the theoretical literature (Luo et al., 2021a; Lyu et al., 2021; Boursier et al., 2022). We characterize simplicity bias as a property that only a small set of *prominent* neurons governs the behavior of the network. These prominent neurons cluster in several directions, which do not depend on the size of the hidden layer.

Specifically, our theoretical analysis divides training into three stages. During the first two stages (Section 4), in which the weights grow from small to constant scale, we prove for general datasets that the most prominent features recovered by the training dynamics cluster around the extrema of a data-dependent function, which does not depend on the number of neurons and disentangles the interactions between them. For the last stage, where the network converges to zero loss, we prove that simplicity bias can become extreme even in non-linearly-separable datasets. In this stage, we cover the case of XOR-like data under an assumption about the convergence of a 4-neuron network, which we validate experimentally. On a methodological level, our work generalizes the analysis of Lyu et al. (2021) beyond linearly separable data and provides an implicit description of the most prominent features for general datasets.

[1]INSAIT, Sofia University, Bulgaria. Correspondence to: Nikita Tsoy <nikita.tsoy@insait.ai>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Our theoretical findings additionally lead to a hypothesis with potential practical implications: networks trained to a very small loss may be prone to stronger simplicity bias and, hence, may be harder to finetune to new tasks. We test this intuition with experiments on a domino dataset of MNIST-CIFAR10 pairs (Shah et al., 2020) and observe experimental support for our hypothesis.[1]

2. Related Work

Simplicity bias of features The simplicity bias of neural networks was observed and linked to generalization and OOD performance in prior work (Valle-Perez et al., 2019; Shah et al., 2020). Several works show that two-layer networks provably learn a linear decision boundary on linearly separable datasets (Brutzkus et al., 2018; Pellegrini & Biroli, 2020; Sarussi et al., 2021; Phuong & Lampert, 2021; Lyu et al., 2021; Englert & Lazic, 2022; Frei et al., 2023b; Kou et al., 2023; Morwani et al., 2023; Wang & Ma, 2023; Chistikov et al., 2023; Min et al., 2024). While we use some of the techniques developed by these works, we focus on the non-linearly-separable case, which requires further theoretical analysis. Safran et al. (2022) prove that, on one-dimensional data, two-layer networks converge to a model with few linear regions. In contrast, we study datasets in $\mathbb{R}^d$. Brutzkus & Globerson (2019) also show simplicity bias for XOR-like data, but only for a 4-point dataset in $\mathbb{R}^2$, while we focus on datasets of arbitrary size in $\mathbb{R}^d$.

Simplicity bias of training dynamics A direction related to our work is simplicity bias in terms of training dynamics, the propensity of neural networks to learn simple patterns first in training. This property was observed in several empirical works (Arpit et al., 2017; Xu et al., 2019; Rahaman et al., 2019; Kalimeris et al., 2019) and demonstrated theoretically in certain settings (Arora et al., 2019; Basri et al., 2020; Luo et al., 2021a; Bowman & Montufar, 2022). Another closely related topic is the distributional simplicity bias proposed by Refinetti et al. (2023). In contrast to these works, we seek to characterize the learned features instead of analyzing some complexity invariant.

Small initialization and initial condensation Our results in Section 4 provably demonstrate initial condensation, the propensity of neural networks to condense neurons in few directions during early stages of training, for two-layer networks with small initialization. Several theoretical works have analyzed this phenomenon previously. Maennel et al. (2018) study condensation in regression and classification problems with two-layer ReLU networks. While they also recognize the function G we use to describe the network dynamics as an important factor in learning, their derivations and discussions of the near-zero initialization regime, which we study in our paper, are informal. In addition, their bounds for the speed of neuron alignment are insufficient to differentiate between prominent and non-prominent neurons, as we do in Section 4. Zhou et al. (2022) give another theoretical description of condensation in regression tasks. However, their results do not apply to ReLU activation or our differentiable approximation of ReLU (due to irregularity at 0). Boursier et al. (2022) also identify the condensation phenomenon in regression tasks, but only for orthogonal data, while we put considerably milder assumptions on the data. Additionally, Xu & Du (2023) recognize the condensation phenomenon in a regression setting with a one-neuron teacher network, which is similar to a linearly separable case because the direction of the gradients will be correlated with the direction of the teacher neuron.

Finally, the concurrent work of Boursier & Flammarion (2024) proves an alignment result similar to that in our Theorem 4.1. However, their result does not quantify the differences in the neurons' growth rates and, hence, can not differentiate between prominent and non-prominent neurons. Moreover, there are several technical differences. Boursier & Flammarion (2024) work directly with (leaky) ReLU and make milder assumptions on the initialization. However, they require the function G to have no saddle points and only show alignment results for neurons that satisfy a certain technical condition (Condition 1 in their manuscript).

Small initialization and mean-field regime Since we work in the small initialization regime of two-layer networks, our setting is similar to the mean-field regime (Chizat & Bach, 2018; Mei et al., 2018; Wei et al., 2019; Li et al., 2020; Ge et al., 2021). However, in our setting, we do not increase the number of neurons when we decrease the initialization scale (which corresponds to the condensed regime in the classification of Luo et al., 2021b). Thus, in our limit, at the beginning of training, the neurons evolve independently, while, in the mean-field limit, they interact via the velocity field (the derivative of a loss function). At the same time, some techniques in our work are similar to the mean-field techniques since both we and the mean-field works analyze the behavior of a loss function near zero.

3. Setting

Throughout the paper, we analyze feature learning on a binary classification problem with a two-layer network initialized with small random parameter values.

Notation We use the following notation: $\boldsymbol{v}^j$ is jth component of vector $\boldsymbol{v}$, so that $\boldsymbol{v} = (v^1, v^2, \ldots, v^d)^\mathsf{T}$, $\|\boldsymbol{v}\| := \sqrt{\sum_{j=1}^d (v^j)^2}$ is the usual l_2-norm of $\boldsymbol{v}$, $\hat{\boldsymbol{v}} := \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$ is

[1] Replication files are available at https://github.com/nikita-tsoy98/simplicity-bias-beyond-linear-replication

the unit direction of $\boldsymbol{v}$, $\boldsymbol{P_v} \coloneqq \mathrm{I} - \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^\mathsf{T}$ is the projector on the space orthogonal to $\boldsymbol{v}$, $[k] \coloneqq \{1, \dots, k\}$, $\mathrm{S}^{d-1} \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^d \,\middle|\, \|\boldsymbol{x}\| = 1\right\}$ is the unit sphere, $\mathrm{D}^d \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^d \,\middle|\, \|\boldsymbol{x}\| < 1\right\}$ is the unit disk.

Objective Denote by $f(\boldsymbol{\theta}, \cdot)$ a network parameterized by $\boldsymbol{\theta}$. The sign of f stands for the classification result. Denote by $D \coloneqq (\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\})_{i=1}^n$ an arbitrary training dataset, such that $\forall i \ \|\boldsymbol{x}_i\| \leq 1$. We consider networks trained to minimize the cross-entropy loss

$$L(\boldsymbol{\theta}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \ell(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i), \text{ where } \ell(z) \coloneqq \ln(1 + \mathrm{e}^{-z}).$$

Architecture We consider two-layer networks

$$f(\boldsymbol{\theta}, \boldsymbol{x}) \coloneqq \sum_{j=1}^{m} u_j \phi(\boldsymbol{v}_j, \boldsymbol{x}),$$

where $\boldsymbol{\theta} \coloneqq (u_1, \dots, u_m, \boldsymbol{v}_1^\mathsf{T}, \dots, \boldsymbol{v}_m^\mathsf{T})^\mathsf{T}$, $u_j \in \mathbb{R}$ and $\boldsymbol{v}_j \in \mathbb{R}^d$, are the network parameters and ϕ is an activation function. We denote $\forall A \subseteq [m] \ \|\boldsymbol{\theta}\|_A \coloneqq \max_{j \in A} \max(|u_j|, \|\boldsymbol{v}_j\|)$.

One of the most commonly used activation functions is ReLU, for which $\phi(\boldsymbol{v}, \boldsymbol{x}) = (\boldsymbol{v}^\mathsf{T}\boldsymbol{x})_+ = \max(\boldsymbol{v}^\mathsf{T}\boldsymbol{x}, 0)$. In our paper, for the technical reasons, we need the activation to be smooth to avoid some degenerate cases in the dynamics of gradient flow. Since this property does not hold for ReLU, we consider a differentiable approximation,

$$\phi(\boldsymbol{v}, \boldsymbol{x}) \coloneqq \phi_{Q,\xi}(\boldsymbol{v}, \boldsymbol{x}) \coloneqq \int_{\mathbb{R}^d} (\boldsymbol{v}^\mathsf{T}(\boldsymbol{x} + \xi\boldsymbol{z}))_+ Q(\mathrm{d}\boldsymbol{z}),$$

where $\xi > 0$, Q is the uniform measure on D^d, and $(z)_+ \coloneqq \max(0, z)$. For the purposes of our analysis, ξ could be set to be much smaller than machine precision. Thus, there is no practical difference between our activation and ReLU. We also note that in our experiments in Sections 6 and 7 we use the usual ReLU activation.

We call a function $f(x, y)$ k-positively homogeneous in x if $f(cx, y) = c^k f(x, y)$ for all vectors x, y and all $c > 0$. Notice that $\phi(\boldsymbol{v}, \boldsymbol{x})$ is 1-positively homogeneous in $\boldsymbol{v}$, i.e., $\forall c > 0 \ \phi(c\boldsymbol{v}, \boldsymbol{x}) = c\phi(\boldsymbol{v}, \boldsymbol{x})$. This property implies that $f(\boldsymbol{\theta}, \boldsymbol{x})$ is 2-positively homogeneous in $\boldsymbol{\theta}$.

Optimization We consider the training of f via gradient flow, $\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = -\nabla L(\boldsymbol{\theta})$, which implies dynamics

$$\begin{aligned} \frac{\mathrm{d}u_j}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^{n} (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i, \\ \frac{\mathrm{d}\boldsymbol{v}_j}{\mathrm{d}t} &= \frac{u_j}{n}\sum_{i=1}^{n} (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i. \end{aligned} \tag{1}$$

We initialize the system with a small weights $\boldsymbol{\theta}(0) = \sigma\boldsymbol{\theta}^0$, where $\sigma \approx 0$, similarly Lyu et al. (2021). For simplicity, we assume that $|u_j^0| = \|\boldsymbol{v}_j^0\|$. The results of Du et al. (2018) imply that $u_j^2 - \|\boldsymbol{v}_j\|^2 = \text{const}$. Then, $\operatorname{sign}(u_j) =: s_j$ is constant (Lemma 1, Boursier et al., 2022).

4. Simplicity Bias for General Data

First, we analyze Equation (1) in the early (Section 4.1) and middle (Section 4.2) training phases, in which $\boldsymbol{\theta}$ remains small or grow to a constant scale, respectively. These phases correspond to Phases 1 and 2 of Lyu et al. (2021).

Key challenge The key challenge in characterizing the features learned throughout training is the lack of universal training invariants to trace. To compensate for this, most of the works that describe features of neural networks make structural assumptions about the dataset, such as linear separability (Lyu et al., 2021), orthogonality or near-orthogonality (e.g., Brutzkus & Globerson, 2019; Phuong & Lampert, 2021; Frei et al., 2023a) or high-dimensionality (e.g., Ba et al., 2022).

In contrast to these works, we do not make structural assumptions about the data. Instead, we exploit that, for small weights, a simpler data-dependent function, G, which does not depend on the number of neurons, can approximate the network dynamics. Specifically, we link the features learned by the original system to the global extrema of G. While our results do not explicitly characterize the learned features, they are sufficient to prove the presence of simplicity bias.

4.1. Feature Learning from a Small Initialization

Disentangling training dynamics First, we informally motivate our approximation of Equation (1) for a small initialization. By the mean value theorem,

$$\begin{aligned} \ell'(f(\boldsymbol{\theta}, \boldsymbol{x})y) - \ell'(0) =& \ell''(\zeta)f(\boldsymbol{\theta}, \boldsymbol{x})y \\ \implies |\ell'(f(\boldsymbol{\theta}, \boldsymbol{x})y) - \ell'(0)| \leq& |f(\boldsymbol{\theta}, \boldsymbol{x})| \sup_{z \in \mathbb{R}} |\ell''(z)|, \end{aligned}$$

for some $\zeta \in [0, f(\boldsymbol{\theta}, \boldsymbol{x})y]$. Since $f(\boldsymbol{\theta}, \boldsymbol{x})$ is 2-homogeneous, we get $|\ell'(f(\boldsymbol{\theta}, \boldsymbol{x})y) - \ell'(0)| = \mathrm{O}(\|\boldsymbol{\theta}\|^2)$. Thus, when $\sigma \approx 0$, Equation (1) behaves similarly to the following system with linearized loss (Maennel et al., 2018),

$$\begin{aligned} \frac{\mathrm{d}u_j^l}{\mathrm{d}t} &\coloneqq G(\boldsymbol{v}_j^l) \coloneqq \frac{1}{n}\sum_{i=1}^{n} (-\ell'(0))\phi(\boldsymbol{v}_j^l, \boldsymbol{x}_i)y_i, \\ \frac{\mathrm{d}\boldsymbol{v}_j^l}{\mathrm{d}t} &\coloneqq u_j^l \nabla G(\boldsymbol{v}_j^l), \end{aligned} \tag{2}$$

where $u_j^l(0) = u_j(0)$ and $\boldsymbol{v}_j^l(0) = \boldsymbol{v}_j(0)$.

Note that the neurons in Equation (2) evolve independently of each other, while the neurons in Equation (1) interact

via $\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i)$. This property significantly facilitates the analysis of Equation (2) compared to the original system.

The following theorem formalizes the link between the two systems and links the features learned by the original dynamics (1) to the global extrema of the function G.

Theorem 4.1 (Proof in Appendix B)**.** *Assume that $\boldsymbol{\theta}$ follows Equation (1), $\forall i\ \|\boldsymbol{x}_i\| \leq 1$, $d \geq 2$, and d is odd. Then $\exists \kappa^* > 0, P \subseteq [m], (\kappa_j > 0, u_j^* \in \mathbb{R}, \hat{\boldsymbol{v}}_j^* \in \mathrm{S}^{d-1})_{j=1}^m$ such that for $\sigma = r^{1+\kappa^*}$, $T_1 \coloneqq \frac{1}{\lambda}\ln\left(\frac{r}{\sigma}\right)$, and $r \to 0$, we get*

$$\begin{aligned} \forall j \in P \qquad & |u_j(T_1) - r u_j^*| \leq \mathrm{O}(r^{1+\kappa^*}), \\ & \|\hat{\boldsymbol{v}}_j(T_1) - \hat{\boldsymbol{v}}_j^*\| \leq \mathrm{O}(r^{\kappa^*}),\ s_j G(\hat{\boldsymbol{v}}_j^*) = \lambda, \\ \forall j \in R \quad & |u_j(T_1)| = \|v_j(T_1)\| \leq \mathrm{O}(r^{1+\kappa_j}), \end{aligned}$$

where $R \coloneqq [m] \setminus P$, $\lambda \coloneqq \max_{\hat{\boldsymbol{v}} \in \mathrm{S}^{d-1}} |G(\hat{\boldsymbol{v}})|$ and G is defined by Equation (2).

Moreover, to ensure that a particular global extrema $\hat{\boldsymbol{v}}^*$ of $|G|$ ($|G(\hat{\boldsymbol{v}}^*)| = \lambda$) is captured ($\exists j \in P : \hat{\boldsymbol{v}}_j^* = \hat{\boldsymbol{v}}^*$) with probability at least $1 - \delta$ over isotropic initialization of $\boldsymbol{\theta}^0$, we need $m = \mathrm{O}(-\ln(\delta))$ neurons, where the constants in the big-O notation depend only on the properties of data.

Remark 4.2. We expect that the result can be extended to the case of even d. However, this extension will require exploiting the concept of o-minimal structures and proving a corresponding Lojasiewicz inequality for the o-minimal structure containing arcsin function (see our proof, Ji & Telgarsky (2020), and Example 1.5 of Loi (2010)). Since such an analysis is not necessarily informative from a machine learning perspective, we stick to d being odd for simplicity.

Discussion Theorem 4.1 suggests that, when we start training from small initialization ($\sigma \to 0$), the neurons either align with the global extrema of G ($j \in P$) or grow very slowly ($j \in R$). In particular, for the neurons in R, $|u_j(T_1)| = \|v_j(T_1)\| = \mathrm{O}(r^{1+\kappa_j})$. Therefore, their contribution to the decision boundary is negligible compared to the *prominent* neurons in P, for which $|u_j(T_1)| = \|\boldsymbol{v}_j(T_1)\| = \Theta(r|u_j^*|) = \Theta(r)$. Thus, at the start of the training, the network exhibits simplicity bias: regardless of the number of neurons, m, only prominent ones contribute to the network's decision boundary. Moreover, these prominent neurons are aligned with the global extrema directions of G, which do not depend on m. When the number of neurons is sufficiently large ($m = \Omega(-\ln(\delta))$), the network will learn all global extrema directions of G, which makes the characterization very precise for small σ.

4.2. Feature Growth to a Constant Scale

Next, we extend our analysis beyond the stage studied in Theorem 4.1 to the point when the network weights reach a constant scale. Specifically, we show that the prominent features preserve their alignment and that the network essentially behaves like a smaller p-neuron network, where p is the number of extrema of G.

Embedding function To formalize our claim, we consider a specific smaller network that describes Equation (1) well. Since the prominent neurons ($j \in P$) cluster around the global extrema of G at the end of the first phase, we can divide them according to their direction

$$\begin{aligned} P = \sqcup_{k=1}^p P_k \text{ s.t. } \forall k \forall i, j \in P_k \hat{\boldsymbol{v}}_i^* = \hat{\boldsymbol{v}}_j^* \\ \wedge \forall k \neq k' \forall i \in P_k, j \in P_{k'}\ \hat{\boldsymbol{v}}_i^* \neq \hat{\boldsymbol{v}}_j^*. \end{aligned}$$

We denote by $\hat{\boldsymbol{v}}_{P_k}^*$ the direction of neurons in P_k and by $s_{P_k} \coloneqq \operatorname{sign}(G(\hat{\boldsymbol{v}}_{P_k}^*))$. Now, consider the following auxiliary system

$$\begin{aligned} \frac{\mathrm{d}u_k^e}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}^e, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_k^e, \boldsymbol{x}_i)y_i, \\ \frac{\mathrm{d}\boldsymbol{v}_k^e}{\mathrm{d}t} &= \frac{u_k^e}{n}\sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}^e, \boldsymbol{x}_i)y_i))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_k^e, \boldsymbol{x}_i)y_i, \end{aligned} \tag{3}$$

where

$$u_k^e(T_1) = s_{P_k} r \sqrt{\sum_{j \in P_k} (u_j^*)^2},\ \boldsymbol{v}_k^e(T_1) = r|u_k^e(T_1)|\hat{\boldsymbol{v}}_{P_k}^*.$$

These equations describe the dynamics of a p-neuron network $\boldsymbol{\theta}^e = (\boldsymbol{v}_1^e, \boldsymbol{v}_2^e, \ldots, \boldsymbol{v}_p^e, u_1^e, u_2^e, \ldots, u_p^e)$, initialized in a way that preserves the alignment and scale of $\boldsymbol{\theta}$.

Our goal will be to show that each neuron in the original network can be approximated by corresponding neuron in the p-neuron network. To do it, below, we define an embedding function (Lyu et al., 2021) that maps the p-neuron network $\boldsymbol{\theta}^e$ to a m-neuron network $\boldsymbol{\theta}^\chi \coloneqq \chi(\boldsymbol{\theta}^e)$

$$\begin{aligned} \forall j \in P_k \quad u_j^\chi &= \frac{r u_j^*}{u_k^e(T_1)} u_k^e, \quad \boldsymbol{v}_j^\chi = \frac{r u_j^*}{u_k^e(T_1)} \boldsymbol{v}_k^e, \\ \forall j \in R \quad u_j^\chi &= 0, \qquad\qquad \boldsymbol{v}_j^\chi = \mathbf{0}. \end{aligned} \tag{4}$$

Propagation of simplicity bias We prove that the original network behaves approximately as an image of the embedding above. We only require a mild assumption on the data that avoids degenerate cases in which a data point perfectly aligns with an extrema of G.

Definition 4.3. A direction $\hat{\boldsymbol{v}}$ is *Δ-regular* if $\forall i |\hat{\boldsymbol{v}}^\mathsf{T} \boldsymbol{x}_i| \geq \Delta$.

The following result holds under the assumption that all global extrema of G are regular. We note that this is a generalization of Assumption 4.5 of Lyu et al. (2021) for the case of non-linearly-separable data.

Theorem 4.4 (Proof in Appendix C)**.** *In the setting of Theorem 4.1, consider Equation (3) and assume that $\forall j \in P\ \hat{\boldsymbol{v}}_j^*$*

is $(\xi + 2\Delta)$*-regular. Then,* $\forall \varepsilon \leq \min\{\Delta, 1/2\}$*, the following holds. First,* $\exists \boldsymbol{\theta}^{\varepsilon,*} \coloneqq \lim_{r\to 0} \boldsymbol{\theta}^e(T_2^\varepsilon)$*, where* $T_2^\varepsilon \coloneqq T_1 + t_4^\varepsilon, t_4^\varepsilon \coloneqq \frac{1}{2\lambda} \ln\Big(\frac{\lambda\varepsilon}{2ar^2\|\boldsymbol{\theta}^*\|_{[m]}^2}\Big)$*, and* $a \coloneqq \frac{m(1+\xi)^2}{4}$*. Second, denote* $\boldsymbol{\theta}^{\chi,\varepsilon,*} \coloneqq \chi(\boldsymbol{\theta}^{\varepsilon,*})$*, then, as* $r \to 0$*, we have*

$$\begin{aligned} \forall j \in P \qquad |u_j(T_2^\varepsilon) - u_j^{\chi,\epsilon,*}| &= \mathrm{O}(r^\kappa + r^2), \\ \|\hat{\boldsymbol{v}}_j(T_2^\varepsilon) - \hat{\boldsymbol{v}}_j^{\chi,\varepsilon,*}\| &= \mathrm{O}(r^\kappa + r^2), \\ \forall j \in R \quad |u_j(T_2^\varepsilon)| = \|\boldsymbol{v}_j(T_2^\varepsilon)\| &= \mathrm{O}(r^{\kappa_j}), \end{aligned}$$

where $\kappa \coloneqq \min\{\kappa^*, \min_{j\in R} \kappa_j\}$.

Moreover, $|u_j^{\chi,\varepsilon,*}| = \Theta(\sqrt{\varepsilon})$ *and* $\forall j \in P \|\hat{\boldsymbol{v}}_j^{\chi,\varepsilon,*} - \hat{\boldsymbol{v}}_j^*\| \leq \varepsilon$.

Remark 4.5. The assumption that the critical directions $\hat{\boldsymbol{v}}^*$ are regular is essentially needed only to show that the Hessian matrix $\nabla^2|G|(\boldsymbol{z})$ is negative semi-definite in some $(\xi + 2\Delta)$-neighborhood of $\hat{\boldsymbol{v}}^*$. Since the Hessian matrix is semi-negative at $\boldsymbol{z} = \hat{\boldsymbol{v}}^*$ and G is twice continuously differentiable, this assumption appears rather mild.

Discussion Similarly to Theorem 4.1, the neurons in R have a negligible effect on the network since u_j and v_j are of smaller magnitude compared to the remaining weights. Theorem 4.4 suggests that the network experiences the simplicity bias not only at the start of the training but also until the weights grow to a constant scale $\Theta(\sqrt{\varepsilon})$. Even in the second phase, the prominent neurons in P stay near the extrema of G, which they learned initially. In addition, $\boldsymbol{\theta}(T_2^\varepsilon) \approx \boldsymbol{\theta}^{\chi,\varepsilon,*}$ implies that the original network is an approximate embedding of the p-neural network above.

Theorems 4.1 and 4.4 show an interesting separation of the training dynamics of two-layer networks trained from small initialization. First, the hidden layer features traverse the unit sphere until some become prominent, capturing a supremum direction of G and aligning with it. Then, the prominent features grow without much change in direction.

5. Extreme Simplicity Bias for Specific Data

Our results so far describe the features learned by two-layer networks in the early stages of training, as the parameters go from small to constant scale. However, as indicated by Lyu et al. (2021); Shah et al. (2020), the simplicity bias might persist not only in the initial stages of training but also when the network reaches perfect accuracy on a training dataset. To test to what extent our mechanism outlined in Section 4 can explain this empirical observation, we extend our analysis to the infinite time training limit.

