IMPLICIT LOW-RANK TRAINING DYNAMICS IN DIRECTLY PARAMETERIZED LIPSCHITZ-BOUNDED DEEP NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work has dedicated efforts to developing deep networks with prescribed Lipschitz bounds, in order to limit sensitivity to noisy or adversarial input perturbations. These networks enforce prescribed Lipschitz bounds through built-in structural constraints, typically derived from linear matrix inequalities (LMIs)/ semidefinite programs (SDPs), or imposed through orthogonality constraints on network layers. While these constraints provide certification, they can also make training less efficient. Empirical evidence shows that the training dynamics of large-scale networks often remain in low-dimensional subspaces and that certain initializations can lead to more effective training. Inspired by recent work on emergent low-rank training dynamics of multi-layer perceptrons (MLPs) under gradient descent (GD), we analyze the training dynamics of an existing 1-Lipschitz layer derived from an SDP certification, known as the "Sandwich layer", and show that the training dynamics of its effective weight concentrate within a low-dimensional subspace under GD. We first theoretically characterize this low-dimensional subspace in a shallow network with smooth, slope-restricted nonlinear activations, and empirically demonstrate that similar low-dimensional structure persists in deep networks under more general settings. Building on these observations, we develop a low-rank Sandwich layer that preserves its Lipschitz guarantees. This low-rank parameterization matches the performance of the fully parameterized Sandwich layer across multiple image classification tasks.

1 INTRODUCTION

The robustness of a deep neural network is important in safety critical applications. It has been shown that deep neural networks are sensitive to adversarial attacks (Szegedy et al., 2014), and many empirical defense methods have been developed to improve robustness (Madry et al., 2018; Shafahi et al., 2019; Wang et al., 2023; Kundu et al., 2021); nevertheless, most methods lack formal guarantees and may fail under sophisticated, carefully designed adversarial attacks (Tramer et al., 2020). In contrast, certified methods (also referred to as robustness verification) provide robustness guarantees. These are generally categorized into probabilistic and deterministic methods (Li et al., 2023). The most notable probabilistic method is randomized smoothing, which provides a certified radius by statistically estimating class probabilities (Cohen et al., 2019; Lecuyer et al., 2019; Yang et al., 2020). Conversely, deterministic methods bound the worst-case behavior of the network over a perturbation set. Common methods include branch-and-bound (Bunel et al., 2020), CROWN-style bound propagation (Zhang et al., 2018), and interval bound propagation (Gowal et al., 2018); however, their computational cost and bound tightness degrade for larger networks. Our work focuses on a deterministic method with built-in Lipschitz constraints, which allow us to compute the lower bound of worst-case robust accuracy and are commonly referred to as Lipschitz neural networks.

Lipschitz neural networks explicitly limit network sensitivity to adversarial inputs, typically by composing layers with prescribed Lipschitz bounds (Tsuzuku et al., 2018). Related work enforces Lipschitz constraints through spectral normalization (Miyato et al., 2018; Farnia et al., 2018), or constructs orthogonal/almost-orthogonal operators to obtain tight Lipschitz bounds (Li et al., 2019; Trockman & Kolter, 2021; Singla & Feizi, 2021; Xu et al., 2022; Prach & Lampert, 2022; Lai et al., 2025). More expressive methods have been developed using dynamical system tools Meunier et al.

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

(2022) and semi-definite programs (SDP) / linear matrix inequalities (LMI) (Fazlyab et al., 2019; Araujo et al., 2023; Pauli et al., 2026); we use one such method (Wang & Manchester, 2023) as a starting point for our work. Recent work scaled up the model to billion-parameter and achieved impressive certified robust accuracy (Hu et al., 2026). However, as Lipschitz networks impose structural constraints on weights, how training dynamics interact with those constraints remains comparatively unexplored.