A key challenge in this setup is the lack of convergence guarantees for non-convex models, which necessitates at least some assumptions on the train data. Given our focus on non-linearly-separable data, we focus on datasets in $\mathbb{R}^d$ that feature an XOR-like pattern in a 2-dimensional subspace as a prime example that breaks the linear separability.

5.1. Data with an XOR-pattern

We consider train data in $\mathbb{R}^d$ that follows an XOR pattern in the 2-dimensional subspace generated by coordinate vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Specifically, the points cluster around four vectors, $\boldsymbol{e}_1, -\boldsymbol{e}_1, \boldsymbol{e}_2, -\boldsymbol{e}_2$. They are symmetric w.r.t. the permutation of the first and second coordinate, the reflection of the first and second coordinate axes, and the reflection through the hyperplane generated by the first and second axes. Points that cluster around the directions $\boldsymbol{e}_1$ and $-\boldsymbol{e}_1$ have positive labels; others have negative labels. Finally, we make an additional assumption on the non-alignment of data points with coordinate axes (which would allow us to apply Theorem 4.4 to this dataset). Notice that similar assumptions appear in Lyu et al. (2021), but in our case the resulting dataset is not linearly separable.

To formalize these assumptions, denote $\boldsymbol{P} \coloneqq \mathrm{I} + (\boldsymbol{e}_2 - \boldsymbol{e}_1)\boldsymbol{e}_1^\mathsf{T} + (\boldsymbol{e}_1 - \boldsymbol{e}_2)\boldsymbol{e}_2^\mathsf{T}$ (permutation of the first and second coordinates), $\forall a \in \{1, 2\} : \boldsymbol{R}_a \coloneqq \mathrm{I} - 2\boldsymbol{e}_a\boldsymbol{e}_a^\mathsf{T}$ (reflection of ath coordinate), $\boldsymbol{R}_r \coloneqq 2(\boldsymbol{e}_1\boldsymbol{e}_1^\mathsf{T} + \boldsymbol{e}_2\boldsymbol{e}_2^\mathsf{T}) - \mathrm{I}$ (reflection of the rest of the coordinates), and $D_{\boldsymbol{x}} \coloneqq \{\boldsymbol{x}_i\}_{i=1}^n$ for $\boldsymbol{x}_i \in \mathbb{R}^d$. Then, the formal assumptions will be the following.

Assumption 5.1. Denote $a(k) \coloneqq k \bmod 2$, $b(k) \coloneqq 2\lfloor \frac{k-1}{2} \rfloor - 1$. There exists $\{S_1, S_2, S_3, S_4\} : \sqcup_k S_k = [n]$ such that the following properties hold.

1. $\exists \delta < \sqrt{2}/2 : \forall i \in S_k \, \|\boldsymbol{x}_i - b(k)\boldsymbol{e}_{a(k)+1}\| \leq \delta \wedge y_i = 1 - 2a(k)$.
2. $\forall a \in \{1, 2, r\} \, \boldsymbol{R}_a D_{\boldsymbol{x}} = D_{\boldsymbol{x}}$.
3. $\boldsymbol{P} D_{\boldsymbol{x}} = D_{\boldsymbol{x}}$.
4. $\exists \Delta > 0 : \forall i, k \, |\boldsymbol{e}_k^\mathsf{T}\boldsymbol{x}_i| \geq \xi + 2\Delta$.

5.2. Initial stages of training

We first use the results of Section 4 to analyze the behavior of a two-layer network trained from small initialization on our XOR-like dataset. To apply Theorem 4.1, we first describe the global extrema of G.

Lemma 5.2 (Proof in Appendix D.1)**.** *If Assumption 5.1 holds and* $\xi + \delta < 1/6$*, the function* $|G|$ *have four extrema directions:* $\boldsymbol{e}_1, -\boldsymbol{e}_1, \boldsymbol{e}_2, -\boldsymbol{e}_2$.

Lemma 5.2 and Theorem 4.1 imply that, at the start of the training, the big neurons will converge in the four directions: $\pm\boldsymbol{e}_1$ and $\pm\boldsymbol{e}_2$. To evaluate the probability of the network capturing all extrema of G, we show the following fact.

Lemma 5.3 (Proof in Appendix D.2)**.** *Assume the setting of Lemma 5.2. The probability of successful initialization that will capture all extrema of* G *is greater than* $(1 - h^m)^4 \geq 1 - 4(3/4)^m(1 + \mathrm{O}(\delta + \xi)) - \mathrm{O}((9/16)^m)$ *for* $m \to \infty$ *and* $\delta + \xi \to 0$*, where* $h \coloneqq 1 - \frac{1}{2}\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\mathrm{S}^{d-1})}$ *and* $A = \big\{\boldsymbol{x} \in \mathrm{S}^{d-1} \,\big|\, \boldsymbol{e}_1^\mathsf{T}\boldsymbol{x} \geq \delta + \xi\big\}$.

The lemmas above and Theorems 4.1 and 4.4 suggest that, with high probability, at the end of Phase 2, $\boldsymbol{\theta}^{\varepsilon,*}$, the four-neuron approximation of the original network, has features aligned with the cluster directions of the data, $\|\hat{\boldsymbol{v}}_k^{\varepsilon,*} - b(k)\boldsymbol{e}_{a(k)+1}\| \leq \varepsilon$.

5.3. Training Dynamics in the Infinite Time Limit

Now we are interested in the training dynamics on the XOR dataset, beyond the stage described by Theorem 4.4 and as $T \to \infty$. To study that, we use the fact that $\boldsymbol{\theta}^e$ provides a good approximation of the original network $\boldsymbol{\theta}$ and formalize this beyond the time T_2^ε used in Theorem 4.4.

The next result proves that the limit dynamics (1) of the original network converge to the same features as those learned by the simpler 4-neuron network from Section 5.2. Our result assumes that the features of the 4-neuron network remain aligned with the extrema of G from Lemma 5.2.

Assumption 5.4. In the setting of Lemma 5.2, the 4-neuron network initialized at $\boldsymbol{\theta}^{\varepsilon,*}$ converges in direction to $\boldsymbol{\theta}^{mm}$, in which $\forall i,j\ \|\boldsymbol{v}_i^{mm}\| = |u_i^{mm}| = |u_j^{mm}|$ and $\forall k\ \hat{\boldsymbol{v}}_k^{mm} = b(k)\boldsymbol{e}_{a(k)+1}$.

We provide experimental and theoretical evidence that this assumption holds in the next subsection. We also note that this assumption concerns the two-layer 4-neuron network and can be checked directly in an experimental manner. In contrast, the result below holds for any sufficiently big two-layer neural networks trained from small initialization.

Lemma 5.5 (Proof in Appendix D.5)**.** *In the setting of Lemma 5.2 under Assumption 5.4, the original network $\boldsymbol{\theta}$ converges in direction to $\chi(\boldsymbol{\theta}^{mm})$ if the initialization scale σ is small:*

$$\exists \sigma^* : \forall \sigma < \sigma^* \lim_{t\to\infty} \left\| \frac{\boldsymbol{\theta}(t)}{\|\boldsymbol{\theta}(t)\|} - \frac{\chi(\boldsymbol{\theta}^{mm})}{\|\chi(\boldsymbol{\theta}^{mm})\|} \right\| = 0.$$

(Notice $\boldsymbol{\theta}(0) = \sigma\boldsymbol{\theta}^0$ and χ depends on $\boldsymbol{\theta}^0$ but not on σ.)

This lemma shows that the conclusions of Theorems 4.1 and 4.4 not only propagate to the later stages of training for the XOR data, but also exacerbate. The network forgets all features except those learned at the beginning of training, causing an extreme simplicity bias (Shah et al., 2020).

Proof sketch Our proof builds upon concepts studied in Lyu & Li 2020. Following their notation, we define the normalized margin $\gamma(\boldsymbol{\theta})$ of f on dataset D by

$$\gamma \coloneqq \min_i f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i / \|\boldsymbol{\theta}\|^2.$$

Notice that, due to 2-homogeneity of f, γ depends only on the direction of $\boldsymbol{\theta}$: $\forall \lambda > 0\ \gamma(\lambda\boldsymbol{\theta}) = \gamma(\boldsymbol{\theta})$. We call a direction that is a (local) solution to the max-margin problem $\max_{\boldsymbol{\theta}} \gamma(\boldsymbol{\theta})$ a (local-)max-margin direction. The main result we build upon in our proof is the following.

Theorem 5.6 (Theorem 5.6, Lyu et al. 2021)**.** *Consider 2-positively-homogeneous network f trained with gradient flow on logistic loss. For any local-max-margin direction, $\hat{\boldsymbol{\theta}}^*$, and $\zeta > 0$, $\exists \omega > 0, \rho \geq 1$ such that for any $\boldsymbol{\theta}^0$ with $\|\boldsymbol{\theta}^0\| \geq \rho$ and $\|\hat{\boldsymbol{\theta}}^0 - \hat{\boldsymbol{\theta}}^*\| \leq \omega$, gradient flow starting with $\boldsymbol{\theta}^0$ directionally converges to some direction $\hat{\boldsymbol{\theta}}$ with the same normalized margin γ as $\hat{\boldsymbol{\theta}}^*$, and $\|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^*\| \leq \zeta$.*

We use this result in the following manner. First, we show that the assumed limit direction of $\boldsymbol{\theta}^e$ is a local-max-margin direction. Second, we employ Theorem 4.4 and classical theorems about continuous dependency on initial conditions for initial value problems to show that our original system will satisfy the conditions of Theorem 5.6. Finally, we apply Theorem 5.6 to prove the desired result.

5.4. Convergence Behavior of Four-Neuron Network

Finally, we provide evidence for Assumption 5.4. Unfortunately, a precise characterization of this smaller network on our dataset is challenging. First, as far as we are aware, general results about convergence to perfect accuracy or zero loss of two-layer networks exist only for 1-neuron networks (Awasthi et al., 2023; Chistikov et al., 2023). Second, even if one can prove that the network converges to zero loss, the additional challenge to analyze the resulting limit direction remains. The only characterization of the limit directions of networks that we are aware of is in terms of the KKT conditions for the dual margin-maximization problem (Lyu & Li, 2020). However, in the general case a direct analysis of the KKT conditions remains intractable.

Thus, we give a theoretical motivation for the proposed direction $\boldsymbol{\theta}^{mm}$ and validate our assumption empirically.

5.4.1. Theoretical Evidence

Convergence to perfect accuracy First, we prove that the network converges to perfect accuracy.

Proposition 5.7 (Proof in Appendix D.3)**.** *Assume that $\max_{i,j} {}^{|u_i^e(0)|}/_{|u_j^e(0)|} \leq 1000$, $\max_k |u_k^e(0)| \in (0.001, 0.5)$, and $\delta + \xi \leq 0.01$. Then there exist a time T^0 such that $\forall i\ f(\boldsymbol{\theta}^e(T^0), \boldsymbol{x}_i) y_i > 4.67 > 0$.*

To comment on the plausibility of the assumptions in this result, first notice that in the setting of Theorem 4.1 neurons evolve almost independently. Thus, due to the symmetry of our dataset, at the end of Phase 1 the scales u_j^* will be sums of m identically distributed values. Then by the law of large numbers they will converge to the same values in the limit $m \to \infty$. Thus, when m is big, at the end of Phase 1, the ratios ${}^{|u_i^e(0)|}/_{|u_j^e(0)|}$ will be close to one. Since in Phase 2 the scales of neurons and the deviation of features is small, we expect these ratios to increase by no more than $1 + \mathrm{O}(\varepsilon)$, implying that these ratios again will be close to one.

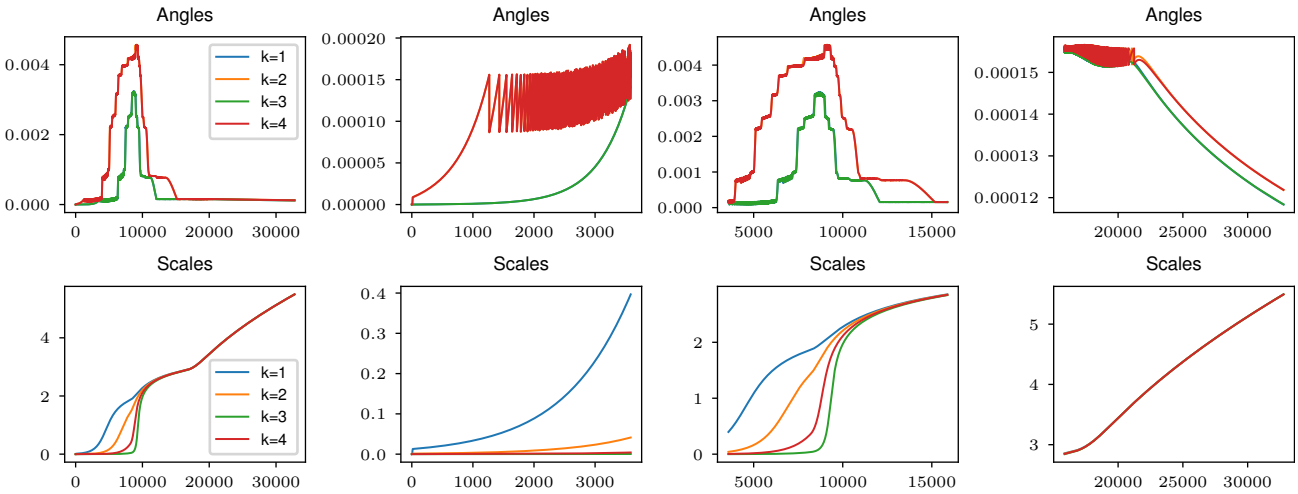

Figure 1. Evolution of 4-neuron network initialized at $(u_1^e(0), u_2^e(0), u_3^e(0), u_4^e(0)) = (10^{-4}, -10^{-5}, 10^{-7}, -10^{-6})$. The first column depicts the whole training process. We additionally depict different stages of training process for visual convenience: the second column depicts the first 3584 training epochs; the third column depicts the epochs from 3584 to 15872; the last column depicts training after the 15872th epoch. Notice that $\alpha_1 \approx \alpha_3$ and $\alpha_2 \approx \alpha_4$, where α_i are the angles between the network features and the cluster directions.

The assumption that $\max_k |u_k^e(0)| \in (0.001, 0.5)$ is implied by the property $u_j^{\varepsilon,*} = \Theta(\sqrt{\varepsilon})$. If we assume that $\Delta \approx 0.001$ and choose $\varepsilon = \Delta$, we can expect that $\boldsymbol{\theta}^e$ will have the desired scale. Finally, we assume $\delta + \xi \leq 0.01$ for technical reasons related to the proof technique. We expect that the property will hold for wider ranges of δ.

Implicit bias at the end of training The previous result about perfect accuracy also suggests that the network may achieve small loss at the end of training. Then, by Theorem 4.4 of Lyu & Li (2020) and Theorem 3.1 of Ji & Telgarsky (2020), the network converges to some KKT point of dual margin-maximization problem

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|^2 \text{ s.t. } \forall i \; f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i \geq 1.$$

Under the additional assumption that the features of this margin direction are $(\xi + 2\Delta)$-regular for some $\Delta > 0$, a slight modification of the results of Vardi et al. (2022) implies that this direction is a local-max-margin direction. Finally, we show that $\boldsymbol{\theta}^{mm}$ is one strict local-max-margin direction, which motivates our hypothesis.

Proposition 5.8 (Proof in Appendix D.4)**.** *Assume the setting of Lemma 5.2. Then, the direction of $\boldsymbol{\theta}^{mm}$ is a strict local-max-margin in the weight space of the 4-neuron network, while its embedding, $\chi(\boldsymbol{\theta}^{mm})$ is a strict local-max-margin in the original weight space.*

5.4.2. Empirical Evidence

Here we present experimental evidence for Assumption 5.4. To this end, we construct a random dataset in $\mathbb{R}^2$ that will satisfy Assumption 5.1 with $\delta \leq 0.01$ and train a four neuron ReLU network using gradient descent for different initializations, in which the neurons are aligned with cluster directions, corresponding to the setting of Lemma 5.2. Specifically, after we pick initialization scales $(u_1^e(0), u_2^e(0), u_3^e(0), u_4^e(0))$, we initialize the first layer as $\boldsymbol{v}_k^e(0) = u_k^e(0) \boldsymbol{e}_{a(k)+1}$ and train our network using plain gradient descent.

We used the following adaptive learning rate schedule η_t. During the first part of training, which corresponds to the evolution initialized at the limit point from Theorem 4.1 to the limit point in Theorem 4.4, the gradients are very stable. Therefore, we use constant-scale learning rates $\eta_t = 4$ for $t < 12$ and $\eta_t = 2^{-7}$ for $12 \leq t < 2^{13}$. The next part corresponds to the training initialized at the limit point from Theorem 4.4. At around $t = 2^{13}$ all features reach constant scales and the cross-entropy loss starts to dump gradients. This allows us to progressively increase the learning rate, in order to speed up the simulation, in the third part of schedule, setting $\eta_t = 2^{-7}\left(1 + 2^5\left(\frac{t}{2^{13}} - 1\right)\right)$ for $2^{13} \leq t < 2^{14}$. Finally, at the end of the third part, the cross-entropy loss causes gradients to decay exponentially and the training process almost stops. To combat this and simulate the later stages of training, we use an exponential learning rate $\eta_t = 2^{-7}\left(1 + 2^{5 + \frac{t - 2^{14}}{2^9}}\right)$ for $t \geq 2^{14}$.

Figure 1 depicts the evolution of the 4-neuron network. Here, plots titled "Angles" depict the signed angles between the network features and the cluster directions. And plots titled "Scales" depict $\|\boldsymbol{v}_k^e\|$. (See more experiments in Appendix E.) As we can see, eventually the scales of the neurons became almost identical and the network start to converge to the desired local-max-margin direction $\boldsymbol{\theta}^{mm}$ (forth

column), empirically supporting our assumption.

6. Empirical Validation of Results

In this section, we empirically validate the predictions of Theorems 4.1, 4.4, and Lemma 5.5. We consider a skewed XOR-like dataset in $\mathbb{R}^2$, similar to the dataset covered by Assumption 5.1, but in which the angle between cluster directions can be arbitrary. We consider two angles: $\alpha = \pi/2$ and $\alpha = \pi/3$. The experiments for $\alpha = \pi/2$ seek to verify our predictions from Section 5, while the experiments for $\alpha = \pi/3$ test if these predictions transfer to non-orthogonal cases. We train a two-layer neural network with $m = 2^{12}$ randomly initialized neurons with initialization scale $\sigma = 2^{-7}$. We train this network for 2^{13} epochs using gradient descent with $\texttt{lr} = 2^{-4}$. At that stage, we observed that the network essentially converged.

When $\alpha = \pi/2$, the scale of the first layer grows from 0.3 to 5.5 during the training (Figure 2, first row, left). At the same time, almost all neurons end up aligned in four directions (Figure 2, second row, left): $0, \pi/2, \pi, -\pi/2$, which are the global extrema of the function G. Similarly, when $\alpha = \pi/3$ the scale of the first layer grows from 0.3 to 5.9 during the training (Figure 2, first row, right). At the same time, almost all neurons end up aligned in four directions (Figure 2, second row, right): $-\pi/6, \pi/2, 5\pi/6, -\pi/2$, which again are the global extrema of the function G. In both cases, the few non-aligned neurons have smaller weights than aligned ones (Figure 2, second row). Therefore, our theory indeed predicts the alignment well in this setting.

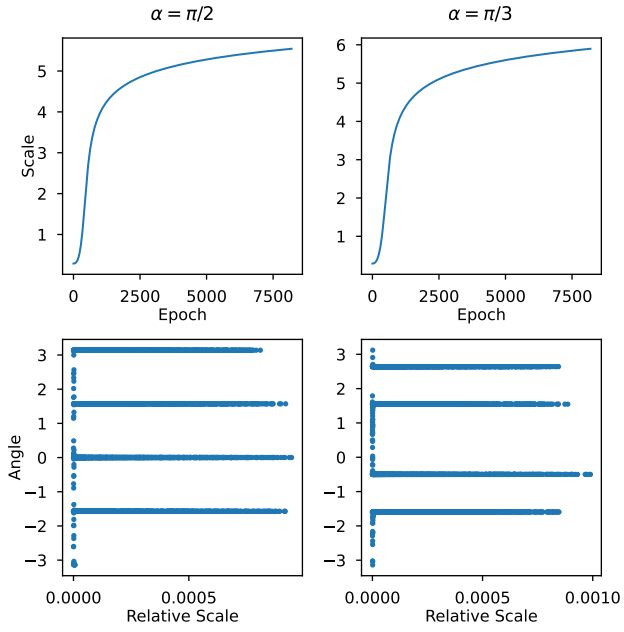

Figure 2. Empirical evidence of simplicity bias on XOR-like data. Relative scale in the second row defined as $\|\boldsymbol{v}_i\|^2 / \sum_{j=1}^m \|\boldsymbol{v}_j\|^2$.

7. Effects of Extreme Simplicity Bias

Finally, we test the possible implications of our theoretical characterization of simplicity bias on real-world datasets. Our previous results suggest that simplicity bias disproportionally amplifies "simple" features that are very informative about the target. Additionally, in the case of extreme simplicity bias, if the "simple" features are enough to classify the training set perfectly, the network effectively "forgets" all features except for the "simple" ones.

The latter fact suggests that if spurious features are enough to classify the target in the train distribution, then the network should progressively lose its ability to fine-tune out of distribution. We test this hypothesis on the MNIST-CIFAR-10 domino dataset proposed by Shah et al. (2020).

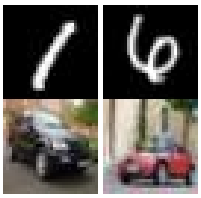

Figure 3. Examples of domino with a car (class 1 in CIFAR-10) in train (left) and test (right) dataset. Notice that the top MNIST image is an image of 1 only for the train data.

Setup The dataset contains vertical concatenations of MNIST and CIFAR-10 images and uses CIFAR-10 image labels. On the train distribution, the MNIST and CIFAR-10 labels of the images are perfectly correlated: digit i from MNIST will always be concatenated with the image from class i in CIFAR-10. On the test distribution, MNIST and CIFAR-10 labels are not correlated, i.e., the concatenations of images from different classes are random. Thus, MNIST images represent "simple" spurious features that can be used to perfectly classify the data. (See Figure 3.)

We fit a ResNet-18 model (He et al., 2016) on this domino dataset and track its parameter trajectory. We use the standard PyTorch initialization, multiplied by 2^{-5} to mimic the small initialization regime studied in the previous sections. We train this model for 2^8 epochs using the usual SGD optimizer with Nesterov momentum and linear schedule with warm-up (similar training recipe was used by Jain et al., 2023). Periodically during training, we apply the current model to our test set and extract its last layer features, X, on test data. We normalize these last layer features (to make feature scales comparable across different epochs), $X^{\text{norm}} = \frac{X}{\sqrt{\frac{1}{n}\sum_{i=1}^n \sum_{j=1}^k X_{ij}^2}}$, where n is the number of test

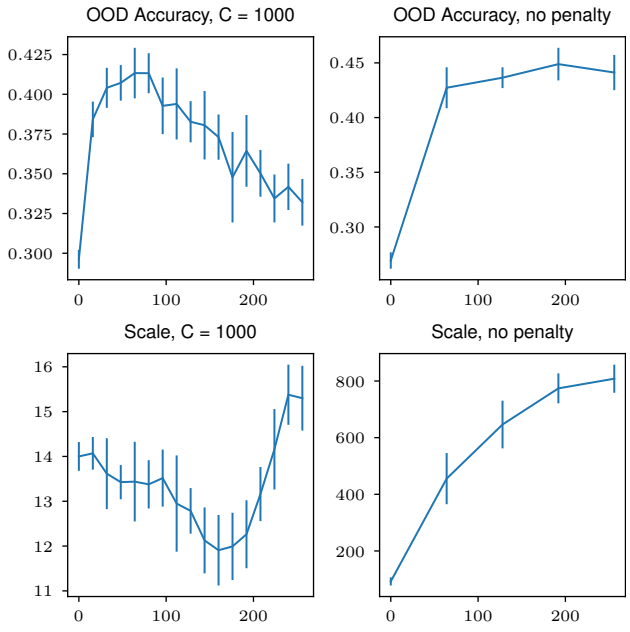

Figure 4. Accuracy and scale of the logistic regression on the validation part of the OOD test set (y-axis) vs. the training epoch at which the ResNet features are extracted (x-axis).

samples, k is the number of features. We then use these normalized last layer features to train simple logistic regression model. We train two types of linear models: with small regularization (with inverse regularization strength equal to 1000) and without regularization. We plot the accuracy on the validation part of the test distribution and the quadratic mean of the regression coefficients on Figure 4. (See training details in Appendix F.1.)

Analysis First, we can see that the model can not make reliable predictions on the test set, peaking in accuracy at about 45%. This is in contrast to training the model using the same recipe on plain CIFAR-10 data, where it achieves around 90% accuracy. At the same time, using the MNIST labels on the test set, we get around 99% accuracy (see Figure 8 in Appendix F.2; in line with the results of Shah et al., 2020; Hermann & Lampinen, 2020). These facts suggest that the network indeed experiences a simplicity bias and mainly relies on "simple" MNIST features for prediction. Second, as we can see, the OOD accuracy increases fast at the beginning of training. This may indicate that even simple MNIST features are better for classification of CIFAR-10 data compared to random features. Finally, we can see that the OOD accuracy does not increase in the latter stages of training, indicating that the simplicity bias persists even if we train longer. In Appendix F.2, we present additional setups, where we do not scale the initialization (Figure 9) or break the perfect correlation between MNIST and CIFAR-10 labels on the train set (Figure 10). In both cases, we also observe the presence of simplicity bias.

Finally, we can see some evidence in favor of the extreme simplicity bias mechanism described in Section 5. For the small regularization setup, we observe a significant drop in OOD accuracy when using features from later training stages (the drop between epochs 64 and 256 is approximately $8.13\% \pm 0.80\%$, which gives a p-value around 10^{-6} according to the t-test). This result suggests a potential presence of extreme simplicity bias, which impedes the learning of complex features and makes the network forget the initial random features. For the no regularization setup, we can see that the network does not lose OOD accuracy, but the last-layer weights become much bigger. This result is again consistent with the presence of extreme simplicity bias, as our proposed mechanism does not force the network to forget "non-simple" features directly. Instead, it makes "simple" features grow faster, so the regression can only use "non-simple" features by applying huge weights (approximately $50\times$ bigger than the regularized regression).

Discussion We can draw three practical conclusions from these experiments. Suppose the training data can be classified using a simple heuristic. Then, it might be beneficial to train networks in a more "lazy regime" (Chizat et al., 2019), which forces the network to remember randomly initialized features. Similarly, one could apply early stopping of training. In this way, one could benefit from new features learned from the training data without losing access to potentially beneficial random features. Finally, since simplicity bias works by making "simple" features disproportionally large, the natural countermeasure is to use additional normalization layers, which have already been proven effective for few-shot transfer (e.g., Perez et al., 2018). At the same time, the "lazy regime" or early stopping might harm in-distribution generalization performance (Telgarsky, 2023; Lyu et al., 2024). Thus, a practical method would need to trade off the benefits of the "lazy regime" or early stopping for the OOD task to their potential harm to generalization.

8. Conclusion

We characterize simplicity bias beyond linearly separable datasets as the tendency of features to cluster in several directions, which do not depend on the network size. We also demonstrate that extreme simplicity bias may appear for non-linearly-separable datasets and observe it experimentally on image data. We see our results as an indication that the characterization of simplicity bias is a crucial step toward improving out-of-distribution generalization.

Acknowledgments

This research was partially funded from the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure). The authors thank Ivan Kirev and Kristian Minchev for their helpful feedback and discussions on this work.

Impact Statement

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

References

Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 09–15 Jun 2019.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A Closer Look at Memorization in Deep Networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 06–11 Aug 2017.

Awasthi, P., Tang, A., and Vijayaraghavan, A. Agnostic Learning of General ReLU Activation Using Gradient Descent. In *The Eleventh International Conference on Learning Representations*, 2023.

Ba, J., Erdogdu, M. A., Suzuki, T., Wang, Z., Wu, D., and Yang, G. High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 37932–37946. Curran Associates, Inc., 2022.

Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., and Kritchman, S. Frequency Bias in Neural Networks for Input of Non-Uniform Density. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 685–694. PMLR, 13–18 Jul 2020.

Boursier, E. and Flammarion, N. Early alignment in two-layer networks training is a two-edged sword, 2024.

Boursier, E., Pillaud-Vivien, L., and Flammarion, N. Gradient flow dynamics of shallow ReLU networks for square loss and orthogonal inputs. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20105–20118. Curran Associates, Inc., 2022.

Bowman, B. and Montufar, G. Implicit Bias of MSE Gradient Optimization in Underparameterized Neural Networks. In *International Conference on Learning Representations*, 2022.

Brutzkus, A. and Globerson, A. Why do Larger Models Generalize Better? A Theoretical Perspective via the XOR Problem. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 822–830. PMLR, 09–15 Jun 2019.

Brutzkus, A., Globerson, A., Malach, E., and Shalev-Shwartz, S. SGD Learns Over-parameterized Networks that Provably Generalize on Linearly Separable Data. In *International Conference on Learning Representations*, 2018.

Chistikov, D., Englert, M., and Lazic, R. Learning a Neuron by a Shallow ReLU Network: Dynamics and Implicit Bias for Correlated Inputs. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 23748–23760. Curran Associates, Inc., 2023.

Chizat, L. and Bach, F. On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Chizat, L., Oyallon, E., and Bach, F. On Lazy Training in Differentiable Programming. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Du, S. S., Hu, W., and Lee, J. D. Algorithmic Regularization in Learning Deep Homogeneous Models: Layers are Automatically Balanced. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Englert, M. and Lazic, R. Adversarial Reprogramming Revisited. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 28588–28600. Curran Associates, Inc., 2022.

Frei, S., Chatterji, N. S., and Bartlett, P. L. Random Feature Amplification: Feature Learning and Generalization in Neural Networks. *Journal of Machine Learning Research*, 24(303):1–49, 2023a.

Frei, S., Vardi, G., Bartlett, P., Srebro, N., and Hu, W. Implicit Bias in Leaky ReLU Networks Trained on High-Dimensional Data. In *The Eleventh International Conference on Learning Representations*, 2023b.

Ge, R., Ren, Y., Wang, X., and Zhou, M. Understanding Deflation Process in Over-parametrized Tensor Decomposition. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1299–1311. Curran Associates, Inc., 2021.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Hartman, P. *Ordinary Differential Equations*. Society for Industrial and Applied Mathematics, second edition, 2002. doi: 10.1137/1.9780898719222.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Hermann, K. and Lampinen, A. What shapes feature representations? exploring datasets, architectures, and training. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9995–10006. Curran Associates, Inc., 2020.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8.

Jain, S., Lawrence, H., Moitra, A., and Madry, A. Distilling Model Failures as Directions in Latent Space. In *The Eleventh International Conference on Learning Representations*, 2023.

Ji, Z. and Telgarsky, M. Directional convergence and alignment in deep learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17176–17186. Curran Associates, Inc., 2020.

Kalimeris, D., Kaplun, G., Nakkiran, P., Edelman, B., Yang, T., Barak, B., and Zhang, H. SGD on Neural Networks Learns Functions of Increasing Complexity. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alche-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Kou, Y., Chen, Z., and Gu, Q. Implicit Bias of Gradient Descent for Two-layer ReLU and Leaky ReLU Networks on Nearly-orthogonal Data. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 30167–30221. Curran Associates, Inc., 2023.

Li, Y., Ma, T., and Zhang, H. R. Learning Over-Parametrized Two-Layer Neural Networks beyond NTK. In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 2613–2682. PMLR, 09–12 Jul 2020.

Loi, T. L. Lecture 1: O-minimal structures. In *The Japanese-Australian Workshop on Real and Complex Singularities: JARCS III*, volume 43, pp. 19–31. Australian National University, Mathematical Sciences Institute, 2010.

Luo, T., Ma, Z., Xu, Z.-Q. J., and Zhang, Y. Theory of the Frequency Principle for General Deep Neural Networks. *CSIAM Transactions on Applied Mathematics*, 2(3):484–507, 2021a. ISSN 2708-0579. doi: 10.4208/csiam-am.SO-2020-0005.

Luo, T., Xu, Z.-Q. J., Ma, Z., and Zhang, Y. Phase Diagram for Two-layer ReLU Neural Networks at Infinite-width Limit. *Journal of Machine Learning Research*, 22(71):1–47, 2021b.

Lyu, K. and Li, J. Gradient Descent Maximizes the Margin of Homogeneous Neural Networks. In *International Conference on Learning Representations*, 2020.

Lyu, K., Li, Z., Wang, R., and Arora, S. Gradient Descent on Two-layer Nets: Margin Maximization and Simplicity Bias. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12978–12991. Curran Associates, Inc., 2021.

Lyu, K., Jin, J., Li, Z., Du, S. S., Lee, J. D., and Hu, W. Dichotomy of early and late phase implicit biases can provably induce grokking. In *The Twelfth International Conference on Learning Representations*, 2024.

Maennel, H., Bousquet, O., and Gelly, S. Gradient Descent Quantizes ReLU Network Features, 2018.

McCoy, T., Pavlick, E., and Linzen, T. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018. doi: 10.1073/pnas.1806579115.

Min, H., Mallada, E., and Vidal, R. Early Neuron Alignment in Two-layer ReLU Networks with Small Initialization. In *The Twelfth International Conference on Learning Representations*, 2024.

Morwani, D., Batra, J., Jain, P., and Netrapalli, P. Simplicity Bias in 1-Hidden Layer Neural Networks. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8048–8075. Curran Associates, Inc., 2023.

Pellegrini, F. and Biroli, G. An analytic theory of shallow networks dynamics for hinge loss classification. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5356–5367. Curran Associates, Inc., 2020.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11671.

Phuong, M. and Lampert, C. H. The inductive bias of ReLU networks on orthogonally separable data. In *International Conference on Learning Representations*, 2021.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the Spectral Bias of Neural Networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019.

Refinetti, M., Ingrosso, A., and Goldt, S. Neural networks trained with SGD learn distributions of increasing complexity. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28843–28863. PMLR, 23–29 Jul 2023.

Safran, I., Vardi, G., and Lee, J. D. On the Effective Number of Linear Regions in Shallow Univariate ReLU Networks: Convergence Guarantees and Implicit Bias. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32667–32679. Curran Associates, Inc., 2022.

Sarussi, R., Brutzkus, A., and Globerson, A. Towards Understanding Learning in Neural Networks with Linear Teachers. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9313–9322. PMLR, 18–24 Jul 2021.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The Pitfalls of Simplicity Bias in Neural Networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9573–9585. Curran Associates, Inc., 2020.

Telgarsky, M. Feature selection and low test error in shallow low-rotation ReLU networks. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Vardi, G., Shamir, O., and Srebro, N. On Margin Maximization in Linear and ReLU Networks. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 37024–37036. Curran Associates, Inc., 2022.

Wang, M. and Ma, C. Understanding Multi-phase Optimization Dynamics and Rich Nonlinear Behaviors of ReLU Networks. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35654–35747. Curran Associates, Inc., 2023.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. Regularization Matters: Generalization and Optimization of Neural Nets v.s. their Induced Kernel. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alche-Buc, F., Fox, E., and Garnett,

R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Xu, W. and Du, S. Over-Parameterization Exponentially Slows Down Gradient Descent for Learning a Single Neuron. In Neu, G. and Rosasco, L. (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 1155–1198. PMLR, 12–15 Jul 2023.

Xu, Z.-Q. J., Zhang, Y., and Xiao, Y. Training Behavior of Deep Neural Network in Frequency Domain. In Gedeon, T., Wong, K. W., and Lee, M. (eds.), *Neural Information Processing*, pp. 264–274, Cham, 2019. Springer International Publishing. ISBN 978-3-030-36708-4.

Zhou, H., Qixuan, Z., Luo, T., Zhang, Y., and Xu, Z.-Q. Towards Understanding the Condensation of Neural Networks at Initial Training. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 2184–2196. Curran Associates, Inc., 2022.

Supplementary Material

The supplementary material is structured as follows.

- Appendix A explains some additional notation used in proofs.
- Appendix B contains the proof of Theorem 4.1.
- Appendix C contains the proof of Theorem 4.4.
- Appendix D contains the proofs of the results from Section 5.
- Appendix E contains the additional experimental results for Section 5.
- Appendix F.1 contains additional experimental results for Section 7.

A. Additional Notation

Denote $\boldsymbol{g}(\boldsymbol{v}) \coloneqq \nabla_{\boldsymbol{v}} G(\boldsymbol{v})$.

B. Proof of Theorem 4.1

We will prove the theorem in several stages. First, we will analyze Equation (2). Then, we couple Equations (1) and (2). This will allow us to prove the desired result.

B.1. Analysis of Equation (2)

B.1.1. Directional Convergence

Now, we want to apply the results of Ji & Telgarsky (2020) about the directional convergence to Equation (2).

Dynamics of neuron direction Notice that

$$\frac{\mathrm{d}\hat{\boldsymbol{v}}}{\mathrm{d}\boldsymbol{v}} = \frac{\mathrm{d}\boldsymbol{v}/\|\boldsymbol{v}\|}{\mathrm{d}\boldsymbol{v}} = \frac{\|\boldsymbol{v}\|^2 - \boldsymbol{v}\boldsymbol{v}^\mathsf{T}}{\|\boldsymbol{v}\|^3}. \tag{5}$$

Thus,

$$\frac{\mathrm{d}\hat{\boldsymbol{v}}_j^l}{\mathrm{d}t} = \frac{\mathrm{d}\hat{\boldsymbol{v}}_j^l}{\mathrm{d}\boldsymbol{v}_j^l}\frac{\mathrm{d}\boldsymbol{v}_j^l}{\mathrm{d}t} = s_j \boldsymbol{P}_{\hat{\boldsymbol{v}}^l} \boldsymbol{g}(\hat{\boldsymbol{v}}_j^l)$$

and $\hat{\boldsymbol{v}}^l(0) = \frac{\boldsymbol{v}^0}{\|\boldsymbol{v}^0\|}$ does not depend on σ.

Consider the auxiliary system

$$\frac{\mathrm{d}\boldsymbol{r}_j}{\mathrm{d}t} = s_j \nabla_{\boldsymbol{r}} G(\hat{\boldsymbol{r}}_j) = \frac{s_j}{\|\boldsymbol{r}_j\|} \boldsymbol{P}_{\boldsymbol{r}_j} \boldsymbol{g}(\hat{\boldsymbol{r}}_j), \tag{6}$$

where $\boldsymbol{r}_j(0) = \hat{\boldsymbol{v}}_j^l(0)$. Notice that

$$\boldsymbol{r}_j \frac{\mathrm{d}\boldsymbol{r}_j}{\mathrm{d}t} = 0 \wedge \|\boldsymbol{r}_j(0)\| = 1 \implies \frac{\mathrm{d}\boldsymbol{r}_j}{\mathrm{d}t} = s_j \boldsymbol{P}_{\hat{\boldsymbol{r}}_j} \boldsymbol{g}(\hat{\boldsymbol{r}}_j).$$

Thus, $\forall t\ \boldsymbol{r}_j(t) = \hat{\boldsymbol{v}}_j^l(t)$: we could use $\boldsymbol{r}_j$ instead of $\hat{\boldsymbol{v}}_j^l$ in further derivations.

We want to show the convergence of $\boldsymbol{r}_j$ similarly to Theorem 3.1 of Ji & Telgarsky (2020). To do it, we want to use Lemma B.11 of Ji & Telgarsky (2020).

Lemma B.1 (Lemma B.11 of Ji & Telgarsky 2020)**.** *Given a locally Lipschitz definable function $f\colon A \to \mathbb{R}$ with an open bounded domain A, there exists $\nu > 0$ and a definable desingularizing function ψ on $[0, \nu)$ such that*

$$\forall \boldsymbol{v} \in f^{-1}((0,\nu))\ \psi'(f(\boldsymbol{v}))\|\nabla_{\boldsymbol{v}} f(\boldsymbol{v})\| \geq 1.$$

To apply it, we need to show that $G(\hat{\boldsymbol{r}})$ is locally Lipschitz and definable. And since

$$G(\hat{\boldsymbol{v}}) = \frac{-\ell'(0)}{n} \sum_{i=1}^{n} \phi(\hat{\boldsymbol{v}}, \boldsymbol{x}_i) y_i,$$

we only need to show the desired properties for ϕ.

Proposition B.2. *Denote* $c \coloneqq \max\Big(-1, \min\Big(1, \frac{\boldsymbol{v}^\mathsf{T}\boldsymbol{x}}{\xi\|\boldsymbol{v}\|}\Big)\Big)$ *and the class of all polynomials of* $x_1, \dots, x_p$ *by* $\mathrm{Poly}(x_1, \dots, x_p)$. *We have*

$$\phi(\boldsymbol{v}, \boldsymbol{x}) = \int_{-c}^{1} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)} \,\mathrm{d}a \in \xi\|\boldsymbol{v}\|\mathrm{Poly}(c, \sqrt{1-c^2}).$$

Proof. By the definition

$$\begin{aligned}
\phi(\boldsymbol{v}, \boldsymbol{x}) &= \int_{\|\boldsymbol{z}\|\le 1} (\boldsymbol{v}^\mathsf{T}(\boldsymbol{x} + \xi\boldsymbol{z}))_+ \frac{\mathrm{d}\boldsymbol{z}}{\mathrm{Vol}(\mathrm{D}^d)} = \int_{-1}^{1} \int_{\|\boldsymbol{b}\|^2 \le 1-a^2} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)_+ \frac{\mathrm{d}\boldsymbol{b}\,\mathrm{d}a}{\mathrm{Vol}(\mathrm{D}^d)} \\
&= \int_{-1}^{1} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)_+ (1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)} \,\mathrm{d}a = \int_{-c}^{1} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)} \,\mathrm{d}a \\
&= \int_{-\arcsin(c)}^{\pi/2} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|\sin(\varphi))\cos(\varphi)^d \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)} \,\mathrm{d}\varphi,
\end{aligned}$$

where $a \coloneqq \hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{z}$ and $\boldsymbol{b} \coloneqq \boldsymbol{P}_{\boldsymbol{v}}\boldsymbol{z}$. To prove that the right hand part is a polynomial in c and $\sqrt{1-c^2}$, notice the following identities.

$$\int_{-\arcsin(c)}^{\pi/2} \sin(\varphi)\cos(\varphi)^d \,\mathrm{d}\varphi = \frac{\cos(\arcsin(c))^{d+1}}{d+1} = \frac{(1-c^2)^{\frac{d+1}{2}}}{d+1}.$$

$$\begin{aligned}
\int_{-\arcsin(c)}^{\pi/2} \cos(\varphi)^d \,\mathrm{d}\varphi &= \int_{-\arcsin(c)}^{\pi/2} \left(\frac{\mathrm{e}^{\mathrm{i}\varphi} + \mathrm{e}^{-\mathrm{i}\varphi}}{2}\right)^d \mathrm{d}\varphi = \sum_{k=0}^{\frac{d-1}{2}} \binom{d}{k} \frac{\mathrm{e}^{i(d-2k)\varphi} - \mathrm{e}^{-i(d-2k)\varphi}}{i(d-2k)2^d}\Bigg|_{-\arcsin(c)}^{\pi/2} \\
&= \sum_{k=0}^{\frac{d-1}{2}} \binom{d}{k} \frac{\sin((d-2k)\arcsin(c)) + \sin(\frac{(d-2k)\pi}{2})}{(d-2k)2^{d-1}}.
\end{aligned}$$

Finally, notice that

$$\sin(k\arcsin(c)) = \mathrm{Im}(\mathrm{e}^{\mathrm{i}k\arcsin(c)}) = \mathrm{Im}((\sqrt{1-c^2} + \mathrm{i}c)^k) \in \mathrm{Poly}(c, \sqrt{1-c^2}).$$

Thus, $\phi(\boldsymbol{v}, \boldsymbol{x}) \in \xi\|\boldsymbol{v}\|\mathrm{Poly}(c, \sqrt{1-c^2})$. □

Remark B.3. Notice that, for even d, the activation function will have a term proportional to $\arcsin(c)$ in addition to the polynomial in c. While $\arcsin(c)$ is not definable in the smallest structure on $(\mathbb{R}, +, \times)$ (see discussion below), this function is definable on the structure $(\mathbb{R}, +, \times, \mathcal{A})$, where $\mathcal{A}$ is the class of all restricted functions on $[-1, 1]^n$ (Example 1.5 of Loi, 2010).

Definability of ϕ To prove that ϕ is definable in the o-minimal structure on $(\mathbb{R}, +, \times)$, we will employ the following properties of definable functions (Appendix B.1, Ji & Telgarsky, 2020).

1. Let $f, g \colon D \to \mathbb{R}$ be definable functions. Then $\forall \alpha, \beta \in \mathbb{R}$ $\alpha f + \beta g$ and fg are definable. If $g \neq 0$, f/g is definable. If $f \ge 0$, $\forall l \in \mathbb{N}$ $\sqrt[l]{f}$ is definable.
2. Let $f \colon D \to \mathbb{R}^d$. Then f is definable iff all coordinate projections of f are definable.
3. Composition of definable functions is definable.
4. Any coordinate permutation of a definable set is definable.

5. The image and pre-image of a definable set by a definable function is definable.

6. Any combination of finitely many definable functions with disjoint domains is definable. For example, the point-wise maximum and minimum of definable functions are definable.

7. Polynomial functions are definable.

Now, notice that $\boldsymbol{v}^\mathsf{T}\boldsymbol{x}$ is a linear function, $\|\boldsymbol{v}\|$ is a square root of a quadratic polynomial. Thus, their ratio $\frac{\boldsymbol{v}^\mathsf{T}\boldsymbol{x}}{\xi\|\boldsymbol{v}\|}$ is definable. Therefore, c is definable since it a composition of the maxima and minima of definable functions. Finally, $\phi(\hat{\boldsymbol{v}}, \boldsymbol{x})$ is definable since it is a polynomial of two definable functions c and $\sqrt{1-c^2}$.

Local Lipschitzness To prove that $\phi(\hat{\boldsymbol{v}}, \boldsymbol{x})$ is locally Lipschitz on A, we prove a stronger property that $\phi(\boldsymbol{v}, \boldsymbol{x})$ is twice continuously differentiable on A, which would be useful later. Notice that

$$\phi(\boldsymbol{v}, \boldsymbol{x}) = \int_{-c}^{1} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a.$$

Denote $H(z, \boldsymbol{v}, \boldsymbol{x}) \coloneqq \int_{-z}^{1} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a$. Notice that $\phi(\boldsymbol{v}, \boldsymbol{x}) = H(c, \boldsymbol{v}, \boldsymbol{x})$.