Empirical observations demonstrate that for standard (i.e., non-Lipschitz) neural networks, training dynamics and large network fine-tuning is contained within low-dimensional subspaces (Gur-Ari et al., 2018; Li et al., 2018; Larsen et al., 2022); this is evidenced by the success of low-rank approaches in training (Balzano et al., 2026) and fine-tuning large neural networks (Hu et al., 2022). (Yaras et al., 2023) and (Kwon et al., 2024) prove that deep *linear* network training dynamics update within initialization-dependent fixed subspaces under gradient descent (GD), while (Xu et al., 2026) extends to multi-layer perceptrons (MLPs) with smooth *nonlinear* activations, which inspired this work. Also, related works reveal low-rank gradients in nonlinear networks (Ba et al., 2022; Zhao et al., 2024; Jaiswal et al., 2025; Sonthalia et al., 2025).

Motivated by recent observations of low-dimensional training dynamics, we study the training dynamics of a 1-Lipschitz layer called the "Sandwich layer", derived from an SDP in (Wang & Manchester, 2023). These dynamics are amenable to theoretical analysis, as they have a clean Cayley structure which yield the desired Lipschitz property without requiring constraints. This then leads to the following contributions:

1. We provide analytical results on the gradient concentration of Cayley transformation variables in the Sandwich layer with smooth, slope-restricted activation functions under GD. We show that the updates of those variables are concentrated in a fixed subspace that depends on the network initialization.
2. Empirical analysis on deep Sandwich networks. We empirically show that deep Sandwich networks also exhibit the aforementioned gradient concentration properties under SGD with momentum.
3. Low-rank parameterization of the Sandwich layer. We leverage the above gradient concentration properties to propose a low-rank parameterization for the Sandwich layer. We show that its performance (both in terms of accuracy and AutoAttack robustness) matches that of the dense Sandwich layer on a variety of classification tasks.

Notations. Let $\mathbb{R}$ be the set of real numbers. $A \succeq 0$ means that a square matrix A is positive semi-definite. $\mathbb{D}^n_{++}$ denotes the set of $n \times n$ positive diagonal matrices. For a matrix A, we use $\|A\|_1$, $\|A\|_2$, $\|A\|_F$, $\|A\|_\infty$, and $\|A\|_{\max}$ denote its 1-norm, spectral, Frobenius, infinity norm, and maximum magnitude entry, respectively. We write $s_i(A)$ for the i-th largest singular value of A, $\mathcal{R}(A)$ for the range of A, and $\mathcal{R}^\perp(A)$ for its orthogonal complement. We use $\odot$ for the Hadamard product. For some $N \in \mathbb{N}$, $[N]$ denotes the set $\{1, 2, \ldots, N\}$. We use $\mathrm{orth}(A)$ denotes any matrix with orthonormal columns spanning $\mathcal{R}(A)$.

2 Problem Setup and Preliminaries

Consider the following L-layer feedforward network:

$$\begin{aligned} z_0 &= x, \\ z_{k+1} &= \sigma(W_k z_k + b_k), \quad k = 0, \ldots, L-1 \\ y &= W_L z_L + b_L, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_0}, z_k \in \mathbb{R}^{n_k}, y \in \mathbb{R}^{n_{L+1}}$ are the inputs, k-th hidden layer and output, respectively. $W_k \in \mathbb{R}^{n_{k+1} \times n_k}$ and $b_k \in \mathbb{R}^{n_{k+1}}$ are the weight matrix and bias vector for the k-th layer, and σ is the element-wise activation function. Thus, the neural network defines a function $f_\phi : \mathbb{R}^{n_0} \to \mathbb{R}^{n_{L+1}}$ with $f_\phi(x) = y$, where ϕ collects all trainable parameters (i.e., weights and biases). We say that a function f_ϕ is *Lipschitz-bounded* if there exists $\beta > 0$ such that for all x_1, x_2 in its domain,

$$\|f_\phi(x_1) - f_\phi(x_2)\| \leq \beta \|x_1 - x_2\| \tag{2}$$

The *Lipschitz constant* of a Lipschitz-bounded function, denoted as $\mathrm{Lip}(f_\phi)$, is the smallest value of β such that equation 2 holds. We call f_ϕ a γ-Lipschitz function if $\mathrm{Lip}(f_\phi) \leq \gamma$.