By the chain rule for total derivative and the Leibniz integral rule (notice that $\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a$ is smooth for $\boldsymbol{v} \neq \boldsymbol{0}$), when $\frac{\boldsymbol{v}^\mathsf{T}\boldsymbol{x}}{\xi\|\boldsymbol{v}\|} \notin \{-1, 1\}$, we get

$$\begin{aligned}
\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}, \boldsymbol{x}) =& \frac{\partial H}{\partial c}\nabla_{\boldsymbol{v}} c + \nabla_{\boldsymbol{v}} H \\
=& (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} - \xi\|\boldsymbol{v}\|c)(1-c^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\nabla_{\boldsymbol{v}} c + \int_{-c}^{1} \left(\boldsymbol{x} + \xi\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}a\right)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a \\
=& \int_{-c}^{1} \left(\boldsymbol{x} + \xi\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}a\right)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a.
\end{aligned}$$

Thus, $\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}, \boldsymbol{x})$ exists and is continuous when $\frac{\boldsymbol{v}^\mathsf{T}\boldsymbol{x}}{\xi\|\boldsymbol{v}\|} \notin \{-1, 1\}$. When $\frac{\boldsymbol{v}^\mathsf{T}\boldsymbol{x}}{\xi\|\boldsymbol{v}\|} \in \{-1, 1\}$ (which implies $c \in \{-1, 1\}$), we could find the derivative by the definition. Notice that c is locally Lipschitz in $\boldsymbol{v}$ when $\boldsymbol{v} \neq \boldsymbol{0}$ since it is a clipping of a continuously differentiable function. Now, using the Leibniz rule again, we get

$$\begin{aligned}
&\lim_{\mathrm{d}\boldsymbol{v}\to\boldsymbol{0}} \frac{\phi(\boldsymbol{v}+\mathrm{d}\boldsymbol{v}, \boldsymbol{x}) - \phi(\boldsymbol{v}, \boldsymbol{x})}{\|\mathrm{d}\boldsymbol{v}\|} \\
&\qquad = \lim_{\mathrm{d}\boldsymbol{v}\to\boldsymbol{0}} \frac{H(c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v}), \boldsymbol{v}+\mathrm{d}\boldsymbol{v}, \boldsymbol{x}) - H(c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v}), \boldsymbol{v}, \boldsymbol{x}) + H(c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v}, \boldsymbol{v}, \boldsymbol{x}) - H(c(\boldsymbol{v}), \boldsymbol{v}, \boldsymbol{x})}{\|\mathrm{d}\boldsymbol{v}\|} \\
&\qquad = \lim_{\mathrm{d}\boldsymbol{v}\to\boldsymbol{0}} \int_{-c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v})}^{1} \left(\boldsymbol{x} + \xi\frac{\boldsymbol{v}+\mathrm{d}\boldsymbol{v}}{\|\boldsymbol{v}+\mathrm{d}\boldsymbol{v}\|}a\right)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a + \mathrm{o}(1) \\
&\qquad\qquad + \frac{\int_{-c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v})}^{-c(\boldsymbol{v})} (\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)(1-a^2)^{\frac{d-1}{2}} \frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a}{\|\mathrm{d}\boldsymbol{v}\|}.
\end{aligned}$$

Since $|c(\boldsymbol{v})| = 1$, the integrand in the last expression satisfy

$$\begin{aligned}
|(\boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \xi\|\boldsymbol{v}\|a)(1-a^2)^{\frac{d-1}{2}}| \leq& (|\boldsymbol{v}^\mathsf{T}\boldsymbol{x}| + \xi\|\boldsymbol{v}\|)(c(\boldsymbol{v})^2 - c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v})^2)^{\frac{d-1}{2}} \\
\leq& (|\boldsymbol{v}^\mathsf{T}\boldsymbol{x}| + \xi\|\boldsymbol{v}\|)2^{\frac{d-1}{2}}|c(\boldsymbol{v}) - c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v})|^{\frac{d-1}{2}} \\
=& \mathrm{O}(\|\mathrm{d}\boldsymbol{v}\|^{\frac{d-1}{2}}).
\end{aligned}$$

Thus, using Lebesgue's dominated convergence theorem, we get

$$\begin{aligned}\lim_{\mathrm{d}\boldsymbol{v}\to\mathbf{0}} \frac{\phi(\boldsymbol{v}+\mathrm{d}\boldsymbol{v},\boldsymbol{x})-\phi(\boldsymbol{v},\boldsymbol{x})}{\|\mathrm{d}\boldsymbol{v}\|} &= \lim_{\mathrm{d}\boldsymbol{v}\to\mathbf{0}} \int_{-c(\boldsymbol{v}+\mathrm{d}\boldsymbol{v})}^{1} \left(\boldsymbol{x}+\xi\frac{\boldsymbol{v}+\mathrm{d}\boldsymbol{v}}{\|\boldsymbol{v}+\mathrm{d}\boldsymbol{v}\|}a\right)(1-a^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a + \mathrm{o}(1) + \mathrm{O}(\|\mathrm{d}\boldsymbol{v}\|^{\frac{d-1}{2}}) \\ &= \int_{-c}^{1}\left(\boldsymbol{x}+\xi\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}a\right)(1-a^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a.\end{aligned}$$

Therefore, $\phi(\boldsymbol{v},\boldsymbol{x})$ is continuously differentiable in $\boldsymbol{v}$ when $\boldsymbol{v}\neq\mathbf{0}$.

Similarly, we get the following Hessian

$$\begin{aligned}&\nabla^2_{\boldsymbol{v}}\phi(\boldsymbol{v},\boldsymbol{x}) \\ &\quad= \int_{-c}^{1}\frac{\xi a}{\|\boldsymbol{v}\|}\boldsymbol{P}_{\boldsymbol{v}}(1-a^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}a + \left(\boldsymbol{x}-\xi\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}c\right)(1-c^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}(\nabla_{\boldsymbol{v}}c)^{\mathsf{T}} \\ &\quad= \boldsymbol{P}_{\boldsymbol{v}}\frac{\xi(1-c^2)^{\frac{d+1}{2}}}{(d+1)\|\boldsymbol{v}\|}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)} + \left(\boldsymbol{x}-\xi\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}c\right)(1-c^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}(\nabla_{\boldsymbol{v}}c)^{\mathsf{T}}.\end{aligned}$$

Notice that when $c\in(-1,1)$

$$\nabla_{\boldsymbol{v}}c = \boldsymbol{P}_{\boldsymbol{v}}\frac{\boldsymbol{x}}{\xi},\ \boldsymbol{x}-\xi\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}c = \boldsymbol{P}_{\boldsymbol{v}}\boldsymbol{x}.$$

Thus, the Hessian exists, is continuous, and has the form $\nabla^2_{\boldsymbol{v}}\phi(\boldsymbol{v},\boldsymbol{x}) = \boldsymbol{P}_{\boldsymbol{v}}\boldsymbol{A}(\boldsymbol{v},\boldsymbol{x})\boldsymbol{P}_{\boldsymbol{v}}$.

Convergence of $\boldsymbol{r}_j$ First, notice that $\exists\lim_{t\to\infty} s_jG(\boldsymbol{r}_j) =: G_j^*$ because $s_jG(\boldsymbol{r}_j)$ is monotonically increasing, $\boldsymbol{r}_j\in\mathrm{S}^{d-1}$, and G is continuous. Now, consider function $\zeta_j(t) \coloneqq \int_0^t\left\|\frac{\mathrm{d}\boldsymbol{r}_j}{\mathrm{d}t}\right\|\mathrm{d}\tau$, function $f_j(\boldsymbol{r}_j)\coloneqq G_j^* - s_jG(\hat{\boldsymbol{r}}_j)$, and desingularizing function ψ_j, corresponding to f_j. Then for big enough t we have

$$\frac{\mathrm{d}f_j}{\mathrm{d}\zeta_j} = -\|\nabla_{\boldsymbol{r}}G(\hat{\boldsymbol{r}}_j)\| = -\|\nabla_{\boldsymbol{r}}f_j(\boldsymbol{r}_j)\| \le -\frac{1}{\psi_j'(f_j(\boldsymbol{r}_j))} \implies -\frac{\mathrm{d}\psi_j}{\mathrm{d}t} \ge \frac{\mathrm{d}\zeta_j}{\mathrm{d}t} \implies \zeta_j(t)\le\zeta_j(t_0)+\psi_j(t_0)-\psi_j(t).$$

Thus, ζ_j is bounded, and hence $\lim_{t\to\infty}\zeta_j$ exists. Therefore, the trajectory of $\boldsymbol{r}_j$ has a finite length, and hence $\lim_{t\to\infty}\boldsymbol{r}_j =: \hat{\boldsymbol{v}}_j^*$ exists. Also, since

$$\frac{\mathrm{d}f_j}{\mathrm{d}t} = -\|\nabla_{\boldsymbol{r}}G(\hat{\boldsymbol{r}}_j)\|^2,$$

$\nabla_{\boldsymbol{r}}G(\hat{\boldsymbol{r}})$ is differentiable, and $\hat{\boldsymbol{r}}_j$ converges. Moreover, $\lim_{t\to\infty}\|\nabla_{\boldsymbol{r}}G(\hat{\boldsymbol{r}}_j)\| = 0$ (otherwise f would increase to infinity). Thus, $\hat{\boldsymbol{v}}_j^*$ is critical: $\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^*}\boldsymbol{g}(\hat{\boldsymbol{v}}_j^*) = \mathbf{0}$.

B.1.2. Dynamics around critical directions

We divide all neurons into two categories: prominent and non-prominent. Prominent neurons achieve the optimum of function G: $s_jG(\hat{\boldsymbol{v}}_j^*) = \lambda \coloneqq \max_{\hat{\boldsymbol{v}}\in\mathrm{S}^{d-1}}|G(\hat{\boldsymbol{v}})|$. Denote $\lambda_j \coloneqq s_jG(\hat{\boldsymbol{v}}_j^*)$, $P\coloneqq\{j\mid\lambda_j=\lambda\}$ (prominent neurons), and $R\coloneqq[m]\setminus P$ (non-prominent neurons).

Dynamics of small neurons First, we describe the dynamics of neurons from R. We have

$$u_j^l = u_j(0)\exp\left(\int_0^t s_jG(\hat{\boldsymbol{v}}_j^l)\,\mathrm{d}\tau\right) \le u_j(0)\mathrm{e}^{\lambda_j t}.$$

Notice that the norm growth of these neurons is slower than $\mathrm{e}^{\lambda t}$.

Dynamics of the big neurons' directions Now, we describe the dynamics of neurons from P. Near critical directions, the Hessian of G restricted on the unit sphere is non-positive if $s_j = 1$ and non-negative if $s_j = -1$ due to the local characterization of extremum. Consider the case $s_j = 1$ (the case $s_j = -1$ is similar). Near critical direction, we have

$$(\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\frac{\mathrm{d}\hat{\boldsymbol{v}}_j^l}{\mathrm{d}t} = (\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^l}\boldsymbol{g}(\hat{\boldsymbol{v}}_j^l) = (\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^l}(\boldsymbol{g}(\hat{\boldsymbol{v}}_j^*) + \nabla^2 G(\hat{\boldsymbol{v}}_j^*)(\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*) + \mathrm{O}(\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|^2))$$

Denote $\hat{\boldsymbol{v}}_j^l = \cos(\alpha)\hat{\boldsymbol{v}}_j^* + \sin(\alpha)\boldsymbol{\epsilon}$, where $\cos(\alpha) = (\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\hat{\boldsymbol{v}}_j^l$ and $\boldsymbol{\epsilon} \coloneqq \frac{\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^*}\hat{\boldsymbol{v}}_j^l}{\|\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^*}\hat{\boldsymbol{v}}_j^l\|}$. We get

$$\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^l}\hat{\boldsymbol{v}}_j^* = (1-\cos(\alpha)^2)\hat{\boldsymbol{v}}_j^* - \sin(\alpha)\cos(\alpha)\boldsymbol{\epsilon}.$$

Since G is homogeneous, we get

$$\boldsymbol{g}(\hat{\boldsymbol{v}}_j^*) = \lambda_j\hat{\boldsymbol{v}}_j^*.$$

Also, we know that

$$\nabla^2 G(\hat{\boldsymbol{v}}) = \boldsymbol{P}_{\hat{\boldsymbol{v}}}\nabla^2 G(\hat{\boldsymbol{v}})\boldsymbol{P}_{\hat{\boldsymbol{v}}} \implies \nabla^2 G(\hat{\boldsymbol{v}}_j^*)(\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*) = \nabla^2 G(\hat{\boldsymbol{v}}_j^*)\sin(\alpha)\boldsymbol{\epsilon}.$$

These equations give

$$\begin{aligned}(\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\frac{\mathrm{d}\hat{\boldsymbol{v}}_j^l}{\mathrm{d}t} =&((1-\cos(\alpha)^2)\hat{\boldsymbol{v}}_j^* - \sin(\alpha)\cos(\alpha)\boldsymbol{\epsilon})^\mathsf{T}(\lambda\hat{\boldsymbol{v}}_j^* + \nabla^2 G(\hat{\boldsymbol{v}}_j^*)\sin(\alpha)\boldsymbol{\epsilon} + \mathrm{O}(\alpha^2))\\ =&\lambda(1-\cos(\alpha)^2) - \boldsymbol{\epsilon}^\mathsf{T}\nabla^2 G(\hat{\boldsymbol{v}}_j^*)\boldsymbol{\epsilon}\sin(\alpha)^2\cos(\alpha) + \mathrm{O}(\alpha^3)\\ \geq&\lambda\|\hat{\boldsymbol{v}}_j^* - \hat{\boldsymbol{v}}_j^l\|^2 + \mathrm{O}(\alpha^3).\end{aligned}$$

This equation implies

$$\frac{\mathrm{d}\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|^2}{\mathrm{d}t} \leq -2\lambda\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|^2 + 2\delta\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|^3 \implies \frac{\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|}{1-\frac{\delta}{\lambda}\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|} \leq \frac{\|\hat{\boldsymbol{v}}_j^l(t_{0,j}) - \hat{\boldsymbol{v}}_j^*(t_{0,j})\|\mathrm{e}^{-\lambda(t-t_{0,j})}}{1-\frac{\delta}{\lambda}\|\hat{\boldsymbol{v}}_j^l(t_{0,j}) - \hat{\boldsymbol{v}}_j^*(t_{0,j})\|}$$

$$\implies \|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\| \leq \frac{\|\hat{\boldsymbol{v}}_j^l(t_{0,j}) - \hat{\boldsymbol{v}}_j^*(t_{0,j})\|\mathrm{e}^{-\lambda(t-t_{0,j})}}{1-\frac{\delta}{\lambda}\|\hat{\boldsymbol{v}}_j^l(t_{0,j}) - \hat{\boldsymbol{v}}_j^*(t_{0,j})\|(1-\mathrm{e}^{-\lambda(t-t_{0,j})})},$$

where $2\delta\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|^3$ comes from the term $\mathrm{O}(\alpha^3)$ and $t_{0,j}$ is chosen such that $\|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\|$ is sufficiently small to bound our big-O term and, at the same time, $\frac{\delta}{\lambda}\|\hat{\boldsymbol{v}}_j^l(t_0) - \hat{\boldsymbol{v}}_j^*(t_0)\| \leq \frac{1}{2}$. Thus, we get an exponentially fast convergence near the critical direction.

Denote

$$t_0 \coloneqq \max_j t_{0,j},\ c_0 \coloneqq \max_j 2\|\hat{\boldsymbol{v}}_j^l(t_{0,j}) - \hat{\boldsymbol{v}}_j^*(t_{0,j})\|\mathrm{e}^{\lambda t_{0,j}}.$$

We get

$$\forall j \in P, t \geq t_0\ \|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j\| \leq c_0\mathrm{e}^{-\lambda t}.$$

Dynamics of the big neurons' scales Now, we will describe the dynamics of u_j^l. Notice

$$\frac{\mathrm{d}u_j^l}{\mathrm{d}t} = u_j^l G(\hat{\boldsymbol{v}}_j^l) \implies u_j^l = u_j^l(0)\exp\left(\int_0^t G(\hat{\boldsymbol{v}}_j^l)\,\mathrm{d}\tau\right).$$

This equality motivates us to consider a limit

$$u_j^* \coloneqq \lim_{t\to\infty} u_j^l\mathrm{e}^{-\lambda t}.$$

This limit exists since the right hand part is monotonically decreasing. We get

$$u_j^l = u_j^l(0)\exp\left(\int_0^t G(\hat{\boldsymbol{v}}_j^l)\,\mathrm{d}\tau\right) = u_j^*\exp\left(\lambda t + \int_t^\infty(\lambda - G(\hat{\boldsymbol{v}}_j^l))\,\mathrm{d}\tau\right).$$

Thus,

$$u_j^*\mathrm{e}^{\lambda t} \leq u_j^l \leq u_j^*\exp\left(\lambda t + \int_t^\infty c_0 b_0\mathrm{e}^{-\lambda\tau}\,\mathrm{d}\tau\right) = u_j^*\exp\left(\lambda t + \frac{c_0 b_0}{\lambda}\mathrm{e}^{-\lambda t}\right),$$

where $b_0 \coloneqq \sup_{\boldsymbol{v}\neq\boldsymbol{v}'}\frac{|G(\boldsymbol{v})-G(\boldsymbol{v}')|}{\|\boldsymbol{v}-\boldsymbol{v}'\|}$. Notice that this derivation also implies that neurons in P grow as $\Theta(\mathrm{e}^{\lambda t})$.

Remark B.4. It is easy to see that $b_0 \leq (-\ell'(0))(1+\xi)$.

Neuron capture Notice that all global extrema are attractive. Therefore, for isotropic initialization, with probability at least $1-(f_j)^m$, some point will be attracted to the global extremum $\hat{\boldsymbol{v}}_j^*$, where $f_j \coloneqq 1 - \frac{\text{volume of attraction region}}{\text{volume of shpere}}$.

B.2. Coupling Equations (1) and (2)

B.2.1. Norm Growth Rate

First, we prove the following proposition and lemma.

Proposition B.5. *Assuming $\|\boldsymbol{x}_i\| \le 1$, the following identities hold*

$$\left|\frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta},\boldsymbol{x}_i)y_i)+\ell'(0))\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)y_i\right| \le a\|\boldsymbol{\theta}\|_{[m]}^2,$$
$$\left|\frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta},\boldsymbol{x}_i)y_i)+\ell'(0))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{z},\boldsymbol{x}_i)|_{\boldsymbol{z}=\hat{\boldsymbol{v}}}y_i\right| \le a\|\boldsymbol{\theta}\|_{[m]}^2,$$

where $a \coloneqq \frac{m(1+\xi)^2}{4}$.

Proof. Notice that

$$|f(\boldsymbol{\theta},\boldsymbol{x}_i)| \le \sum_{j=1}^{m} u_j^2|\phi(\hat{\boldsymbol{v}}_j,\boldsymbol{x}_i)| \le (1+\xi)\sum_{j=1}^{m}u_j^2 \le m(1+\xi)\|\boldsymbol{\theta}\|_{[m]}^2,$$

and

$$|-\ell'(f(\boldsymbol{\theta},\boldsymbol{x}_i)y_i)+\ell'(0)| \le |f(\boldsymbol{\theta},\boldsymbol{x}_i)|\sup_z|\ell''(z)| \le \frac{m(1+\xi)}{4}\|\boldsymbol{\theta}\|_{[m]}^2.$$

Thus,

$$\left|\frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta},\boldsymbol{x}_i)y_i)+\ell'(0))\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)y_i\right| \le \frac{m(1+\xi)}{4}\|\boldsymbol{\theta}\|_{[m]}^2\max_{i,\hat{\boldsymbol{v}}}|\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)| \le a\|\boldsymbol{\theta}\|_{[m]}^2.$$

Similarly for the gradients of activation. □

Lemma B.6. *Assume that $\boldsymbol{\theta}$ follows Equation (1) and $|u_j(0)| = \|\boldsymbol{v}_j(0)\|$. Then*

$$\forall t \le t_1\ \|\boldsymbol{\theta}\|_{[m]}^2 \le 2\|\boldsymbol{\theta}(0)\|_{[m]}^2 \mathrm{e}^{2\lambda t},$$

where $t_1 \coloneqq \frac{1}{2\lambda}\ln\left(\frac{\lambda}{2a\|\boldsymbol{\theta}(0)\|_{[m]}^2}\right)$.

Proof. Thus,

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = G(\boldsymbol{v}_j) + \frac{1}{n}\sum_{i=1}^{n}(p_i(\boldsymbol{\theta})+\ell'(0))\phi(\boldsymbol{v}_j,\boldsymbol{x}_i)y_i \implies \left|\frac{\mathrm{d}u_j}{\mathrm{d}t}\right| \le |u_j||G(\hat{\boldsymbol{v}}_j)| + a\|\boldsymbol{\theta}\|_{[m]}^2|u_j| \le \lambda\|\boldsymbol{\theta}\|_{[m]} + a\|\boldsymbol{\theta}\|_{[m]}^3$$
$$\implies \int \frac{\mathrm{d}\|\boldsymbol{\theta}\|_{[m]}^2}{\|\boldsymbol{\theta}\|_{[m]}^2 + \frac{a}{\lambda}\|\boldsymbol{\theta}\|_{[m]}^4} \le 2\lambda t \implies \|\boldsymbol{\theta}\|_{[m]}^2 \le \frac{\|\boldsymbol{\theta}(0)\|_{[m]}^2\mathrm{e}^{2\lambda t}}{1-\frac{a}{\lambda}\|\boldsymbol{\theta}(0)\|_{[m]}^2(\mathrm{e}^{2\lambda t}-1)}.$$

Therefore,

$$\|\boldsymbol{\theta}\|_{[m]}^2 \le 2\|\boldsymbol{\theta}(0)\|_{[m]}^2\mathrm{e}^{2\lambda t}\ \forall t \le \frac{1}{2\lambda}\ln\left(\frac{\lambda}{2a\|\boldsymbol{\theta}(0)\|_{[m]}^2}\right).$$

□

B.2.2. COUPLING DIRECTIONS

We have

$$\frac{\mathrm{d}\hat{\boldsymbol{v}}_j}{\mathrm{d}t} = s_j \boldsymbol{P}_{\boldsymbol{v}} \left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j) + \frac{1}{n} \sum_{i=1}^{n} (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i) + \ell'(0)) \nabla_{\boldsymbol{v}} \phi(\boldsymbol{v}_j, \boldsymbol{x}_i) y_i \right),$$
$$\frac{\mathrm{d}\hat{\boldsymbol{v}}_j^l}{\mathrm{d}t} = s_j \boldsymbol{P}_{\boldsymbol{v}^l} \boldsymbol{g}(\hat{\boldsymbol{v}}_j^l).$$

Denote $b_1 \coloneqq \sup_{\hat{\boldsymbol{v}}, \hat{\boldsymbol{v}}' \in \mathbb{S}^{d-1}, \hat{\boldsymbol{v}} \neq \hat{\boldsymbol{v}}'} \frac{\|\boldsymbol{P}_{\hat{\boldsymbol{v}}} \boldsymbol{g}(\hat{\boldsymbol{v}}) - \boldsymbol{P}_{\hat{\boldsymbol{v}}'} \boldsymbol{g}(\hat{\boldsymbol{v}}')\|}{\|\hat{\boldsymbol{v}} - \hat{\boldsymbol{v}}'\|}$ (since ϕ is twice continuously differentiable, this constant is defined).

Remark B.7. It is easy to see that $b_1 \leq 2 \sup_{\hat{\boldsymbol{v}}} \|\boldsymbol{g}(\hat{\boldsymbol{v}})\| + (-\ell'(0)) \sup_{i, \hat{\boldsymbol{v}}} \|\nabla_{\boldsymbol{v}}^2 \phi(\boldsymbol{v}, \boldsymbol{x}_i)\| \implies b_1 = \mathrm{O}(1/\xi)$, when $\xi \to 0$.

We get

$$\begin{aligned}
\frac{\mathrm{d}\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^l\|}{\mathrm{d}t} \leq & \left\| \frac{\mathrm{d}\hat{\boldsymbol{v}}_j}{\mathrm{d}t} - \frac{\mathrm{d}\hat{\boldsymbol{v}}_j^l}{\mathrm{d}t} \right\| \leq \left\| \frac{1}{n} \sum_{i=1}^{n} (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i) + \ell'(0)) \nabla_{\boldsymbol{v}} \phi(\boldsymbol{v}_j, \boldsymbol{x}_i) y_i \right\| + \|\boldsymbol{P}_{\hat{\boldsymbol{v}}} \boldsymbol{g}(\hat{\boldsymbol{v}}) - \boldsymbol{P}_{\hat{\boldsymbol{v}}^l} \boldsymbol{g}(\hat{\boldsymbol{v}}^l)\| \\
\leq & a \|\boldsymbol{\theta}\|_{[m]}^2 + b_1 \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^l\|.
\end{aligned}$$

Consider function $h \coloneqq \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^l\| \mathrm{e}^{-b_1 t}$, we get

$$\begin{aligned}
& \forall t \leq t_1 \ \frac{\mathrm{d}h}{\mathrm{d}t} \leq a \|\boldsymbol{\theta}\|_{[m]}^2 \mathrm{e}^{-b_1 t} \implies \forall t \leq t_1 \ h \leq \int_0^t 2a \|\boldsymbol{\theta}(0)\|_{[m]}^2 \mathrm{e}^{(2\lambda - b_1)\tau} \, \mathrm{d}\tau \\
\implies & \forall t \leq t_1 \ h \leq \frac{2a}{b_1 - 2\lambda} \|\boldsymbol{\theta}(0)\|_{[m]}^2 (1 - \mathrm{e}^{(2\lambda - b_1)t}) \implies \forall t \leq t_1 \ \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^l\| \leq \frac{2a}{b_1 - 2\lambda} \|\boldsymbol{\theta}(0)\|_{[m]}^2 \mathrm{e}^{b_1 t},
\end{aligned}$$

where we have assumed that $b_1 > 2\lambda$ (which holds for small enough ξ).

B.2.3. COUPLING SCALES

We have

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = G(\boldsymbol{v}_j) + \frac{1}{n} \sum_{i=1}^{n} (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i) + \ell'(0)) \phi(\boldsymbol{v}_j, \boldsymbol{x}_i) y_i,$$
$$\frac{\mathrm{d}u_j^l}{\mathrm{d}t} = G(\boldsymbol{v}_j^l).$$

It gives

$$\begin{aligned}
\forall t \leq t_1 \ \frac{\mathrm{d}|u_j - u_j^l|}{\mathrm{d}t} \leq & \left| \frac{\mathrm{d}u_j}{\mathrm{d}t} - \frac{\mathrm{d}u_j^l}{\mathrm{d}t} \right| \leq a \|\boldsymbol{\theta}\|_{[m]}^2 |u_j| + |u_j^l| |G(\hat{\boldsymbol{v}}_j) - G(\hat{\boldsymbol{v}}_j^l)| + |u_j - u_j^l| |G(\hat{\boldsymbol{v}}_j)| \\
\leq & 2\sqrt{2} a \|\boldsymbol{\theta}(0)\|_{[m]}^3 \mathrm{e}^{3\lambda t} + \frac{2ab_0 \|\boldsymbol{\theta}(0)\|_{[m]}^3}{b_1 - 2\lambda} \mathrm{e}^{(\lambda_j + b_1)t} + \lambda |u_j - u_j^l|.
\end{aligned}$$

As previously, consider function $h \coloneqq |u_j - u_j^l| \mathrm{e}^{-\lambda t}$, we get

$$\begin{aligned}
& \forall t \leq t_1 \ h \leq 2\sqrt{2} a \|\boldsymbol{\theta}(0)\|_{[m]}^3 \mathrm{e}^{2\lambda t} + \frac{2ab_0 \|\boldsymbol{\theta}(0)\|_{[m]}^3}{b_1 - 2\lambda} \mathrm{e}^{(\lambda_j + b_1 - \lambda)t} \\
\implies & \forall t \leq t_1 \ |u_j - u_j^l| \leq \frac{a\sqrt{2}}{\lambda} \|\boldsymbol{\theta}(0)\|_{[m]}^3 (\mathrm{e}^{3\lambda t} - \mathrm{e}^{\lambda t}) + \frac{2ab_0}{(b_1 - 2\lambda)(\lambda_j + b_1 - \lambda)} \|\boldsymbol{\theta}(0)\|_{[m]}^3 (\mathrm{e}^{(\lambda_j + b_1)t} - \mathrm{e}^{\lambda t}) \\
\implies & \forall t \leq t_1 \ |u_j - u_j^l| \leq c_1 \|\boldsymbol{\theta}(0)\|_{[m]}^3 \mathrm{e}^{(\lambda_j + b_1)t},
\end{aligned}$$

where we have assumed that $b_1 \geq 4\lambda$ and denoted $c_1 \coloneqq \frac{a\sqrt{2}}{\lambda} + \frac{2ab_0}{(b_1 - 2\lambda)^2}$.

B.3. Final Bound

The result of the previous sections show

$$\begin{aligned}
&\forall j \in P, t \in [t_0, t_1] \qquad \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^*\| \le \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^l\| + \|\hat{\boldsymbol{v}}_j^l - \hat{\boldsymbol{v}}_j^*\| \le \frac{2a}{b_1 - 2\lambda}\|\boldsymbol{\theta}(0)\|_{[m]}^2 \mathrm{e}^{c^{\boldsymbol{g}} t} + c_0 \mathrm{e}^{-\lambda t},\\
&\forall j \in P, t \in [t_0, t_1] \quad |u_j - u_j^* \mathrm{e}^{\lambda t}| \le |u_j - u_j^l| + |u_j^l - u_j^* \mathrm{e}^{\lambda t}|\\
&\qquad\qquad\qquad\qquad \le c_1 \|\boldsymbol{\theta}(0)\|_{[m]}^3 \mathrm{e}^{(\lambda + b_1)t} + |u_j^*| \mathrm{e}^{\lambda t}\left(\exp\left(\frac{c_0 b_0}{\lambda}\mathrm{e}^{-\lambda t}\right) - 1\right),\\
&\forall j \in R, t \in [0, t_1] \qquad\qquad |u_j| \le c_1 \|\boldsymbol{\theta}(0)\|_{[m]}^3 \mathrm{e}^{(\lambda_j + c^{\boldsymbol{g}})t} + |u_j(0)| \mathrm{e}^{\lambda_j t}.
\end{aligned}$$

We want to control these errors until the point when the fastest growing directions will have a predefined scale, r. To do this, we will choose $T_1 = \frac{1}{\lambda}\ln\left(\frac{r}{\sigma}\right)$ and $\sigma = r^{\kappa+1}$. (Notice that t_0 does not depend on r, and $t_1 = -\ln(\sigma)/\lambda + \mathrm{O}(1)$. Thus, for sufficiently small r, we have $t_0 < T_1 < t_1$.)

This functional form will give us the following errors:

$$\begin{aligned}
&\forall j \in P \qquad \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^*\| \le \mathrm{O}(r^{2-\kappa(b_1/\lambda - 2)} + r^{\kappa}),\\
&\forall j \in P \quad |u_j - u_j^* \mathrm{e}^{\lambda T_1}| \le \mathrm{O}(r^{3-\kappa(b_1/\lambda - 2)} + r^{1+\kappa}),\\
&\forall j \in R \qquad\qquad |u_j| \le \mathrm{O}\left(r^{3-\kappa\left(\frac{b_1+\lambda_j}{\lambda} - 3\right)} + r^{1+\kappa\left(1-\frac{\lambda_j}{\lambda}\right)}\right).
\end{aligned}$$

We choose $\kappa^* = \frac{2}{\frac{b_1}{\lambda} - 1}$. It will give

$$\begin{aligned}
&\forall j \in P \qquad \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^*\| \le \mathrm{O}(r^{\kappa^*}),\\
&\forall j \in P \quad |u_j - u_j^* \mathrm{e}^{\lambda T_1}| \le \mathrm{O}(r^{1+\kappa^*}),\\
&\forall j \in R \qquad\qquad |u_j| \le \mathrm{O}(r^{1+\kappa_j}),
\end{aligned}$$

where $\kappa_j \coloneqq \kappa^*(1 - \lambda_j/\lambda)$.

C. Proof of Theorem 4.4

By Lemma 5.3 of Lyu et al. (2021), we could write the following dynamics for embedding (4)

$$\begin{aligned}
\frac{\mathrm{d}u_j^\chi}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i) y_i)) \phi(\boldsymbol{v}_j^\chi, \boldsymbol{x}_i) y_i,\\
\frac{\mathrm{d}\boldsymbol{v}_j^\chi}{\mathrm{d}t} &= \frac{u_j^\chi}{n}\sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i) y_i)) \nabla_{\boldsymbol{v}} \phi(\boldsymbol{v}_j^\chi, \boldsymbol{x}_i) y_i.
\end{aligned} \tag{7}$$

Denote $\boldsymbol{\theta}^* \coloneqq \boldsymbol{\theta}^\chi(T_1)/r$.

We will prove the theorem in two stages. First, we will couple Equations (1) and (7). Then we will investigate the sensitivity of Equation (7) to initialization scale. This analysis will allow us to analyze the solution to Equation (1).

C.1. Coupling Equations (1) and (7)

C.1.1. Norm Growth

Since $\|\boldsymbol{\theta}^\chi(T_1)\|_{[m]} = \Theta(r)$ and $\|\boldsymbol{\theta}^\chi(T_1) - \boldsymbol{\theta}(T_1)\|_{[m]} = \mathrm{o}(r)$, we have that $\|\boldsymbol{\theta}(T_1)\| \le \sqrt{2}\|\boldsymbol{\theta}^\chi(T_1)\|$ for sufficiently small r. Using this fact and Lemma B.6, we get

$$\begin{aligned}
&\forall t \in [T_1, T_1 + t_3] \qquad \|\boldsymbol{\theta}(T_1)\|_{[m]}^2 \le 2q^2 \mathrm{e}^{2\lambda(t - T_1)},\\
&\forall t \in [T_1, T_1 + t_3] \quad \|\boldsymbol{\theta}^\chi(T_1)\|_{[m]}^2 \le 2q^2 \mathrm{e}^{2\lambda(t - T_1)},
\end{aligned}$$

where $t_3 \coloneqq \frac{1}{2\lambda}\ln\left(\frac{\lambda}{2aq^2}\right)$ and $q^2 \coloneqq 2\|\boldsymbol{\theta}^\chi(T_1)\|_{[m]}^2 = 2r^2\|\boldsymbol{\theta}^*\|_{[m]}^2$.

C.1.2. BOUNDING $\|\hat{\boldsymbol{v}}_j^\chi - \hat{\boldsymbol{v}}_j^*\|$

W.l.o.g. assume that $s_j = 1$. Additionally, assume $\|\hat{\boldsymbol{v}}_j^\chi - \hat{\boldsymbol{v}}_j^*\| \leq \varepsilon$ and denote $\hat{\boldsymbol{v}}_j^\chi = \cos(\alpha')\hat{\boldsymbol{v}}_j^* + \sin(\alpha')\boldsymbol{\epsilon}$. First, notice that $\|\hat{\boldsymbol{v}}_j^\chi - \hat{\boldsymbol{v}}_j^*\| \leq \varepsilon \leq \Delta$ implies

$$\phi(\hat{\boldsymbol{v}}_j^\chi, \boldsymbol{x}_i) = ((\hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\boldsymbol{x}_i)_+ \implies \boldsymbol{g}(\hat{\boldsymbol{v}}_j^\chi) = \boldsymbol{g}(\hat{\boldsymbol{v}}_j^*).$$

We have

$$\begin{aligned}
\forall j \in P\ (\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\frac{\mathrm{d}\hat{\boldsymbol{v}}_j^\chi}{\mathrm{d}t} =& (\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j^*)^\mathsf{T}\left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j^\chi) + \frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)y_i) + \ell'(0))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j^\chi, \boldsymbol{x}_i)y_i\right)\\
\geq& ((1-\cos(\alpha')^2)\hat{\boldsymbol{v}}_j^* - \sin(\alpha')\cos(\alpha')\boldsymbol{\epsilon})^\mathsf{T}\boldsymbol{g}(\hat{\boldsymbol{v}}_j^\chi) - \sin(\alpha')a\|\boldsymbol{\theta}^\chi\|_{[m]}^2\\
=& \lambda\sin(\alpha')^2 - \sin(\alpha')a\|\boldsymbol{\theta}^\chi\|_{[m]}^2
\end{aligned}$$

where $\boldsymbol{z} \in [\hat{\boldsymbol{v}}_j^*, \hat{\boldsymbol{v}}_j^\chi]$. It implies

$$\forall t \in [T_1, T_1 + t_3]\ \frac{\mathrm{d}\alpha'}{\mathrm{d}t} \leq 2aq^2\mathrm{e}^{2\lambda(t-T_1)} - \lambda\sin(\alpha') \implies \alpha' \leq \frac{a}{\lambda}q^2(\mathrm{e}^{2\lambda(t-T_1)} - 1).$$

So, to ensure $\|\hat{\boldsymbol{v}}_j^\chi - \hat{\boldsymbol{v}}_j^*\| \leq \varepsilon$, it is sufficient to have

$$\frac{a}{\lambda}q^2\mathrm{e}^{2\lambda(t-T_1)} \leq \varepsilon \impliedby t - T_1 \leq \frac{1}{2\lambda}\ln\left(\frac{\lambda\varepsilon}{aq^2}\right) =: t_4^\varepsilon.$$

C.1.3. BOUNDING $\|\boldsymbol{\theta} - \boldsymbol{\theta}^\chi\|$

Coupling Directions First, we want to bound $\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|$. Assuming that $\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\| \leq \varepsilon$, we get $\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^*\| \leq 2\varepsilon$ and

$$\begin{aligned}
\frac{\mathrm{d}\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|^2}{\mathrm{d}t} =& -\hat{\boldsymbol{v}}_j^\mathsf{T}\frac{\mathrm{d}\hat{\boldsymbol{v}}_j^\chi}{\mathrm{d}t} - (\hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\frac{\mathrm{d}\hat{\boldsymbol{v}}_j}{\mathrm{d}t}\\
=& -(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j)^\mathsf{T}\left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j^\chi) + \frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)y_i) + \ell'(0))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j^\chi, \boldsymbol{x}_i)y_i\right)\\
&-(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j}\hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j) + \frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i) + \ell'(0))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i\right)\\
=& -(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j)^\mathsf{T}(\boldsymbol{g}(\hat{\boldsymbol{v}}_j^\chi) - \boldsymbol{g}(\hat{\boldsymbol{v}}_j))\\
&-(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j)^\mathsf{T}\left(\frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)y_i) + \ell'(0))(\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j^\chi, \boldsymbol{x}_i) - \nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i))y_i\right)\\
&-(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j)^\mathsf{T}\left(\frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)y_i) + \ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i\right)\\
&-(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j + \boldsymbol{P}_{\hat{\boldsymbol{v}}_j}\hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j) + \frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i) + \ell'(0))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i\right)\\
=& -(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j)^\mathsf{T}\left(\frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)y_i) + \ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i\right)\\
&-(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi}\hat{\boldsymbol{v}}_j + \boldsymbol{P}_{\hat{\boldsymbol{v}}_j}\hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j) + \frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i) + \ell'(0))\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i\right).
\end{aligned}$$

Denote $\cos(\beta) \coloneqq \hat{\boldsymbol{v}}_j^\mathsf{T}\hat{\boldsymbol{v}}_j^\chi$. We get

$$\|\boldsymbol{P}_{\hat{\boldsymbol{v}}_j}\hat{\boldsymbol{v}}_j^\chi\|^2 = \|\hat{\boldsymbol{v}}_j^\chi - \cos(\beta)\hat{\boldsymbol{v}}_j\|^2 = 1 - \cos(\beta)^2 \leq \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|^2.$$

Now, notice

$$\begin{aligned}
|f(\boldsymbol{\theta}, \boldsymbol{x}_i) - f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)| =& \left| \sum_{j=1}^m (u_j^2 \phi(\hat{\boldsymbol{v}}_j, \boldsymbol{x}_i) - (u_j^\chi)^2 \phi(\hat{\boldsymbol{v}}_j^\chi, \boldsymbol{x}_i)) \right| \\
=& \left| \sum_{j=1}^m ((u_j^2 - (u_j^\chi)^2)\phi(\hat{\boldsymbol{v}}_j, \boldsymbol{x}_i) + (u_j^\chi)^2 (\phi(\hat{\boldsymbol{v}}_j, \boldsymbol{x}_i) - \phi(\hat{\boldsymbol{v}}_j^\chi, \boldsymbol{x}_i))) \right| \\
=& \left| \sum_{j=1}^m ((u_j^2 - (u_j^\chi)^2)\phi(\hat{\boldsymbol{v}}_j, \boldsymbol{x}_i) + (u_j^\chi)^2 [(\hat{\boldsymbol{v}}_j^*)^\mathsf{T} \boldsymbol{x}_i \geq 0](\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi)^\mathsf{T} \boldsymbol{x}_i) \right| \\
\leq& |P|(1+\xi)(\|\boldsymbol{\theta}\|_P + \|\boldsymbol{\theta}^\chi\|_P)U + |R|(1+\xi)\|\boldsymbol{\theta}\|_R^2 + |P|\|\boldsymbol{\theta}\|_{[m]}^2 V,
\end{aligned}$$

where $U \coloneqq \max_{j \in P} |u_j - u_j^\chi|$, $V \coloneqq \max_{j \in P} \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|$. This formula implies

$$\begin{aligned}
&\left\| (\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi} \hat{\boldsymbol{v}}_j)^\mathsf{T} \left(\frac{1}{n} \sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i) y_i) + \ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i)) \nabla_{\boldsymbol{v}} \phi(\boldsymbol{v}_j, \boldsymbol{x}_i) y_i \right) \right\| \\
&\qquad \leq aU(\|\boldsymbol{\theta}\|_{[m]} + \|\boldsymbol{\theta}^\chi\|_{[m]}) \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\| + a\|\boldsymbol{\theta}\|_R^2 \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\| + a\|\boldsymbol{\theta}^\chi\|_{[m]}^2 V \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|.
\end{aligned}$$

Finally,

$$\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi} \hat{\boldsymbol{v}}_j + \boldsymbol{P}_{\hat{\boldsymbol{v}}_j} \hat{\boldsymbol{v}}_j^\chi = (1 - \cos(\beta))(\hat{\boldsymbol{v}}_j + \hat{\boldsymbol{v}}_j^\chi) = \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|^2 \frac{\hat{\boldsymbol{v}}_j + \hat{\boldsymbol{v}}_j^\chi}{2},$$

which gives

$$\begin{aligned}
&(\boldsymbol{P}_{\hat{\boldsymbol{v}}_j^\chi} \hat{\boldsymbol{v}}_j + \boldsymbol{P}_{\hat{\boldsymbol{v}}_j} \hat{\boldsymbol{v}}_j^\chi)^\mathsf{T} \left(\boldsymbol{g}(\hat{\boldsymbol{v}}_j) + \frac{1}{n} \sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i) + \ell'(0)) \nabla_{\boldsymbol{v}} \phi(\boldsymbol{v}_j, \boldsymbol{x}_i) y_i \right) \\
&\qquad \geq \lambda \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|^2 \frac{\cos(\alpha) + \cos(\alpha')}{2} - a\|\boldsymbol{\theta}\|_{[m]}^2 \|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|^2,
\end{aligned}$$

where $\cos(\alpha) = \hat{\boldsymbol{v}}_j^\mathsf{T} \hat{\boldsymbol{v}}_j^*$. Together, we get

$$\begin{aligned}
\frac{\mathrm{d}V^2}{\mathrm{d}t} \leq& a(\|\boldsymbol{\theta}\|_{[m]}^2 + \|\boldsymbol{\theta}^\chi\|_{[m]}^2) V^2 + a\|\boldsymbol{\theta}\|_R^2 V + a(\|\boldsymbol{\theta}\|_{[m]} + \|\boldsymbol{\theta}^\chi\|_{[m]}) UV \\
\leq& 4aq^2 \mathrm{e}^{2\lambda(t-T_1)} V^2 + a\|\boldsymbol{\theta}\|_R^2 V + 2\sqrt{2} aq \mathrm{e}^{\lambda(t-T_1)} UV \\
\implies \frac{\mathrm{d}V}{\mathrm{d}t} \leq& 2aq^2 \mathrm{e}^{2\lambda(t-T_1)} V + \frac{a}{2}\|\boldsymbol{\theta}\|_R^2 + \sqrt{2} aq \mathrm{e}^{\lambda(t-T_1)} U.
\end{aligned}$$

Bounding $\|\boldsymbol{\theta}\|_R$ We get

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = G(\boldsymbol{v}_j) + \frac{1}{n} \sum_{i=1}^n (-\ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i) y_i) + \ell'(0)) \phi(\boldsymbol{v}_j, \boldsymbol{x}_i) y_i \implies \left| \frac{\mathrm{d}u_j}{\mathrm{d}t} \right| \leq |u_j|(\lambda + a\|\boldsymbol{\theta}\|_{[m]}^2).$$

Thus,

$$|u_j| \leq |u_j(T_1)| \exp\left(\int_{T_1}^t \lambda + a\|\boldsymbol{\theta}\|_{[m]}^2 \, \mathrm{d}\tau \right) \leq |u_j(T_1)| \exp\left(\lambda(t - T_1) + \frac{a}{\lambda} q^2 \mathrm{e}^{2\lambda(t-T_1)} \right).$$

If $t - T_1 \leq t_4^\varepsilon$, we get

$$|u_j| \leq \mathrm{e}^{\lambda(t-T_1)+\varepsilon} |u_j(T_1)| \implies \|\boldsymbol{\theta}\|_R \leq \mathrm{e}^{\lambda(t-T_1)+\varepsilon} \|\boldsymbol{\theta}(T_1)\|_R.$$

Bounding U Similarly to the previous cases, we have

$$\frac{\mathrm{d}|u_j - u_j^\chi|}{\mathrm{d}t} \le \left|\frac{\mathrm{d}u_j}{\mathrm{d}t} - \frac{\mathrm{d}u_j^\chi}{\mathrm{d}t}\right|$$
$$\le \left|u_j G(\hat{\boldsymbol{v}}_j) - u_j^\chi G(\hat{\boldsymbol{v}}_j^\chi)\right| + \left|\frac{1}{n}\sum_{i=1}^n ((\ell'(0) - \ell'(f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_j, \boldsymbol{x}_i) - (\ell'(0) - \ell'(f(\boldsymbol{\theta}^\chi, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_j^\chi, \boldsymbol{x}_i))y_i\right| \le$$
$$\lambda U + \|\boldsymbol{\theta}^\chi\|_{[m]}|(\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\boldsymbol{g}(\hat{\boldsymbol{v}}_j)| + a\|\boldsymbol{\theta}^\chi\|_{[m]}(U(\|\boldsymbol{\theta}\|_{[m]} + \|\boldsymbol{\theta}^\chi\|_{[m]}) + \|\boldsymbol{\theta}\|_R^2 + \|\boldsymbol{\theta}^\chi\|_{[m]}^2 V) + a\|\boldsymbol{\theta}\|_{[m]}^2\|\boldsymbol{v}_j - \boldsymbol{v}_j^\chi\|.$$

Notice

$$|(\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi)^\mathsf{T}\boldsymbol{g}(\hat{\boldsymbol{v}}_j)| = \lambda|\cos(\alpha) - \cos(\alpha')| = 2\lambda\sin\left(\frac{\alpha+\alpha'}{2}\right)\sin\left(\frac{|\alpha-\alpha'|}{2}\right) \le 2\lambda\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\beta}{2} + \alpha'\right)$$
$$\le \lambda\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|\left(\frac{\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|}{2} + \alpha'\right).$$

Also, notice

$$\|\boldsymbol{v}_j - \boldsymbol{v}_j^\chi\| = \|u_j\hat{\boldsymbol{v}}_j - u_j^\chi\hat{\boldsymbol{v}}_j^\chi\| \le |u_j - u_j^\chi| + |u_j^\chi|\|\hat{\boldsymbol{v}}_j - \hat{\boldsymbol{v}}_j^\chi\|.$$

It will give

$$\frac{\mathrm{d}U}{\mathrm{d}t} \le \lambda U + 6aq^2\mathrm{e}^{2\lambda(t-T_1)}U + \lambda\sqrt{2}q\mathrm{e}^{\lambda(t-T_1)}\frac{V^2}{2} + \sqrt{2}aq\|\boldsymbol{\theta}(T_1)\|_R^2\mathrm{e}^{3\lambda(t-T_1)+\varepsilon} + 5\sqrt{2}aq^3\mathrm{e}^{3\lambda(t-T_1)}V$$

Bounding U and V Consider $h(t) \coloneqq \exp\left(-\int_{T_1}^t 2aq^2\mathrm{e}^{2\lambda(\tau-T_1)}\,\mathrm{d}\tau\right) \le 1$, $\tilde{U} \coloneqq \frac{U}{q}\mathrm{e}^{-\lambda(t-T_1)}h(t)^3$ and $\tilde{V} \coloneqq Vh(t)$. Notice that

$$h(t) = \exp\left(-\frac{aq^2}{\lambda}\mathrm{e}^{2\lambda(t-T_1)}\right) \ge \mathrm{e}^{-\varepsilon}\ \forall t \in [T_1, T_2^\varepsilon],$$

where $T_2^\varepsilon \coloneqq T_1 + t_4^\varepsilon$. For $t \in [T_1, T_2^\varepsilon]$, we get

$$\frac{\mathrm{d}\tilde{U}}{\mathrm{d}t} \le \frac{\sqrt{2}\lambda}{2}\tilde{V}^2 + \sqrt{2}a\mathrm{e}^{2\lambda(t-T_1)+\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2 + 5\sqrt{2}aq^2\mathrm{e}^{2\lambda(t-T_1)}\tilde{V},$$
$$\frac{\mathrm{d}\tilde{V}}{\mathrm{d}t} \le \frac{a}{2}\mathrm{e}^{2\lambda(t-T_1)+\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2 + \sqrt{2}aq^2\mathrm{e}^{2\lambda(t-T_1)+2\varepsilon}\tilde{U}.$$

Consider $\tilde{W} \coloneqq \max\{\tilde{V}, \tilde{U}\}$. We get

$$\frac{\mathrm{d}\tilde{W}}{\mathrm{d}t} \le \lambda\tilde{W}^2 + 2a\mathrm{e}^{2\lambda(t-T_1)+\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2 + 8aq^2\mathrm{e}^{2\lambda(t-T_1)+2\varepsilon}\tilde{W}.$$

Again, consider $W \coloneqq \tilde{W}h(t)^4$, we get

$$\frac{\mathrm{d}W}{\mathrm{d}t} \le \lambda\mathrm{e}^{4\varepsilon}W^2 + 2a\mathrm{e}^{2\lambda(t-T_1)+5\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2.$$

Now, consider the following system

$$\frac{\mathrm{d}\bar{W}}{\mathrm{d}t} = \lambda\mathrm{e}^{4\varepsilon}\bar{W}^2 + 2a\mathrm{e}^{2\lambda(t-T_1)+5\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2.$$

where $\bar{W}(T_1) = W(T_1)$. It is easy to see that $\bar{W} \ge W$ since the right-hand part is the same function in both cases, this function is increasing in W, and the initial conditions are the same. Also, notice that $\bar{W}$ is increasing. It implies

$$\bar{W}(t) \le \bar{W}(T_1) + \lambda\mathrm{e}^{4\varepsilon}\bar{W}(t)^2(t - T_1) + \frac{a}{\lambda}\mathrm{e}^{2\lambda(t-T_1)+5\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2.$$

Assuming that $\lambda \mathrm{e}^{4\varepsilon}\bar{W}(t)(t-T_1) \le \frac{1}{2}$, we get

$$\bar{W} \le 2\bar{W}(T_1) + \frac{2a}{\lambda}\mathrm{e}^{2\lambda(t-T_1)+5\varepsilon}\|\boldsymbol{\theta}(T_1)\|_R^2 \le 2\bar{W}(T_1) + \frac{2\varepsilon \mathrm{e}^{5\varepsilon}}{q^2}\|\boldsymbol{\theta}(T_1)\|_R^2 = \mathrm{O}(r^{\kappa^*} + r^{\kappa_R}),$$

where $\kappa_R \coloneqq \min_{j\in R}\kappa_j$. Thus, $\lambda \mathrm{e}^{4\varepsilon}\bar{W}(t)(t-T_1) = \mathrm{O}(-\ln(r)(r^{\kappa^*} + r^{\kappa_R}))$. So, for sufficiently small r, our assumption holds.