A Lipschitz neural network is a network of the form 1 trained to minimize some loss function $\mathcal{L}(\cdot)$ while preserving γ-Lipschitz, i.e.,

$$\min_{\phi} \mathcal{L}(f_\phi) \quad \text{s.t.} \quad \text{Lip}(f_\phi) \leq \gamma \tag{3}$$

We now briefly review the Lipschitz neural network method from Wang & Manchester (2023), which is the subject of our analysis; this starts with the following standing assumption (which will apply throughout this paper):

Assumption 2.1. The nonlinear activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is slope-restricted in $[0, 1]$.

Under this assumption, a sufficient condition for $\text{Lip}(f_\phi) \leq \gamma$, via Fazlyab et al. (2019), is:

$$H := \begin{bmatrix} \gamma I & -U^\top \Lambda & 0 \\ -\Lambda U & 2\Lambda - \Lambda W - W^\top \Lambda & -Y^\top \\ 0 & -Y & \gamma I \end{bmatrix} \succeq 0 \tag{4}$$

where $\Lambda \in \mathbb{D}^n_{++}$ and $n = \sum_{k=1}^{L} n_k$. W, U, and Y collect the intermediate, input, and output weight matrices, respectively (details in Appendix A.1). This constraint can be enforced during training, or parameterized to facilitate unconstrained training (in new parameters). In Wang & Manchester (2023), instead of directly using networks of the form 1, the so-called Sandwich layer is proposed:

$$h_{k+1} = \sqrt{2} A_k^\top \Psi_k \sigma\left(\sqrt{2}\Psi_k^{-1} B_k h_k + b_k\right). \tag{5a}$$

$$\begin{bmatrix} A_k^\top \\ B_k^\top \end{bmatrix} = \text{Cayley}\left(\begin{bmatrix} X_k \\ Y_k \end{bmatrix}\right) := \begin{bmatrix} (I + Z_k)^{-1}(I - Z_k) \\ -2Y_k(I + Z_k)^{-1} \end{bmatrix}, \quad Z_k := X_k - X_k^\top + Y_k^\top Y_k, \tag{5b}$$

where trainable parameters A_k and B_k are parameterized by unconstrained parameters X_k and Y_k through the Cayley transformation. Throughout this work, we use the simplified setting $\Psi_k = I$ and $b_k = 0$ for tractability and to isolate the low-rank training dynamics, though this restriction may modestly reduce empirical performance. By construction, layer 5a is 1-Lipschitz. Then, the overall γ-Lipschitz neural network composed of stacked Sandwich layers is

$$h_0 = \sqrt{\gamma} x, \quad h_{k+1} = \sqrt{2} A_k^\top \sigma\left(\sqrt{2} B_k h_k\right) \text{ for } 1 \leq k < L, \quad y = \sqrt{\gamma} B_L h_{L-1}, \tag{6}$$

where the Lipschitz constant of the overall network is the product of the Lipschitz constants of individual layers; in this case, hidden layers are all 1-Lipschitz. For our analysis, we allow arbitrary distribution of the Lipschitz constant across layers. We let g_k denote the per-layer Lipschitz constant, and require $\prod_{k=0}^{L} g_k = \gamma$. To facilitate analysis of unconstrained parameters X_k and Y_k, we define $S_k := X_k - X_k^\top$ and $C_k := \left(I + S_k + Y_k^\top Y_k\right)^{-1}$. Then, we can rewrite network 6 as

$$h_0 = x, h_{k+1} = \sqrt{2}(2C_k - I)\sigma\left(-2\sqrt{2} g_k C_k^\top Y_k^\top h_k\right), \text{for } 1 \leq k < L, y = -2 g_L C_L^\top Y_L^\top h_L. \tag{7}$$