Therefore, we get the following bounds

$$U = \mathrm{O}(r^{\kappa^*} + r^{\kappa_R}),\ V = \mathrm{O}(r^{\kappa^*} + r^{\kappa_R}),\ \forall j \in R\ |u_j| = \mathrm{O}(r^{\kappa_j}).$$

C.2. Finding a limit of Equation (7)

Now, we want to find $\lim_{r\to 0}\boldsymbol{\theta}^{\chi}(T_2^{\varepsilon})$ to couple $\boldsymbol{\theta}(T_2^{\varepsilon})$ with the vector that does not depend on r. To do this, we want to shift time $t \mapsto t - T_2^{\varepsilon}$ and consider Equation (7) for different r.

$$\begin{aligned}\frac{\mathrm{d}u_j^{\chi,r}}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta}^{\chi,r},\boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_j^{\chi,r},\boldsymbol{x}_i)y_i,\\ \frac{\mathrm{d}\boldsymbol{v}_j^{\chi,r}}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta}^{\chi,r},\boldsymbol{x}_i)y_i))u_j^{\chi,r}\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j^{\chi,r},\boldsymbol{x}_i)y_i,\end{aligned} \tag{8}$$

where

$$u_j^{\chi,r}(-t_4^{\varepsilon,r}) = \begin{cases} ru_j^*, & j\in P,\\ 0, & j\in R,\end{cases}\quad \boldsymbol{v}_j^{\chi,r}(-t_4^{\varepsilon,r}) = \begin{cases} r|u_j^*|\hat{\boldsymbol{v}}_j^*, & j\in P,\\ 0, & j\in R,\end{cases}$$

and $t_4^{\varepsilon,r} \coloneqq \frac{1}{2\lambda}\ln\left(\frac{\lambda\varepsilon}{2ar^2\|\boldsymbol{\theta}^*\|_{[m]}^2}\right)$.

Similarly to the previous subsections, we get

$$\forall t \in [-t_4^{\varepsilon,r}, 0]\ \|\boldsymbol{\theta}^{\chi,r}\|_{[m]}^2 \le 2r^2\|\boldsymbol{\theta}^*\|_{[m]}^2\mathrm{e}^{2\lambda(t+t_4^{\varepsilon,r})},$$

and

$$\alpha^r(t) \coloneqq \arccos((\hat{\boldsymbol{v}}_j^*)^{\mathsf{T}}\hat{\boldsymbol{v}}_j^{\chi,r}) \le \frac{a}{\lambda}r^2\|\boldsymbol{\theta}^*\|_{[m]}^2(\mathrm{e}^{2\lambda(t+t_4^{\varepsilon,r})} - 1).$$

Therefore, for $r' \ge r$

$$\alpha^r(-t_4^{\varepsilon,r'}) \le \frac{a}{\lambda}r^2\|\boldsymbol{\theta}^*\|_{[m]}^2(\mathrm{e}^{2\lambda(-t_4^{\varepsilon,r'}+t_4^{\varepsilon,r})} - 1) = \frac{a}{\lambda}\|\boldsymbol{\theta}^*\|_{[m]}^2((r')^2 - r^2) = \mathrm{O}((r')^2).$$

For $u_j^{\chi,r}$, we get

$$\begin{aligned}\frac{\mathrm{d}(s_j u_j^{\chi,r}\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})})}{\mathrm{d}t} &= |u_j^{\chi,r}|\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})}\left(\lambda(\cos(\alpha^r)-1) + \frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta}^{\chi,r},\boldsymbol{x}_i)y_i) + \ell'(0))|u_j^{\chi}|\phi(\hat{\boldsymbol{v}}_j^{\chi},\boldsymbol{x}_i)y_i\right)\\ &\le a\|\boldsymbol{\theta}^{\chi,r}\|_{[m]}^2|u_j^{\chi,r}|\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})}\\ \implies \quad \left|\frac{u_j^{\chi,r}\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})}}{ru^*}\right| &\le \exp\left(a\int_{-t_4^{\varepsilon,r}}^{t}\|\boldsymbol{\theta}^{\chi,r}\|_{[m]}^2\right) \le \exp\Big(\frac{a}{\lambda}r^2\|\boldsymbol{\theta}^*\|_{[m]}^2(\mathrm{e}^{2\lambda(t+t_4^{\varepsilon,r})} - 1)\Big).\end{aligned}$$

Notice that since $\varepsilon \le 1/2$, we get $\alpha^r/4 \le \varepsilon/4 \le 1$. Thus,

$$1 - \cos(\alpha^r) \le \frac{(\alpha^r)^2}{2} \le 2\alpha^r.$$

It gives

$$\frac{\mathrm{d}(s_j u_j^{\chi,r}\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})})}{\mathrm{d}t} \geq |u_j^{\chi,r}|\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})}(\lambda(\cos(\alpha^r)-1)-a\|\boldsymbol{\theta}^{\chi,r}\|_{[m]}^2) \geq |u_j^{\chi,r}|\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})}(-2\lambda\alpha^r - a\|\boldsymbol{\theta}^{\chi,r}\|_{[m]}^2)$$
$$\implies |\frac{u_j^{\chi,r}\mathrm{e}^{-\lambda(t+t_4^{\varepsilon,r})}}{ru^*}| \geq \exp\left(\frac{-2a}{\lambda}r^2\|\boldsymbol{\theta}^*\|_{[m]}^2(\mathrm{e}^{2\lambda(t+t_4^{\varepsilon,r})}-1)\right).$$

Therefore,

$$\left|\ln\left(\frac{u_j^{\chi,r}(-t_4^{\varepsilon,r'})}{u_j^{\chi,r'}(-t_4^{\varepsilon,r'})}\right)\right| \leq \frac{2a}{\lambda}\|\boldsymbol{\theta}^*\|_{[m]}^2((r')^2-r^2) \implies |u_j^{\chi,r}(-t_4^{\varepsilon,r'})-u_j^{\chi,r'}(-t_4^{\varepsilon,r'})| \leq \mathrm{O}((r')^3).$$

Notice that the above derivations also imply that

$$u_j^{\chi,r}(0) = \Theta(ru_j^*\mathrm{e}^{\lambda t_4^{\varepsilon,r}}) = \Theta(\sqrt{\varepsilon}).$$

Thus, we can use the results of previous subsection with $\kappa = 2$ and get that

$$\|\hat{\boldsymbol{v}}_j^{\chi,r}(0)-\hat{\boldsymbol{v}}_j^{\chi,r'}(0)\| = \mathrm{O}((r')^2), |u_j^{\chi,r}(0)-u_j^{\chi,r'}(0)| = \mathrm{O}((r')^2).$$

This property ensures that the sequences $\hat{\boldsymbol{v}}_j^{\chi,r}(0)$ and $u_j^{\chi,r}(0)$ are fundamental. Since $u_j^{\chi,r}(0)$ is also bounded, $\exists \lim_{r\to 0}\boldsymbol{\theta}^{\chi,r}(0) = \boldsymbol{\theta}^{\chi,\varepsilon,*}$. Since $\boldsymbol{\theta}^{\chi,r}(0)$ is an image of embedding (4), we could find $\boldsymbol{\theta}^{\varepsilon,*}$ such that $\boldsymbol{\theta}^{\chi,\varepsilon,*} = \chi(\boldsymbol{\theta}^{\varepsilon,*})$. Also, since we have $\forall j \in P\ \|\hat{\boldsymbol{v}}_j^{\chi,\varepsilon,r}(0)-\hat{\boldsymbol{v}}^*\| \leq \varepsilon$, this property will also hold for the limit parameters.

C.3. Final Bound

Finally, using the results of both subsections, we have

$$\begin{aligned}
\forall j \in P \quad & |u_j(T_2^\varepsilon) - u_j^{\chi,\varepsilon,*}| = \mathrm{O}(r^2 + r^{\kappa^*} + r^{\kappa_R}),\\
\forall j \in P \quad & \|\hat{\boldsymbol{v}}_j(T_2^\varepsilon) - \hat{\boldsymbol{v}}_j^{\chi,\varepsilon,*}\| = \mathrm{O}(r^2 + r^{\kappa^*} + r^{\kappa_R}),\\
\forall j \in R \quad & |u_j(T_2^\varepsilon)| = \mathrm{O}(r^{\kappa_j}).
\end{aligned}$$

D. Proofs for Section 5

D.1. Proof of Lemma 5.2

We only proof that $\boldsymbol{e}_1$ and $-\boldsymbol{e}_1$ are the maxima of function G. The proof that $\boldsymbol{e}_2$ and $-\boldsymbol{e}_2$ are the minima is similar.

First, we have

$$G(\boldsymbol{e}_1) = \frac{-\ell'(0)}{n}\left(\sum_{i\in S_1}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) + \sum_{i\in S_3}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) - \sum_{i\in S_2}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) - \sum_{i\in S_4}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)\right).$$

Notice

$$\begin{aligned}
&\sum_{i\in S_1}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) + \sum_{i\in S_3}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) = \sum_{i\in S_1}\boldsymbol{x}_i^1 \geq \frac{n}{4}(1-\delta),\\
&\sum_{i\in S_2}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) + \sum_{i\in S_4}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i) \leq \sum_{i\in S_2}(\boldsymbol{x}_i^1+\xi)_+ + \sum_{i\in S_4}(\boldsymbol{x}_i^1+\xi)_+ \leq \frac{n}{2}(\delta+\xi).
\end{aligned}$$

Thus,

$$G(\boldsymbol{e}_1) \geq \frac{-\ell'(0)}{4}(1-3\delta-2\xi).$$

Let $\hat{\boldsymbol{v}}^*$ be a maximum direction of G. W.l.o.g., assume that $\boldsymbol{e}_1^\mathsf{T}\hat{\boldsymbol{v}}^* \geq 0$. Then we have

$$\begin{aligned}\sum_{i\in S_1}\phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i)+\sum_{i\in S_3}\phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i) \leq& \frac{n}{4}(\boldsymbol{e}_1^\mathsf{T}\hat{\boldsymbol{v}}^*+2\xi+2\delta),\\ \sum_{i\in S_2}\phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i)+\sum_{i\in S_4}\phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i) \geq& 0.\end{aligned}$$

Therefore,

$$G(\hat{\boldsymbol{v}}^*) \leq \frac{-\ell'(0)}{4}(\boldsymbol{e}_1^\mathsf{T}\hat{\boldsymbol{v}}^*+2\xi+2\delta).$$

These properties imply

$$\boldsymbol{e}_1^\mathsf{T}\hat{\boldsymbol{v}}^* \geq 1-5\delta-4\xi > \delta+\xi.$$

Hence, $\hat{\boldsymbol{v}}^*$ should satisfy

$$\begin{aligned}&\forall i\in S_3 \quad (\hat{\boldsymbol{v}}^*)^\mathsf{T}\boldsymbol{x}_i \leq -\xi \quad \Longrightarrow \phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i)=0,\\ &\forall i\in S_1 \quad (\hat{\boldsymbol{v}}^*)^\mathsf{T}\boldsymbol{x}_i \geq \xi \quad \Longrightarrow \phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i)=(\hat{\boldsymbol{v}}^*)^\mathsf{T}\boldsymbol{x}_i.\end{aligned}$$

Thus,

$$\sum_{i\in S_3}\phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i)+\sum_{i\in S_1}\phi(\hat{\boldsymbol{v}}^*,\boldsymbol{x}_i)=(\hat{\boldsymbol{v}}^*)^\mathsf{T}\left(\sum_{i\in S_1}\boldsymbol{x}_i\right).$$

Similarly,

$$\sum_{i\in S_3}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)+\sum_{i\in S_1}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)=\boldsymbol{e}_1^\mathsf{T}\left(\sum_{i\in S_1}\boldsymbol{x}_i\right)\geq\frac{n}{4}(1-\delta).$$

Notice that, due to symmetry $\boldsymbol{R}_2\boldsymbol{R}_r D_{\boldsymbol{x}}=D_{\boldsymbol{x}}$, $\sum_{i\in S_1}\boldsymbol{x}_i$ is the multiple of $\boldsymbol{e}_1$. Thus,

$$(\hat{\boldsymbol{v}}^*)^\mathsf{T}\left(\sum_{i\in S_1}\boldsymbol{x}_i\right)=\hat{\boldsymbol{v}}^{*,1}\boldsymbol{e}_1^\mathsf{T}\left(\sum_{i\in S_1}\boldsymbol{x}_i\right).$$

Now, consider the points from S_2 and S_4. Define $\boldsymbol{x}_i^r$ by the following equality $(x_i^1,(\boldsymbol{x}_i^r)^\mathsf{T})^\mathsf{T}=\boldsymbol{x}_i$. Since $\boldsymbol{R}_2\boldsymbol{R}_r D_{\boldsymbol{x}}=D_{\boldsymbol{x}}$, all points $i\in S_2$ have corresponding point $i^-\in S_4$ such that $\boldsymbol{x}_{i^-}=\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x}_i$. Since $\boldsymbol{R}_1 D_{\boldsymbol{x}}=D_{\boldsymbol{x}}$, all $i\in S_2$ have a corresponding point $i^+\in S_2$ such that $\boldsymbol{x}_{i^+}=\boldsymbol{R}_1\boldsymbol{x}_i$. Thus,

$$\begin{aligned}\sum_{i\in S_2}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\sum_{i\in S_4}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)&=\sum_{i\in S_2}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{i^-})\\ &=\sum_{\substack{i\in S_2\\ x_i^1>0}}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{i^-})+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{i^+})+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{(i^+)^-})+\sum_{\substack{i\in S_2\\ x_i^1=0}}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{i^-}).\end{aligned}$$

Now,

$$\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{i^-})=\int_{\boldsymbol{z}\in\mathrm{D}^d}((\hat{\boldsymbol{v}})^\mathsf{T}(\boldsymbol{x}_i+\xi\boldsymbol{z}))_+ + ((\hat{\boldsymbol{v}})^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r(\boldsymbol{x}_i+\xi\boldsymbol{z}))_+ Q(\mathrm{d}\boldsymbol{z}).$$

Denote $\boldsymbol{x}\coloneqq\boldsymbol{x}_i+\xi\boldsymbol{z}$ and consider integrand

$$((\hat{\boldsymbol{v}})^\mathsf{T}\boldsymbol{x})_+ + ((\hat{\boldsymbol{v}})^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x})_+ = (\hat{\boldsymbol{v}}^1x^1+(\hat{\boldsymbol{v}}^r)^\mathsf{T}\boldsymbol{x}^r)_+ + (\hat{\boldsymbol{v}}^1x^1-(\hat{\boldsymbol{v}}^r)^\mathsf{T}\boldsymbol{x}^r)_+ \geq (\hat{\boldsymbol{v}}^1x^1)_+.$$

Therefore,

$$\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_{i^-})\geq\int_{\boldsymbol{z}\in\mathrm{D}^d}(\hat{\boldsymbol{v}}^1(x_i^1+\xi z^1))_+ Q(\mathrm{d}\boldsymbol{z}).$$

So,

$$\sum_{i\in S_2}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\sum_{i\in S_4}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)\geq\sum_{\substack{i\in S_2\\ x_i^1>0}}\int_{\boldsymbol{z}\in\mathrm{D}^d}|\hat{\boldsymbol{v}}^1||x_i^1+\xi z^1|Q(\mathrm{d}\boldsymbol{z})+\sum_{\substack{i\in S_2\\ x_i^1=0}}\int_{\boldsymbol{z}\in\mathrm{D}^d}|\hat{\boldsymbol{v}}^1|(\xi z^1)_+ Q(\mathrm{d}\boldsymbol{z}).$$

Similarly,

$$\sum_{i\in S_2}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)+\sum_{i\in S_4}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)=\sum_{\substack{i\in S_2\\ x_i^1>0}}\int_{\boldsymbol{z}\in\mathrm{D}^d}|x_i^1+\xi z^1|Q(\mathrm{d}\boldsymbol{z})+\sum_{\substack{i\in S_2\\ x_i^1=0}}\int_{\boldsymbol{z}\in\mathrm{D}^d}(\xi z^1)_+Q(\mathrm{d}\boldsymbol{z})\le\frac{n}{2}(\delta+\xi).$$

Thus,

$$\begin{aligned}\sum_{i\in S_2}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)+\sum_{i\in S_4}\phi(\hat{\boldsymbol{v}},\boldsymbol{x}_i)-&\sum_{i\in S_2}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)-\sum_{i\in S_4}\phi(\boldsymbol{e}_1,\boldsymbol{x}_i)\\ &\ge(|\hat{\boldsymbol{v}}^1|-1)\left(\sum_{\substack{i\in S_2\\ x_i^1>0}}\int_{\boldsymbol{z}\in\mathrm{D}^d}|x_i^1+\xi z^1|Q(\mathrm{d}\boldsymbol{z})+\sum_{\substack{i\in S_2\\ x_i^1=0}}\int_{\boldsymbol{z}\in\mathrm{D}^d}(\xi z^1)_+Q(\mathrm{d}\boldsymbol{z})\right)\\ &\ge\frac{n}{2}(|\hat{\boldsymbol{v}}^1|-1)(\delta+\xi).\end{aligned}$$

So, we have

$$0\ge G(\boldsymbol{e}_1)-G(\hat{\boldsymbol{v}}^*)\ge\frac{(-\ell'(0))}{4}(1-\hat{\boldsymbol{v}}^{*,1})(1-\delta)-\frac{(-\ell'(0))}{2}(1-\hat{\boldsymbol{v}}^{*,1})(\delta+\xi)\ge\frac{(-\ell'(0))}{4}(1-\hat{\boldsymbol{v}}^{*,1})(1-3\delta-2\xi).$$

Therefore, $\hat{\boldsymbol{v}}^{*,1}=1\implies\hat{\boldsymbol{v}}^*=\boldsymbol{e}_1$.

D.2. Proof of Lemma 5.3

W.l.o.g., consider direction $\boldsymbol{e}_1$. We will show that this direction is attractive for positive neurons ($s_j=1$) at the beginning of training in the region $\boldsymbol{e}_1^\mathsf{T}\hat{\boldsymbol{v}}^*\ge\delta+\xi$. Thus, the direction $\boldsymbol{e}_1$ captures the random neuron with probability greater than $h\coloneqq\frac{1}{2}\frac{\mathrm{Vol}(A)}{\mathrm{Vol}(\mathrm{S}^{d-1})}\ge\frac{1}{2}\left(\frac{1}{2}-\frac{\mathrm{Vol}(\mathrm{D}^{d-2})}{\mathrm{Vol}(\mathrm{S}^{d-1})}\arcsin(\delta+\xi)\right)=\frac{1}{4}(1-\mathrm{O}(\delta+\xi))$, where $A=\left\{\boldsymbol{x}\in\mathrm{S}^{d-1}\,\middle|\,\boldsymbol{e}_1^\mathsf{T}\boldsymbol{x}\ge\delta+\xi\right\}$. This bound implies the following probability of success

$$\Pr(\text{success})\ge(1-(1-h)^m)^4=1-4(1-h)^m-\mathrm{O}((1-h)^{2m})=1-4\left(\frac{3}{4}\right)^m(1+\mathrm{O}(\delta+\xi))-\mathrm{O}\left(\left(\frac{9}{16}\right)^m\right).$$

Now, we will proof that A is attractive region for $\boldsymbol{e}_1$. Consider positive neuron with $\hat{\boldsymbol{v}}\in\mathrm{S}^{d-1}$ such that $\boldsymbol{e}_1^\mathsf{T}\hat{\boldsymbol{v}}\ge\delta+\xi$. Equation (5) gives

$$\frac{\mathrm{d}\hat{\boldsymbol{v}}}{\mathrm{d}t}=\boldsymbol{P}_{\hat{\boldsymbol{v}}}\boldsymbol{g}(\hat{\boldsymbol{v}}).$$

We have

$$\frac{\mathrm{d}\|\hat{\boldsymbol{v}}-\boldsymbol{e}_1\|^2}{\mathrm{d}t}=-2\boldsymbol{e}_1^\mathsf{T}\frac{\mathrm{d}\hat{\boldsymbol{v}}}{\mathrm{d}t}.$$

So, we only need to show that $\boldsymbol{e}_1\frac{\mathrm{d}\hat{\boldsymbol{v}}}{\mathrm{d}t}$ is positive.

Denote $\hat{\boldsymbol{v}}=:\boldsymbol{e}_1\cos(\alpha)+\boldsymbol{\epsilon}\sin(\alpha)$. Notice

$$\boldsymbol{P}_{\hat{\boldsymbol{v}}}\boldsymbol{e}_1=\boldsymbol{e}_1-\hat{\boldsymbol{v}}\cos(\alpha)=\boldsymbol{e}_1\sin(\alpha)^2-\boldsymbol{\epsilon}\sin(\alpha)\cos(\alpha).$$

Similarly to the proof of Lemma 5.2, we have

$$\boldsymbol{g}(\hat{\boldsymbol{v}})=\frac{-\ell'(0)}{n}\left(\sum_{i\in S_1}\boldsymbol{x}_i-\sum_{i\in S_2}\nabla_{\boldsymbol{v}}(\phi(\boldsymbol{z},\boldsymbol{x}_i)+\phi(\boldsymbol{z},\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x}_i))|_{\boldsymbol{z}=\hat{\boldsymbol{v}}}\right).$$

Notice

$$\begin{aligned}&\nabla_{\boldsymbol{v}}(\phi(\boldsymbol{z},\boldsymbol{x}_i)+\phi(\boldsymbol{z},\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x}_i))|_{\boldsymbol{z}=\hat{\boldsymbol{v}}}\\ &\qquad=\int_{\boldsymbol{z}\in\mathrm{D}^d}([\hat{\boldsymbol{v}}^\mathsf{T}(\boldsymbol{x}_i+\xi\boldsymbol{z})\ge0](\boldsymbol{x}_i+\xi\boldsymbol{z})+[\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r(\boldsymbol{x}_i+\xi\boldsymbol{z})\ge0]\boldsymbol{R}_2\boldsymbol{R}_r(\boldsymbol{x}_i+\xi\boldsymbol{z}))Q(\mathrm{d}\boldsymbol{z}).\end{aligned}$$

Consider the integrand. Denote $\boldsymbol{x}\coloneqq\boldsymbol{x}_i+\xi\boldsymbol{z}$. We have four potential cases:

1. $\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{x} \le 0 \wedge \hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x} \le 0$,
2. $\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{x} > 0 \wedge \hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x} \le 0$,
3. $\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{x} \le 0 \wedge \hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x} > 0$,
4. $\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{x} > 0 \wedge \hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x} > 0$.

In the first case, the contribution of the integrand to expression $\boldsymbol{e}_1 \frac{\mathrm{d}\hat{\boldsymbol{v}}}{\mathrm{d}t}$ is zero. In the fourth case, the contribution of the integrand is

$$-(\boldsymbol{P}_{\hat{\boldsymbol{v}}}\boldsymbol{e}_1)^\mathsf{T}(\boldsymbol{x} + \boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x}) = -2x^1\sin(\alpha)^2 \ge -2(\delta+\xi)\sin(\alpha)^2.$$

The second and third cases are symmetric. So, we consider only the second case. In the second case, we have

$$\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{x} = x^1\cos(\alpha) + \boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{x}^r\sin(\alpha) \ge 0.$$

Thus, $\boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{x}^r \ge 0$ (otherwise $\hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{R}_2\boldsymbol{R}_r\boldsymbol{x} \ge \hat{\boldsymbol{v}}^\mathsf{T}\boldsymbol{x} > 0$). It implies

$$-(\boldsymbol{P}_{\hat{\boldsymbol{v}}}\boldsymbol{e}_1)^\mathsf{T}\boldsymbol{x} = -x^1\sin(\alpha)^2 + \boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{x}\cos(\alpha)\sin(\alpha) \ge -x^1\sin(\alpha)^2.$$

Therefore, we get

$$\boldsymbol{e}_1\frac{\mathrm{d}\hat{\boldsymbol{v}}}{\mathrm{d}t} \ge \frac{-\ell'(0)}{n}\Big(\frac{n}{4}(1-\delta) - \frac{n}{4}(2(\delta+\xi))\Big)\sin(\alpha)^2 \ge \frac{-\ell'(0)}{8}\sin(\alpha)^2.$$

It gives

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} \le \frac{-\ell'(0)}{8}\sin(\alpha) \le \frac{-\ell'(0)}{4\pi}\alpha,$$

which implies exponentially fast convergnce to $\boldsymbol{e}_1$.