This γ-Lipschitz network is the one we will analyze and provide low-rank parameterizations for. We denote it as $f_\Theta : \mathbb{R}^d \to \mathbb{R}^K$, where $\Theta := \{(X_k, Y_k)\}_{1 \leq k \leq L}$ collects trainable parameters. We assume a uniform hidden width m for notation simplicity, with $h_0 = x \in \mathbb{R}^d$ and $h_k \in \mathbb{R}^m$. Thus, $Y_1 \in \mathbb{R}^{d \times m}, Y_k \in \mathbb{R}^{m \times m}, Y_L \in \mathbb{R}^{m \times K}$ with $X_k, S_k, C_k \in \mathbb{R}^{m \times m}$ and $X_L, S_L, C_L \in \mathbb{R}^{K \times K}$. In terms of data, we let $\Xi \in \mathbb{R}^{d \times N}$ denote inputs and $T \in \mathbb{R}^{K \times N}$ denote labels, where d is the data dimension, N is the number of samples, and K is the label dimension. The network f_Θ is trained to solve 3, with the trainable parameters specialized to the Sandwich parameters Θ. The set of parameters is updated through GD with step size η, and t refers to the iteration number: $\Theta(t+1) = \Theta(t) - \eta \nabla \mathcal{L}\big(\Theta(t)\big)$.

3 Analysis of low-rank training dynamics in Sandwich network

We first provide a motivating example which shows that the training dynamics of Y_k in a multi-layer Sandwich network exhibits low-rank structure. We train a 4-layer Sandwich network with input and hidden dimensions $d = m = 32$ on a synthetic classification task with class $K = 4$ (details in A.2) with a variety of activation functions (Tanh, CELU, ELU, ReLU). We record singular values along with top-K, bottom-K, and middle $d - 2K$ singular subspaces of Y_1 and Y_2 (i.e., the first

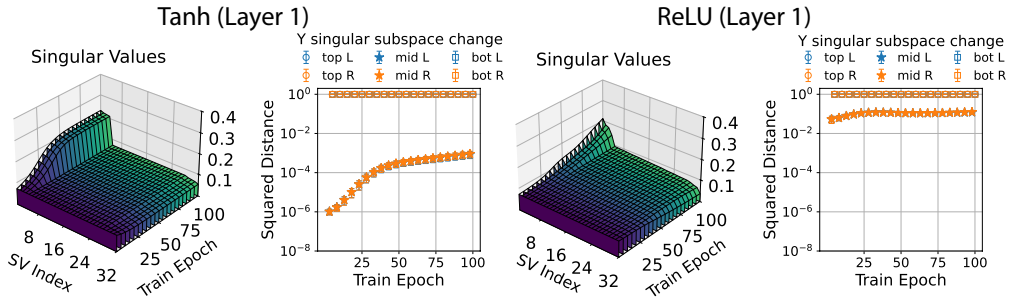


Figure 1: The middle singular subspace change of first layer Y_1 with the smooth activation is significantly smaller than the non-smooth one, suggesting that singular values change mostly happen in certain direction. Combining the singular-value and subspace-change plots, we observe that the change of Y_1 is concentrated along certain directions, with this concentration being much stronger for smooth activation functions.

2 layers); Fig. 1 shows these quantities. The middle singular subspace of Tanh activation changes much slower compared to ReLU activation in the first layer. Similar trends also exist in the second layer and across smoother functions (more ablation study see A.2).

This suggests the low-rank training dynamics of MLP discussed in (Xu et al., 2026) also exist in the Sandwich network, which has much stronger built-in constraints. And low-rank training dynamics are stronger in smooth activation functions, consistent with previous MLP results. Therefore, in this section, we focus on theoretically characterizing these low-rank training dynamics in the Sandwich network with smooth slope-restricted activation functions. We now state the important assumptions.