D.3. Proof of Proposition 5.7

First, notice that, due to symmetry $\boldsymbol{R}_r D_{\boldsymbol{x}} = D_{\boldsymbol{x}}$, the features of 4-neuron network belong to a sub-space generated by vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ and clustered around these directions.

Lemma D.1. *$\boldsymbol{v}_j^{*,\varepsilon}$ form Theorem 4.4 belong to a sub-space generated by vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ ($\boldsymbol{R}_r\boldsymbol{v}_j^{*,\varepsilon} = \boldsymbol{v}_j^{*,\varepsilon}$). Moreover, this symmetry is preserved under gradient flow dynamics (1).*

Proof. Consider Equation (7). Notice that at the start of function $h(\boldsymbol{x}) \coloneqq f(\boldsymbol{\theta}^\chi(T_1), \boldsymbol{x})$ depends only on x^1 and x^2. Due to symmetry $\boldsymbol{R}_r D_{\boldsymbol{x}} = D_{\boldsymbol{x}}$, this property will hold for the right-hand part of Equation (7). Therefore, features will not be able to escape the sub-space generated by $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Thus, limit features $\boldsymbol{v}_j^{*,\varepsilon}$ will also lie in the sub-space generated by $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Since the limit and dataset respect symmetry $\boldsymbol{R}_r$, it will propagate to further stages of training. □

Consider dynamics

$$\frac{\mathrm{d}u_k^e}{\mathrm{d}t} = \sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^e, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_k^e, \boldsymbol{x}_i)y_i,$$
$$\frac{\mathrm{d}\boldsymbol{v}_k^e}{\mathrm{d}t} = \sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^e, \boldsymbol{x}_i)y_i))u_k^e\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_k^e, \boldsymbol{x}_i)y_i,$$

where $u_k^e(0) = s_{P_k}\sqrt{\sum_{j\in P_k}(u_j^{\varepsilon,*})^2}$ and $\boldsymbol{v}_k^e(0) = |u_k^e(0)|\hat{\boldsymbol{v}}_{P_k}^{\varepsilon,*}$. Notice that, by Lemma D.1, $\boldsymbol{v}_k^e \in \operatorname{span}\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$.

Proposition D.2. *Denote a signed angle between $\boldsymbol{R}_1^{\lfloor\frac{k-1}{2}\rfloor}\boldsymbol{P}^{a(k)}\boldsymbol{v}_k^e$ and $\boldsymbol{e}_{a(k)+1}$ by α_k. Then we have $|\sin(\alpha_k)| \le \delta + \xi$.*

Proof. W.l.o.g., consider $\boldsymbol{v}_1^e$ and assume $\alpha_1 \ge 0$. We have

$$\frac{\mathrm{d}\hat{\boldsymbol{v}}_1^e}{\mathrm{d}t} = \sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^e, \boldsymbol{x}_i)y_i))\boldsymbol{P}_{\boldsymbol{v}_1^e}\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_1^e, \boldsymbol{x}_i)y_i.$$

Notice

$$\boldsymbol{e}_1^\mathsf{T}\boldsymbol{P}_{\boldsymbol{v}_1^e}\nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_1^e,\boldsymbol{x}_i)y_i=\int_{\boldsymbol{z}\in\mathrm{D}^d}[(\boldsymbol{v}_1^e)^\mathsf{T}(\boldsymbol{x}_i+\xi\boldsymbol{z})\geq 0]\boldsymbol{e}_1^\mathsf{T}\boldsymbol{P}_{\boldsymbol{v}_1^e}(\boldsymbol{x}_i+\xi\boldsymbol{z})y_iQ(\mathrm{d}\boldsymbol{z}).$$

Consider the integrand and denote $\boldsymbol{z}_i\coloneqq\boldsymbol{x}_i+\xi\boldsymbol{z}$. Assuming that $\alpha_1\leq\pi/4$, we get

$$\begin{aligned}
&\forall i\in S_1\ [(\boldsymbol{v}_1^e)^\mathsf{T}\boldsymbol{z}_i\geq 0]\boldsymbol{e}_1^\mathsf{T}\boldsymbol{P}_{\boldsymbol{v}_1^e}\boldsymbol{z}_iy_i=(\boldsymbol{e}_1\sin(\alpha_1)-\boldsymbol{e}_2\cos(\alpha_1))^\mathsf{T}\sin(\alpha_1)(\boldsymbol{e}_1+\boldsymbol{z}_i-\boldsymbol{e}_1)\geq(\sin(\alpha_1)-(\delta+\xi))\sin(\alpha_1),\\
&\forall i\in S_2\ [(\boldsymbol{v}_1^e)^\mathsf{T}\boldsymbol{z}_i\geq 0]\boldsymbol{e}_1^\mathsf{T}\boldsymbol{P}_{\boldsymbol{v}_1^e}\boldsymbol{z}_iy_i=-[(\boldsymbol{v}_1^e)^\mathsf{T}\boldsymbol{z}_i\geq 0](\boldsymbol{e}_1\sin(\alpha_1)-\boldsymbol{e}_2\cos(\alpha_1))^\mathsf{T}\sin(\alpha_1)(\boldsymbol{e}_2+\boldsymbol{z}_i-\boldsymbol{e}_2)\\
&\qquad\qquad\qquad\qquad\geq(\cos(\alpha_1)-(\delta+\xi))\sin(\alpha_1)\\
&\qquad\qquad\qquad\qquad\geq 0,\\
&\forall i\in S_3\ [(\boldsymbol{v}_1^e)^\mathsf{T}\boldsymbol{z}_i\geq 0]\boldsymbol{e}_1^\mathsf{T}\boldsymbol{P}_{\boldsymbol{v}_1^e}\boldsymbol{z}_iy_i=0,\\
&\forall i\in S_4\ [(\boldsymbol{v}_1^e)^\mathsf{T}\boldsymbol{z}_i\geq 0]\boldsymbol{e}_1^\mathsf{T}\boldsymbol{P}_{\boldsymbol{v}_1^e}\boldsymbol{z}_iy_i\geq-[\sin(\alpha_1)\leq\delta+\xi](1+\delta+\xi).
\end{aligned}$$

Thus, the function $h(\alpha_1)=\max(\alpha_1,\arcsin(\delta+\xi))$ is always decreasing (when α_1 is less than $\arcsin(\delta+\xi)$, the time derivative of $h(\alpha_1)$ is zero, when α_1 is greater than $\arcsin(\delta+\xi)$, the time derivative of $h(\alpha_1)$ is determined only by the points from S_1 and S_2 and hence negative). Therefore, α_1 never exceeds $\arcsin(\delta+\xi)$. □

Proof of Proposition 5.7. Denote $a\coloneqq 0.01$, $b\coloneqq 0.001$, $c=1000$, $q\coloneqq 8/3$, $T\coloneqq\inf\{t>0\mid\max_k|u_k^e(t)|\geq q\}$.

From the previous proposition, we know that $|\sin(\alpha_k)|\leq\delta+\xi\leq a$. Thus, for $t\in[0,T]$ and $i\in S_1$, we get

$$f(\boldsymbol{\theta}^e,x_i)=(u_1^e)^2(\hat{\boldsymbol{v}}_1^e)^\mathsf{T}\boldsymbol{x}_i-(u_2^e)^2\phi(\hat{\boldsymbol{v}}_2^e,\boldsymbol{x}_i)-(u_4^e)^2\phi(\hat{\boldsymbol{v}}_4^e,\boldsymbol{x}_i).$$

Therefore,

$$\begin{aligned}
&f(\boldsymbol{\theta}^e,x_i)\leq(u_1^e)^2(1+\delta)\leq(u_1^e)^2(1+a),\\
&f(\boldsymbol{\theta}^e,x_i)\geq(u_1^e)^2(\cos(\alpha_k)-\delta)-q^2(\phi(\hat{\boldsymbol{v}}_2^e,\boldsymbol{x}_i)+\phi(\hat{\boldsymbol{v}}_4^e,\boldsymbol{x}_i))\geq(u_1^e)^2(1-(\delta+\xi)^2-\delta)-q^2(4\delta+2\xi)\\
&\qquad\geq(u_1^e)^2(1-2a)-4q^2a,
\end{aligned}$$

where the last inequality follows from the following property:

$$\begin{aligned}
\forall j\in\{2,4\}\ \phi(\hat{\boldsymbol{v}}_j^e,\boldsymbol{x}_i)&=\int_{-1}^1((\hat{\boldsymbol{v}}_j^e)^\mathsf{T}\boldsymbol{x}_i+\xi z)_+(1-z^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}z\\
&\leq\int_{-1}^1(|\sin(\alpha_j)|+\delta+\xi z)_+(1-z^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}z\\
&\leq\int_{-1}^1(2\delta+\xi+\xi z)(1-z^2)^{\frac{d-1}{2}}\frac{\mathrm{Vol}(\mathrm{D}^{d-1})}{\mathrm{Vol}(\mathrm{D}^d)}\,\mathrm{d}z\\
&=2\delta+\xi.
\end{aligned}$$

Similarly, we could derive the same inequalities for $S_2,\dots,S_4$.

These inequalities imply

$$\begin{aligned}
\frac{\mathrm{d}u_1^e}{\mathrm{d}t}=&\frac{1}{n}\sum_{i=1}^n(-\ell'(f(\boldsymbol{\theta}^e,\boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_1^e,\boldsymbol{x}_i)y_i\\
\geq&\frac{u_1^e}{4}\left(\frac{1-2a}{1+\exp((u_1^e)^2(1+a))}-\frac{2a}{1+\exp((u_2^e)^2(1-2a)-4q^2a)}-\frac{2a}{1+\exp((u_4^e)^2(1-2a)-4q^2a)}\right)
\end{aligned}$$

and

$$\frac{\mathrm{d}u_1^e}{\mathrm{d}t}\leq\frac{u_1^e(1+a)}{4(1+\exp((u_1^e)^2(1-2a)-4q^2a))}.$$

Denote $x \coloneqq \min_k |u_k^e|$ and $y \coloneqq \max_k |u_k^e|$. The property above implies

$$\frac{\mathrm{d}x}{\mathrm{d}t} \geq \frac{x}{4}\left(\frac{1-2a}{1+\exp(x^2(1+a))} - \frac{4a}{1+\exp(x^2(1-2a)-4q^2a)}\right) \geq \frac{x((1-2a)e^{-7q^2a}-4a)}{4(1+\exp(x^2(1-2a)-4q^2a))},$$
$$\frac{\mathrm{d}y}{\mathrm{d}t} \leq \frac{y(1+a)}{4(1+\exp(y^2(1-2a)-4q^2a)}.$$

Now, denote $X \coloneqq x^2(1-2a)$, $Y \coloneqq y^2(1-2a)$, $A \coloneqq \frac{(1-2a)e^{-7q^2a}-4a}{2}$, $B \coloneqq \frac{1+a}{2}$, and $C \coloneqq e^{-4q^2a}$. We get

$$\frac{\mathrm{d}X}{\mathrm{d}t} \geq \frac{AX}{1+C\mathrm{e}^X} \quad \Longrightarrow \quad \int_{X(0)}^{X(t)} \frac{\mathrm{d}X(1+C\mathrm{e}^X)}{X} \geq At,$$
$$\frac{\mathrm{d}Y}{\mathrm{d}t} \leq \frac{BY}{1+C\mathrm{e}^Y} \quad \Longrightarrow \quad \int_{Y(0)}^{Y(t)} \frac{\mathrm{d}Y(1+C\mathrm{e}^Y)}{Y} \leq Bt.$$

Denote $h(x) \coloneqq \int_1^x \frac{\mathrm{d}z(1+C\mathrm{e}^z)}{z}$. We get

$$h(X(t)) - h(X(0)) \geq At,\ h(Y(t)) - h(Y(0)) \leq Bt.$$

We have two possible cases: $T < \infty$ and $T = \infty$. In the second case, we notice that our lower bound for the derivative of x holds on the whole timeline. In this case, $\lim_{t\to\infty} x(t) = \infty$. Thus, at some time point we would get $x > q$, which contradicts definition of T. Therefore, T is finite.

Now, notice that $Y(T) = (1-2a)q^2$ and we have

$$B(h(X(T)) - h(X(0))) \geq ABT \geq A(h(Y(T)) - h(Y(0)))$$
$$\Longrightarrow h(X(T)) \geq \frac{A}{B}h(Y(T)) + \frac{B-A}{B}h(Y(0)) - h(Y(0)) + h(X(0)).$$

By the definition of h,

$$h(Y(T)) = \ln(Y(T)) + C\int_1^{Y(T)} \frac{\mathrm{e}^z}{z}\,\mathrm{d}z,$$
$$h(Y(0)) = -\int_{Y(0)}^1 \frac{1+C\mathrm{e}^z}{z}\,\mathrm{d}z \geq (1+C\mathrm{e})\ln(Y(0)) = (1+C\mathrm{e})\ln((1-2a)b^2),$$
$$h(Y(0)) - h(X(0)) = \int_{X(0)}^{Y(0)} \frac{1+C\mathrm{e}^z}{z}\,\mathrm{d}z \leq (1+C\mathrm{e}^{Y(0)})\ln\left(\frac{Y(0)}{X(0)}\right) \leq (1+C\mathrm{e}^{\frac{1-2a}{4}})\ln(c^2).$$

Using numerical integration, we get $h(X(T)) \geq 31.52$, which implies that $x(T) \geq 9/4$.

Therefore, at time T, we have

$$f(\boldsymbol{\theta}^e(T), \boldsymbol{x}_i)y_i \geq (1-2a)x(T)^2 - 4q^2a > 4.67 > 0,$$

the network classifies all points correctly. □

D.4. Proof of Proposition 5.8

First, consider direction $\boldsymbol{\theta}^{mm}$ and w.l.o.g. assume that $\|\boldsymbol{\theta}^{mm}\|^2 = 8$, which implies $\forall k\ |u_k^{mm}| = 1$. Consider some orbit, M, of data points under the group generated by $\boldsymbol{P}, \boldsymbol{R}_1, \boldsymbol{R}_2, \boldsymbol{R}_r$.

Choose $i^* \in M \cap S_1 : \boldsymbol{e}_2^\mathsf{T}\boldsymbol{x}_{i^*} \geq 0$. For this point, we have

$$f(\boldsymbol{\theta}^{mm}, \boldsymbol{x}_{i^*}) = \sum_{k=1}^4 (u_k^{mm})^2 \phi(\hat{\boldsymbol{v}}_k^{mm}, \boldsymbol{x}_{i^*}) = \phi(\hat{\boldsymbol{v}}_1^{mm}, \boldsymbol{x}_{i^*}) - \phi(\hat{\boldsymbol{v}}_2^{mm}, \boldsymbol{x}_{i^*}).$$

Since $\forall a\ |\boldsymbol{e}_a^\top \boldsymbol{x}_{i^*}| > \xi$, we get

$$f(\boldsymbol{\theta}^{mm}, \boldsymbol{x}_{i^*}) = x_{i^*}^1 - x_{i^*}^2.$$

Due to symmetry of the network, all points in M will have the same margin.

Now, consider a small perturbation, $\boldsymbol{\theta}$, of $\chi(\boldsymbol{\theta}^{mm})$, which does not change activation patterns for points in M. Notice

$$\min_{i\in M} \frac{f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i}{\|\boldsymbol{\theta}\|^2} \le \frac{1}{16\|\boldsymbol{\theta}\|^2} \sum_{i\in M} f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i.$$

However,

$$\begin{aligned}\sum_{i\in M} f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i =& \sum_{k=1}^{4} \sum_{j\in P_k} \sum_{l=1}^{4} \sum_{i\in S_l} u_j \phi(\boldsymbol{v}_j, \boldsymbol{x}_i)y_i = \sum_k \sum_{j\in P_k} 4|u_j||\boldsymbol{v}_j^{a(k)+1}|(x_{i^*}^1 - x_{i^*}^2) \\ \le& \sum_k \sum_{j\in P_k} 2(|u_j|^2 + \|\boldsymbol{v}_j\|^2)(x_{i^*}^1 - x_{i^*}^2).\end{aligned}$$

Thus,

$$\min_{i\in M} \frac{f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i}{\|\boldsymbol{\theta}\|^2} \le \frac{x_i^1 - x_i^2}{8} = \min_{i\in M} \frac{f(\boldsymbol{\theta}^{mm}, \boldsymbol{x}_i)y_i}{\|\boldsymbol{\theta}^{mm}\|^2}.$$

It implies that, for sufficiently small perturbation,

$$\min_i \frac{f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i}{\|\boldsymbol{\theta}\|^2} = \min_M \min_{i\in M} \frac{f(\boldsymbol{\theta}, \boldsymbol{x}_i)y_i}{\|\boldsymbol{\theta}\|^2} \le \min_i \frac{f(\boldsymbol{\theta}^{mm}, \boldsymbol{x}_i)y_i}{\|\boldsymbol{\theta}^{mm}\|^2},$$

where the equality is achieved only if $|u_j| = \|\boldsymbol{v}_j\|$ and $\boldsymbol{v}_j$ is aligned with $b(k)\boldsymbol{e}_{a(k)+1}$. However, in this case,

$$f(\boldsymbol{\theta}, \boldsymbol{x}_{i^*}) = U_1^2 x_{i^*}^1 - U_2^2 x_{i^*}^2,$$

where $U_k = s_{P_k}\sqrt{\sum_{j\in P_k} u_j^2}$. W.l.o.g., we could assume that $U_1^2 = \min_k U_k^2$, then, we get

$$\frac{f(\boldsymbol{\theta}, \boldsymbol{x}_{i^*})}{\|\boldsymbol{\theta}\|^2} \le \frac{U_1^2 x_{i^*}^1 - U_1^2 x_{i^*}^2}{2\sum_k U_k^2}.$$

But this margin should be equal to $\frac{x_{i^*}^1 - x_{i^*}^2}{8}$. Thus, $4U_1^2 \ge \sum_k U_k^2$. However, this implies that $U_1^2 = U_2^2 = U_3^2 = U_4^2$, i.e., $\boldsymbol{\theta}$ is proportional to $\chi(\boldsymbol{\theta}^{mm})$. Therefore, $\chi(\boldsymbol{\theta}^{mm})$ is indeed local extremum. Similarly, we can show that $\boldsymbol{\theta}^{mm}$ is an isolated local extremum.

D.5. Proof of Lemma 5.5

Denote 4-neuron network initialized at $\boldsymbol{\theta}^{\varepsilon,*}$ as $\boldsymbol{\theta}^f$. We get

$$\begin{aligned}\frac{\mathrm{d}u_j^f}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta}^f, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_j^f, \boldsymbol{x}_i)y_i, \\ \frac{\mathrm{d}\boldsymbol{v}_j^f}{\mathrm{d}t} &= \frac{1}{n}\sum_{i=1}^{n}(-\ell'(f(\boldsymbol{\theta}^f, \boldsymbol{x}_i)y_i))u_j^f \nabla_{\boldsymbol{v}}\phi(\boldsymbol{v}_j^f, \boldsymbol{x}_i)y_i,\end{aligned}$$

where $\boldsymbol{\theta}^f(0) = \boldsymbol{\theta}^{\varepsilon,*}$.

First, we want to show that $\|\boldsymbol{\theta}^f\| \xrightarrow{t\to\infty} \infty$. Similarly to the proof of Proposition 5.7, using Proposition D.2, we get that for all $i \in S_1$

$$\begin{aligned}f(\boldsymbol{\theta}^f, \boldsymbol{x}_i) &\le (u_1^e)^2(1+a), \\ f(\boldsymbol{\theta}^f, \boldsymbol{x}_i) &\ge (u_1^e)^2(1-2a) - 2a((u_2^e)^2 + (u_4^e)^2),\end{aligned}$$

where $a \coloneqq \delta + \xi$. Since $\boldsymbol{\theta}^f$ converges in direction to $\boldsymbol{\theta}^{mm}$, we could find a moment when ratio $\frac{8(u_k^e)^2}{\|\boldsymbol{\theta}^f\|^2}$ will lie in interval $(1-\epsilon, 1+\epsilon)$, where $\frac{3\epsilon}{1-\epsilon} < \frac{1-6a}{4a}$. Then,

$$f(\boldsymbol{\theta}^f, \boldsymbol{x}_i) \geq b\|\boldsymbol{\theta}^f\|^2,$$

where $b \coloneqq \frac{(1-\epsilon)(1-2a)-(1+\epsilon)2a}{8} > 0$. Similarly, for other clusters. Also define $c \coloneqq \frac{(1+\epsilon)(1+a)}{8}$. Now, notice that

$$\begin{aligned}\frac{1}{4}\frac{\mathrm{d}\|\boldsymbol{\theta}\|^2}{\mathrm{d}t} =& \sum_{k=1}^{4} u_k^e \frac{\mathrm{d}u_k^e}{\mathrm{d}t} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{4} u_k^e(-\ell'(f(\boldsymbol{\theta}^f, \boldsymbol{x}_i)y_i))\phi(\boldsymbol{v}_k^e, \boldsymbol{x}_i)y_i = \frac{1}{n}\sum_{i=1}^{n}\frac{f(\boldsymbol{\theta}^f, \boldsymbol{x}_i)y_i}{1+\exp(f(\boldsymbol{\theta}^f, \boldsymbol{x}_i)y_i)} \\ \geq& \frac{b\|\boldsymbol{\theta}^f\|^2}{1+\exp(c\|\boldsymbol{\theta}^f\|)}.\end{aligned}$$

Similarly to the proof of Proposition 5.7, this differential inequality means that $\|\boldsymbol{\theta}^f\| \xrightarrow{t\to\infty} \infty$.

Now, we want to apply Theorem 5.6. To do it, choose ζ small enough so that $\chi(\boldsymbol{\theta}^{mm})$ becomes the biggest local-max-margin direction in ζ-neighborhood around its image in the original weight space. Then, apply Theorem 5.6 for $\chi(\boldsymbol{\theta}^{mm})$, which gives us parameters ω and ρ. Consider time T when $\chi(\boldsymbol{\theta}^f)$ converged to the desired local-max-margin direction closer than $\omega/2$ and its scale became bigger than 2ρ. After that, apply Theorem 2.1, Chapter 5 from Hartman (2002) (notice that our activation function is twice continuously differentiable) and choose the initial scale σ to be sufficiently small so that original system at time $\boldsymbol{\theta}(T + T_2^{\varepsilon})$ is bigger than ρ and $\omega/2$-close in direction to $\boldsymbol{\theta}^f$. Then, $\boldsymbol{\theta}$ will converge to some direction with normalized margin bigger than that of $\chi(\boldsymbol{\theta}^{mm})$. However, since $\chi(\boldsymbol{\theta}^{mm})$ is a strict local-max-margin direction, this would mean that $\boldsymbol{\theta}$ will converge in direction to $\chi(\boldsymbol{\theta}^{mm})$.