We follow previous work to make a standard assumption the input data is whitened (Xu et al., 2026; Yaras et al., 2023; Arora et al., 2018; Braun et al., 2022; Domine et al., 2025). For the low-rank training dynamics, we assume that the output dimension (i.e., the number of classes) is much smaller than the input dimension, $K \ll d$, as is common in classification settings (Papyan et al., 2020). The output dimension K determines the rank of the gradient-concentrated subspaces.

3.1 Two layer Sandwich network theoretical analysis

Assumption 3.1 (Input data)**.** The input data $\Xi \in \mathbb{R}^{d\times N}$ is whitened as $\Xi\Xi^\top = vI_d$, and the label $T \in \mathbb{R}^{K\times N}$ with $K < d/2$.

We discuss the two-layer Sandwich network for analytical analysis in this section. The goal is to show that $Y_1 \in \mathbb{R}^{d\times m}$ is updated in a low-dimensional subspaces. Following previous work, we freeze the second layer to simplify the analysis (Xu et al., 2026; Oymak & Soltanolkotabi, 2019; 2020; Frei et al., 2022; Kou et al., 2023; Gopalani & Mukherjee, 2025). We freeze X_1, X_2 and focus on Y_1, since only the skew-symmetric part of X_1, X_2 enters the network.

Assumption 3.2 (Two-layer Sandwich network)**.** We consider the network 7 with $L = 2$, consisting of a single Sandwich layer followed by a *frozen* linear readout. Throughout training, Y_1 is the only trainable parameter, whereas X_1, X_2, and the readout parameters are held fixed.

$$S_1 := X_1 - X_1^\top = 0, \quad C_1 := \left(I + S_1 + Y_1^\top Y_1\right)^{-1} = \left(I + Y_1^\top Y_1\right)^{-1} \in \mathbb{R}^{m\times m}, \tag{8a}$$

$$S_2 := X_2 - X_2^\top = 0, \quad C_2 := \left(I + S_2 + Y_2^\top Y_2\right)^{-1} = \left(I + Y_2(0)^\top Y_2(0)\right)^{-1} \in \mathbb{R}^{m\times m}. \tag{8b}$$

The readout is initialized from $Y_2(0)$ and then frozen: $W_2(0) := -2g_2C_2Y_2(0)^\top,\ W_2(t) \equiv W_2(0)$.

In this work, we focus on the squared-error loss, which is also commonly discussed in prior work (Oymak & Soltanolkotabi, 2019; 2020; Bao et al., 2023; Chistikov et al., 2023; Zhou et al., 2022; Wang et al., 2025). We next state important assumptions for theoratical analysis.

Assumption 3.3 (Activation, loss and training)**.** The following conditions hold:

- **Activation.** The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is twice differentiable and satisfies

$$\sigma(0) = 0, \qquad a := \sigma'(0) > 0, \qquad 0 \le \sigma'(z) \le 1, \qquad |\sigma''(z)| \le \mu, \quad \forall z \in \mathbb{R}.$$

• **Loss.** Given training data (Ξ, T) with N samples, the objective is average squared error

$$\mathcal{L}(Y_1) := \frac{1}{2N} \left\| f_{Y_1}(\Xi) - T \right\|_F^2 .$$

• **Gradient descent.** The parameter Y_1 is updated via GD with step size η and define gradient on Y_1 as $G_1(t) := \nabla_{Y_1} \mathcal{L}(Y_1(t))$.

Assumption 3.4 (Training trajectory)**.** For a fixed horizon H, there exist constants $c_1, c_2 \geq 1$, $c > 0$, and $c_3 \in (0, 1]$, such that, for all $0 \leq t < H$,

$$\frac{\|G_1(t)\|_F}{\|G_1(0)\|_F} \leq c_1 \lambda^t, \quad \lambda := e^{-c\eta} \tag{9}$$

$$\frac{s_i(G_1(t))}{s_i(G_1(0))} \leq c_2 \frac{\|G_1(t)\|_F}{\|G_1(0)\|_F}, \quad i = 1, \ldots, K, \tag{10}$$

$$s_K(M) - s_{K+1}(G_1(t)) \geq c_3 s_K(M), \quad M := \frac{4ag_1}{N} \Xi T^\top W_2. \tag{11}$$

The above assumption is introduced for technical convenience. As in (Xu et al., 2026), it comes strict due to the non-convexity of the objective; however, we empirically validate that the gradient norm and singular values decrease at comparable rates under GD (see Appendix Fig. 13).

Inequality 11 assumes that the spectral tail of $G_1(t)$ is much smaller than $s_K(\frac{4ag_1}{N} \Xi T^\top W_2)$, which allows us to use Wedin's Sin Theorem (Wedin, 1972) to analyze singular subspace alignment. Unlike (Xu et al., 2026), we do not separately assume $\|f_{Y_1}(\Xi) - T\|_{\max}$ is bounded, since this follows from the Sandwich parameterization (See Appendix A.9.2).

We next introduce the metric to measure subspaces alignment and state the simplified main theorem.

Definition 3.5 (Principal angles between subspaces)**.** Let U_1, U_2 be the orthonormal basis of two r-dimensional subspaces of $\mathbb{R}^d$. Then the i-th principal angle θ_i between two bases is defined as

$$\theta_i = \arccos(s_i(U_1^\top U_2)) \in [0, \frac{\pi}{2}], \quad \text{for } i \in [r]. \tag{12}$$

Then, we use the following metric to measure bases alignment:

$$\left\| \sin \Phi \left(U_1, U_2 \right) \right\|_2 := \sqrt{\sum_{i=1}^{r} \sin^2(\theta_i)} \in [0, \sqrt{r}]. \tag{13}$$

Smaller principal angles θ_i lead to smaller $\left\| \sin \Phi \left(U_1, U_2 \right) \right\|_2$, implying two bases are well aligned. Next theorem shows the alignment between subspaces during training and how initialization affects the subspaces update under GD.

Theorem 3.6. *Assume assumptions 3.1, 3.2, 3.3 and 3.4 hold, let $L_{1,1}(t)$ and $R_{1,1}(t)$ denote the top-K left and right singular subspaces of $G_1(t)$ and let $p := d - 2K > 0$. Define*

$$A(t) := \max \left\{ \left\| \sin \Phi(L_{1,1}(t), L_{1,1}(0)) \right\|_2 , \left\| \sin \Phi(R_{1,1}(t), R_{1,1}(0)) \right\|_2 \right\}. \tag{14}$$

Suppose $Y_1(0) = \rho Q,\ Q \in \mathbb{R}^{d \times m},\ QQ^\top = I_d$ with sufficiently small ρ and learning rate η, there exist fixed orthogonal matrices $L \in \mathbb{R}^{d \times d},\ R \in \mathbb{R}^{m \times m}$ which only depends on $Y_1(0)$ and $G_1(0)$, such that, for $0 \leq t < H$:

$$Y_1(t) = L \widetilde{Y}_1(t) R^\top = L \begin{bmatrix} \widetilde{Y}_{11}(t) & \widetilde{Y}_{12}(t) \\ \widetilde{Y}_{13}(t) & \widetilde{Y}_{14}(t) \end{bmatrix} R^\top. \tag{15}$$

where $\widetilde{Y}_{1,1}(t) \in \mathbb{R}^{2K \times (m-p)},\ \widetilde{Y}_{1,2}(t) \in \mathbb{R}^{2K \times p},\ \widetilde{Y}_{1,3}(t) \in \mathbb{R}^{p \times (m-p)},\ \widetilde{Y}_{1,4}(t) \in \mathbb{R}^{p \times p}$ with $\widetilde{Y}_{1,2}(0) = 0,\ \widetilde{Y}_{1,3}(0) = 0,\ \widetilde{Y}_{1,4}(0) = \rho I_p$. Moreover, for $j \in \{2, 3, 4\}$,

$$\frac{\|\widetilde{Y}_{1,j}(t+1) - \widetilde{Y}_{1,j}(t)\|_F}{\sqrt{p}} \leq \eta\, e(t), \quad e(t) := \sqrt{s_1^2(G_1(t)) \frac{A^2(t)}{p} + s_{K+1}^2(G_1(t))}. \tag{16}$$

where

$$e(0) = O(\rho), \quad e(t) \lesssim \frac{\lambda^t(1 - \lambda^t)}{\sqrt{p}} + s_{K+1}(G_1(t)), \quad 1 \leq t < H \tag{17}$$