E. Additional Experiments for Section 5

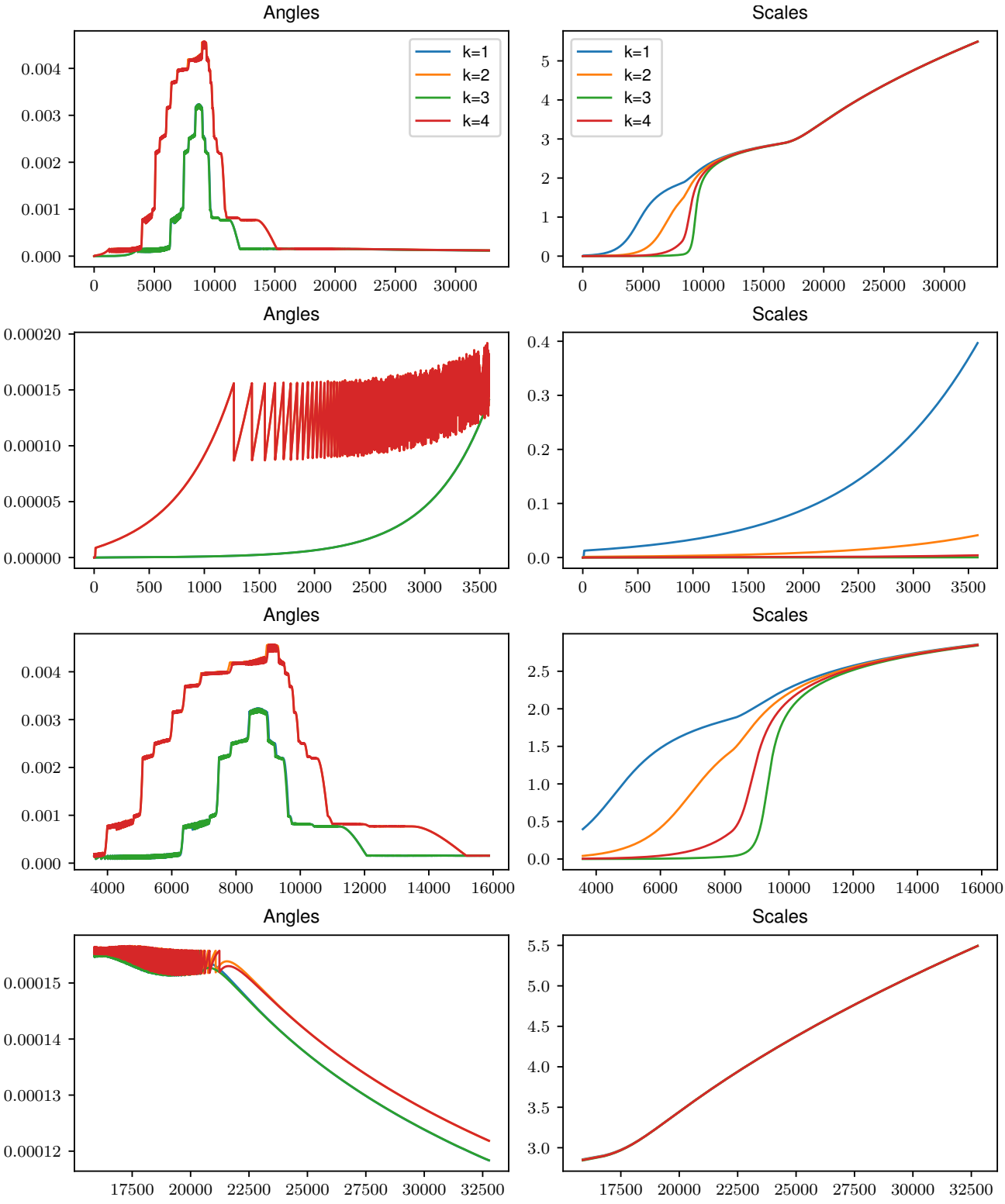

Figure 5. Evolution of 4-neuron network initialized at $(u_1^e(0), u_2^e(0), u_3^e(0), u_4^e(0)) = (10^{-4}, -10^{-5}, 10^{-7}, -10^{-6})$. The first row depicts the whole training process; the second row depicts the first 3584 training epochs; the third row depicts the epochs from 3584 to 15872; the last row depicts training after the 15872th epoch. Notice that $\alpha_1 \approx \alpha_3$ and $\alpha_2 \approx \alpha_4$.

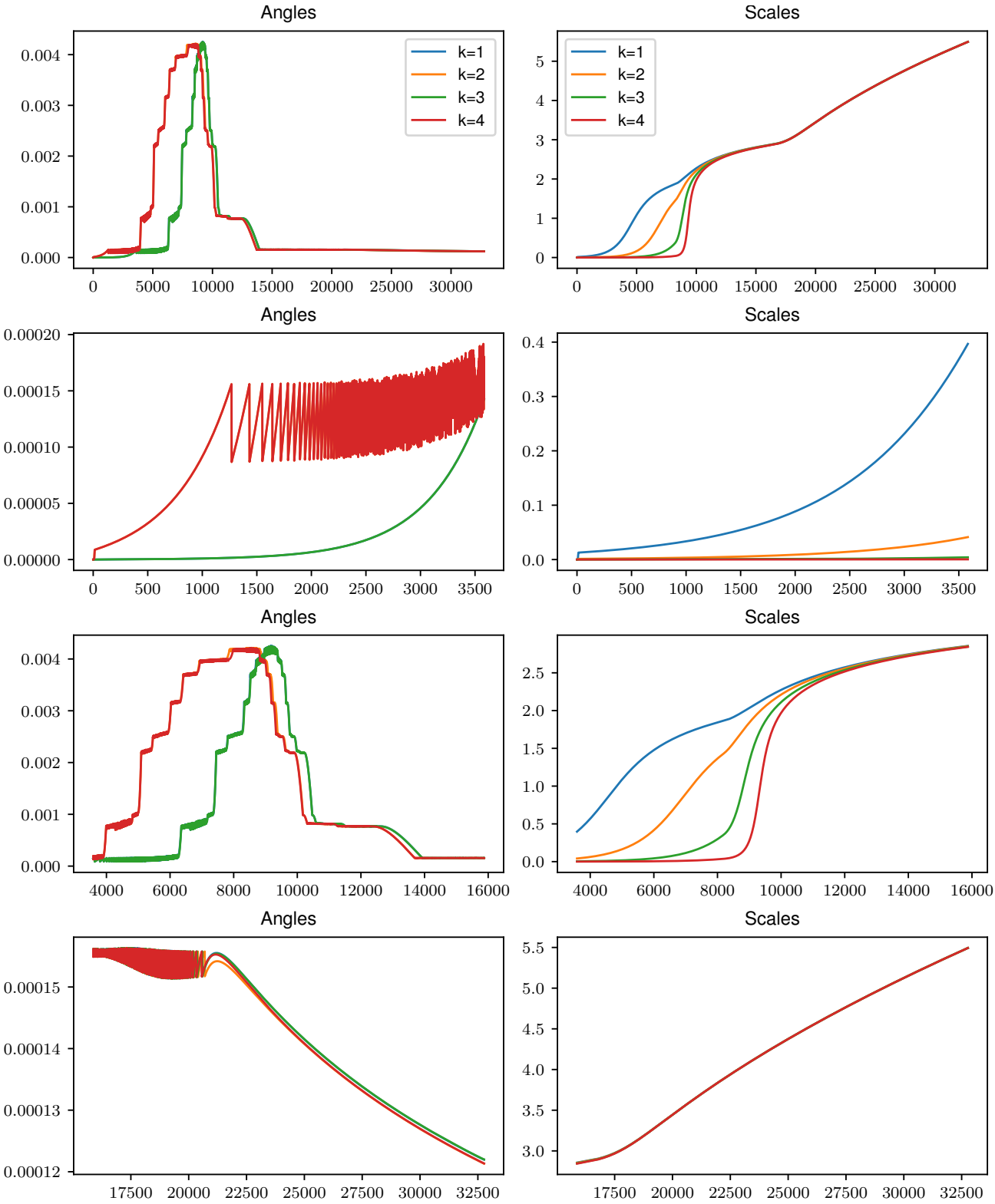

Figure 6. Evolution of 4-neuron network initialized at $(u_1^e(0), u_2^e(0), u_3^e(0), u_4^e(0)) = (10^{-4}, -10^{-5}, 10^{-6}, -10^{-7})$. The first row depicts the whole training process; the second row depicts the first 3584 training epochs; the third row depicts the epochs from 3584 to 15872; the last row depicts training after the 15872th epoch. Notice that $\alpha_1 \approx \alpha_3$ and $\alpha_2 \approx \alpha_4$.

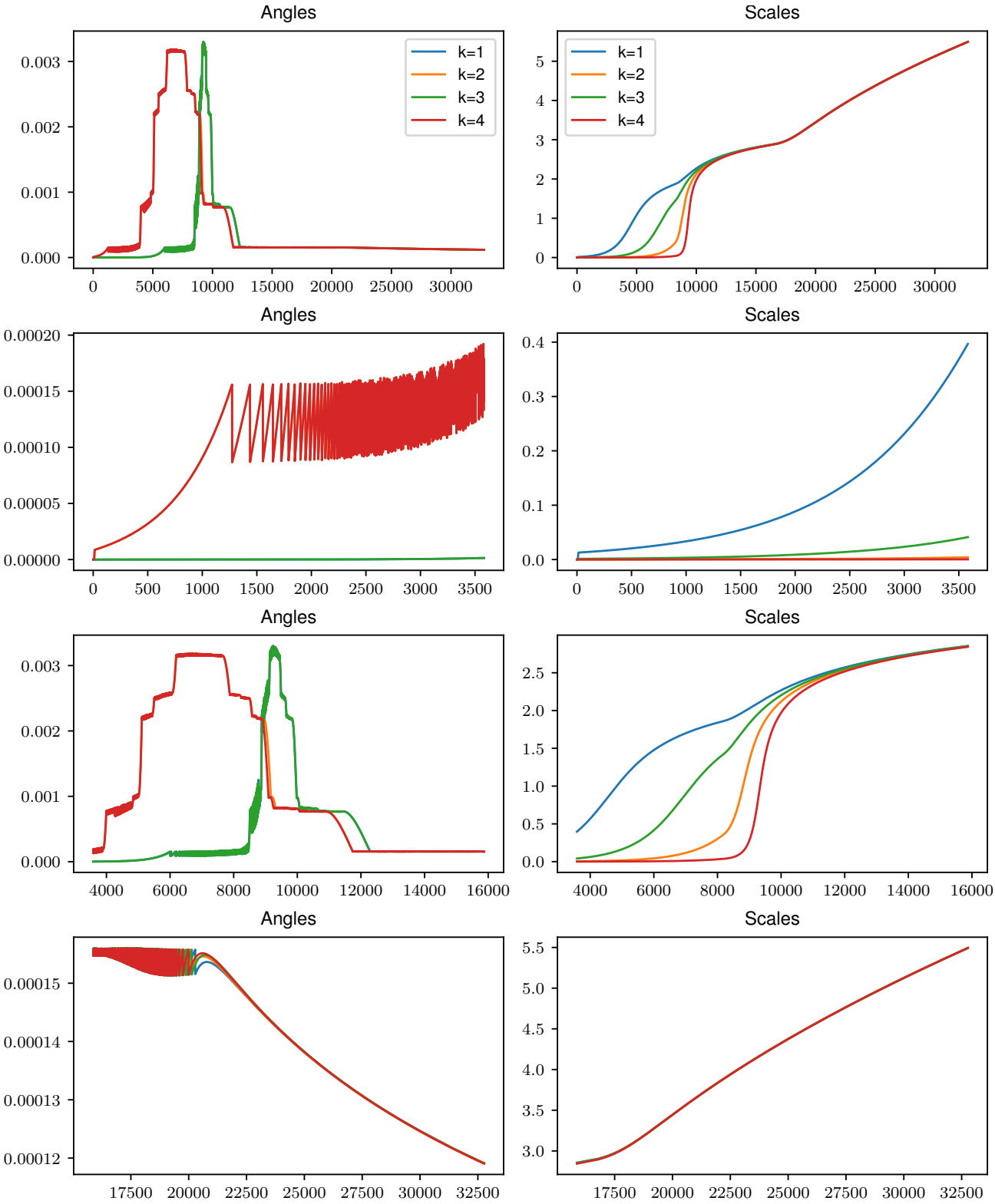

Figure 7. Evolution of 4-neuron network initialized at $(u_1^e(0), u_2^e(0), u_3^e(0), u_4^e(0)) = (10^{-4}, -10^{-6}, 10^{-5}, -10^{-7})$. The first row depicts the whole training process; the second row depicts the first 3584 training epochs; the third row depicts the epochs from 3584 to 15872; the last row depicts training after the 15872th epoch. Notice that $\alpha_1 \approx \alpha_3$ and $\alpha_2 \approx \alpha_4$.

F. Additional Experimental Results

F.1. Experimental Details

Data We used the usual MNIST and CIFAR-10 datasets for creation of dominos. For creating train and test data, we used the default train-test split of these datasets, resulting in 50000 images in train set and 10000 images in test set. We further devoted 25% train and test data for validation, giving us four datasets: train-train, train-validation, test-train, and test-validation. We also normalize images in these datasets using the default values for MNIST and CIFAR-10. During training, we also apply random horizontal flip augmentation.

Model We used the standard model from Torchvision library, but changed the first layer to 3×3 convolutions instead of the default 7×7 convolutions. Additionally, after initialization we multiplied all parameters of the model by a factor 2^{-5} to capture the desired simplicity bias mechanism.

Optimization procedure We use the standard SGD optimizer from PyTorch and linear learning scheduler with warm-up from Transformers library. The parameters of data and optimizer are listed below.

`batch_size`	128
`lr`	0.125
`momentum`	0.9
`nesterov`	True
`weight_decay`	0.0005
Share of warm-up steps	12.5%

Parameters of logistic regression We used the standard implementation of the logistic regression from scikit-learn library. By default, we use the following parameters.

`penalty`	l2
`C`	1000
`max_iter`	20000

Notice that effectively the current version of scikit-learn library does not allow to change the parameter `maxfun` parameter of the `lbfgs` optimizer. Thus, to ensure convergence we rerun fitting procedure 50 times using warm start.

F.2. Additional Experiments

Figures 8, 9, and 10 depict additional results for Section 7.

For Figure 8, we repeated the experiment but used MNIST labels on the test set. As we can see, the learned features are sufficient to achieve almost perfect accuracy on MNIST label, indicating that the network learned "simple" MNIST features.

For Figure 9, we repeated the experiment but did not scale the model at initialization. As we can see, the drop in OOD accuracy is less pronounced for the model initialized from the normal scale (approximately $6.43\% \pm 2.43\%$, implying p-value around 0.00062). This experiment indicates that closeness to more lazy training regime is indeed beneficial for OOD generalization in presence of simplicity bias.

For Figure 10, we repeated the experiment on different train data, on which the correlation between MNIST and CIFAR-10 classes is not perfect and with 5% probability MNIST class might not match CIFAR-10 class. As we can see, the model still experience simplicity bias. However, the drop in OOD accuracy at the end of training disappears.

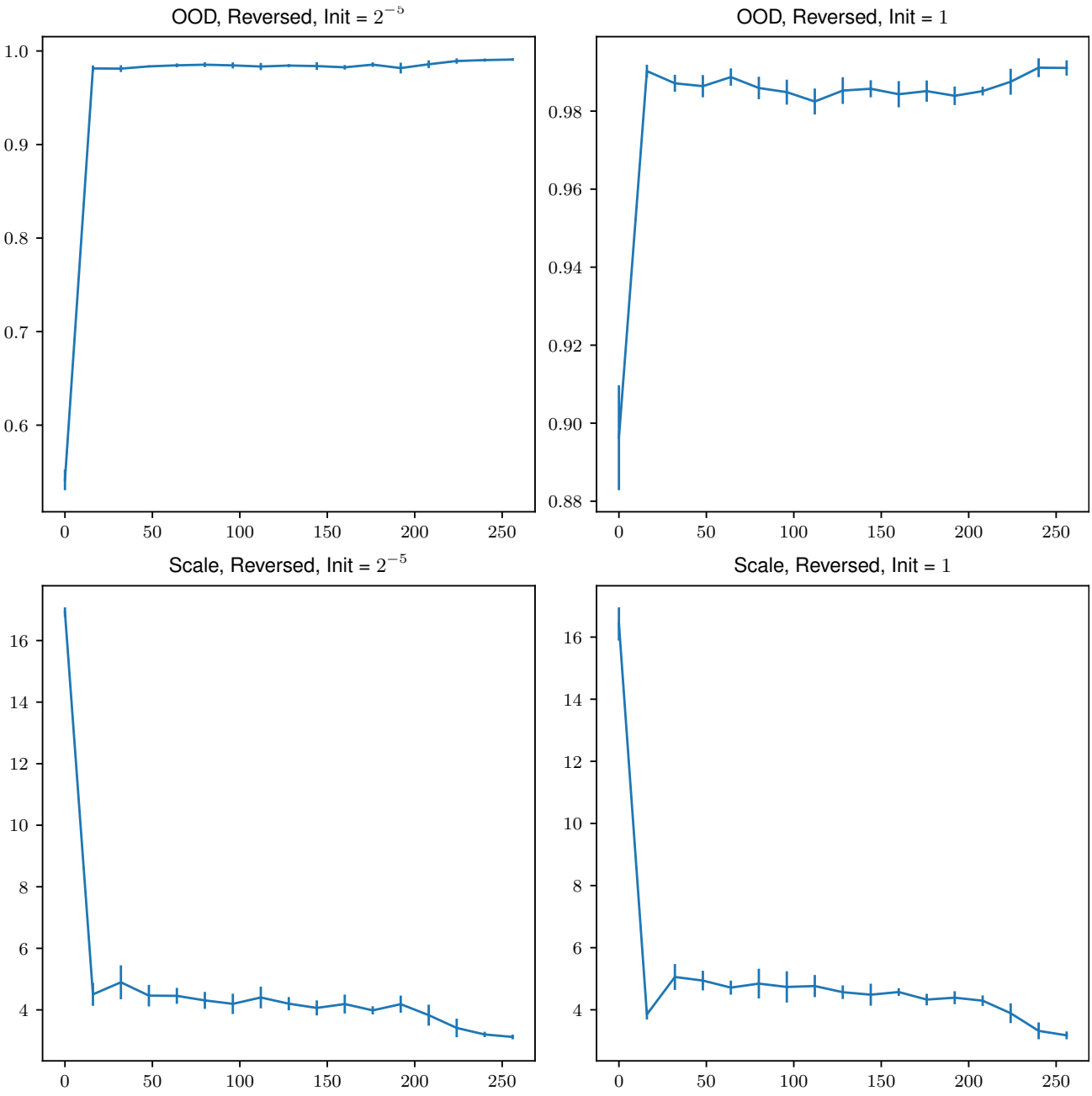

Figure 8. Accuracy and scale of the logistic regression on the validation part of the OOD test set (y-axis) vs. the training epoch at which the ResNet features are extracted (x-axis).

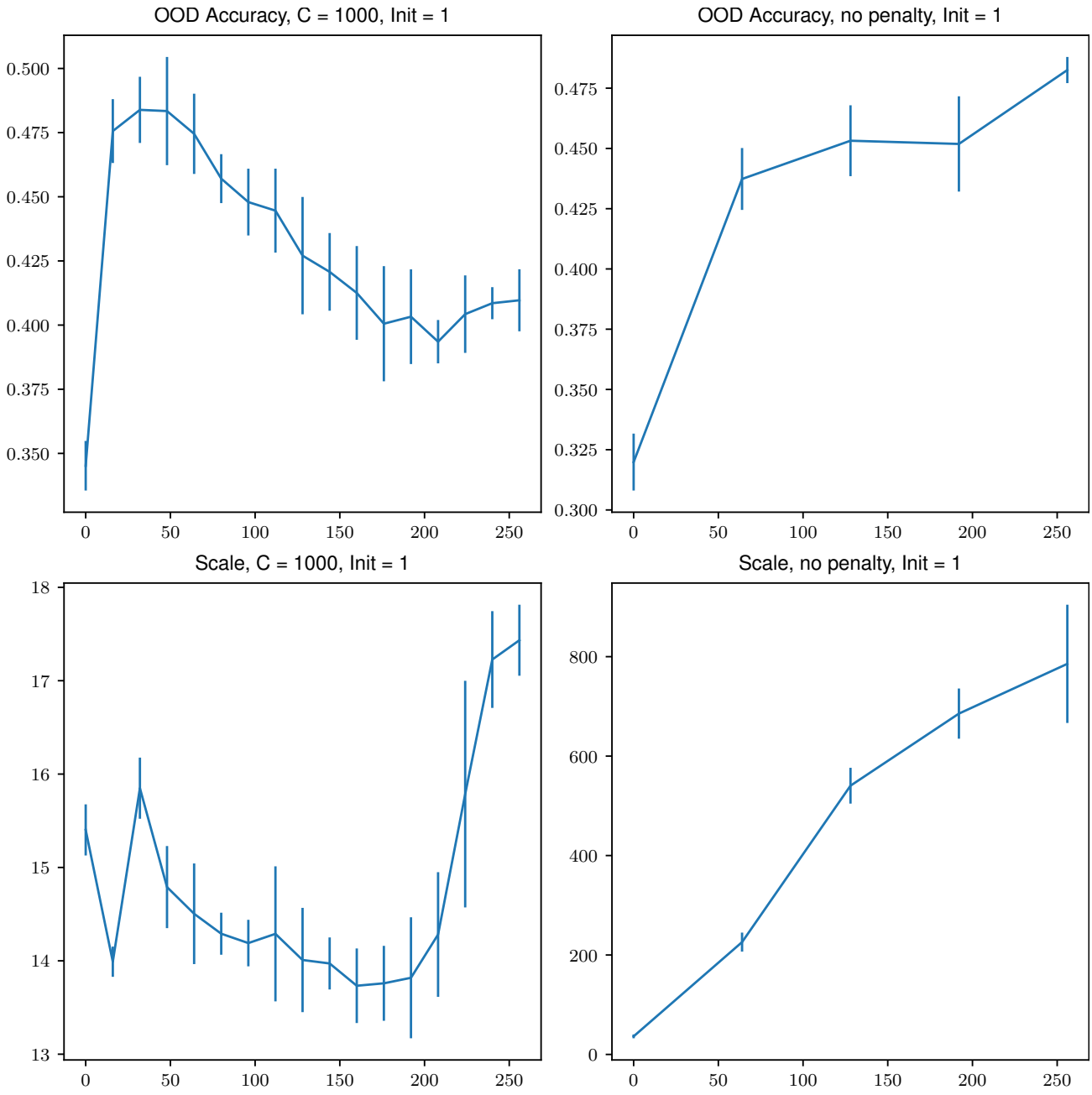

Figure 9. Accuracy and scale of the logistic regression on the validation part of the OOD test set (y-axis) vs. the training epoch at which the ResNet features are extracted (x-axis).

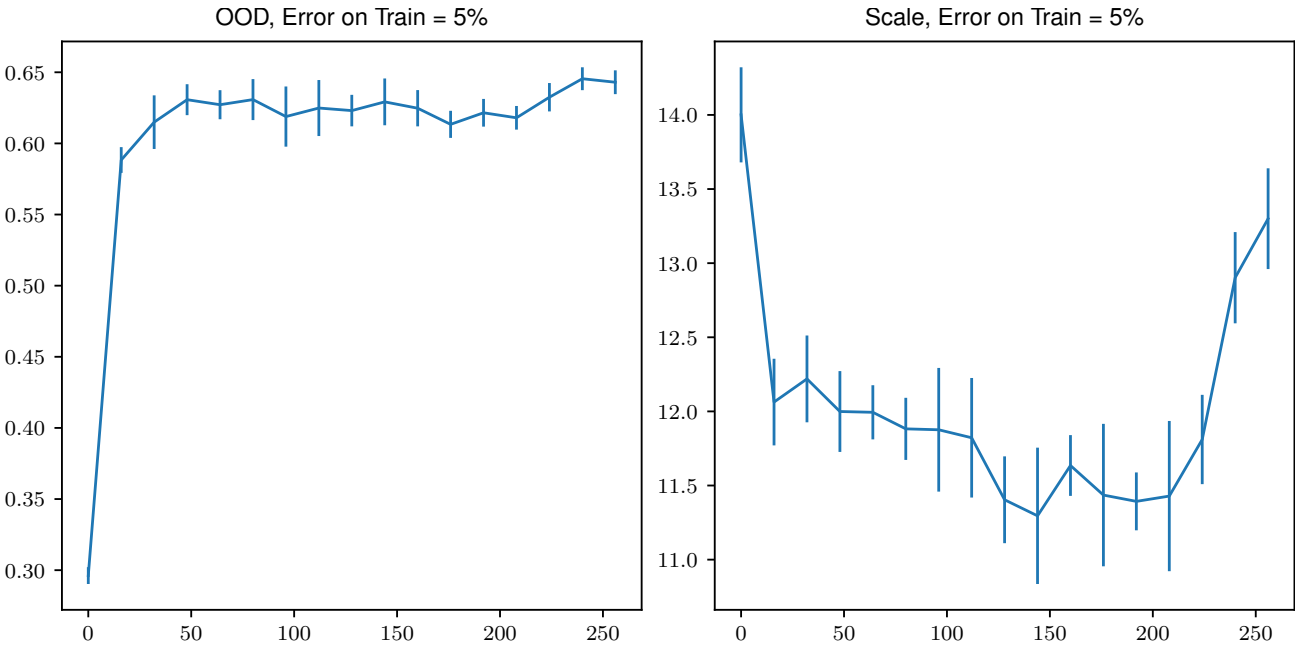

Figure 10. Accuracy and scale of the logistic regression on the validation part of the OOD test set (y-axis) vs. the training epoch at which the ResNet features are extracted (x-axis).