Proof sketch. First, we show that the initial gradient is approximately rank-K. Under the small row-orthogonal initialization $Y_1(0) = \rho Q$, the smoothness of the activation function implies $\sigma'(Z_1(0)) = \sigma'(0)\mathbf{1} + O(\rho)$. Derive the analytical gradient on Y_1, we have:

$$\|G_1(0) - M\|_2 \le r_\rho, \quad M := \frac{4ag_1}{N}\Xi T^\top W_2, \quad r_\rho = O(\rho), \tag{18}$$

where $\mathrm{rank}(M) \le K$ and the singular value of $G_1(0)$ is approximately rank-K.

Second, similar to (Yaras et al., 2023; 2024; Xu et al., 2026), we identify fixed left and right subspaces onto which the gradient has small components. Let $L_{1,2}(0)$ denote the bottom $d-K$ left singular subspace and $R_{1,1}(0)$ the top-K right singular subspace of $G_1(0)$. We define:

$$\mathcal{S} := \mathcal{R}(L_{1,2}(0)) \cap \mathcal{R}^\perp(Y_1(0)R_{1,1}(0)/\rho) = \mathcal{R}(L_{1,2}(0)) \cap \mathcal{R}^\perp(QR_{1,1}(0)) \tag{19}$$

The two components make the left and paired right directions orthogonal to the initial dominant gradient subspaces under $Y_1(0)$ initialization. Choosing an orthonormal basis $L_s \in \mathbb{R}^{d\times p}$ from $\mathcal{S}$ and setting $R_s := Q^\top L_s$ gives a pair of fixed left-right directions that avoid the dominant initial gradient subspaces. We then show that $\|L_s^\top G_1(t)\|_F$ and $\|G_1(t)R_s\|_F$ remain small. Completing L_s and R_s to fixed orthogonal bases then yields the block decomposition in Theorem 3.6. See Appendix A.9 for details.

Theorem 3.6 shows that **low-rank training dynamics emerge for Y_1 when the fixed subspaces are chosen based on initial gradient**, and such subspaces remain well aligned in early training phase. The leakage bound $e(t)$ has two sources: rotation of the dominant gradient singular subspaces and the spectral tail $s_{K+1}(G_1(t))$. Although, the dominant subspaces rotate during training, their singular values decay. This is reflected in $\lambda^t(1-\lambda^t)$ term and is empirically validated (see Appendix Fig. 13). **Small initialization helps control the growth of leakage**. Leakage at initialization is $\|\widetilde{Y}_{1,4}(0)\|_F/\sqrt{p} = \rho$, and satisfies $e(0) = O(\rho)$. Therefore, smaller initialization produces smaller leakage and initial updates outside $\widetilde{Y}_{1,1}(0)$, implying more concentrated training dynamics. Similar phenomenon is observed in (Frei et al., 2022; Kou et al., 2023),and discussed in (Xu et al., 2026). The complete theorem can be found in Appendix A.9.

Under Assumptions 3.1, 3.2, 3.3, and 3.4, the updates of the two-layer Sandwich network concentrate in $\widetilde{Y}_{1,1}(t)$ as shown in Fig. 2. The y-axis plots the square distance from initialization normalized by the total update $\|\widetilde{Y}_{1,j}(t) - \widetilde{Y}_{1,j}(0)\|_F^2/\|\widetilde{Y}_1(t) - \widetilde{Y}_1(0)\|_F^2$. See details in Appendix A.3.

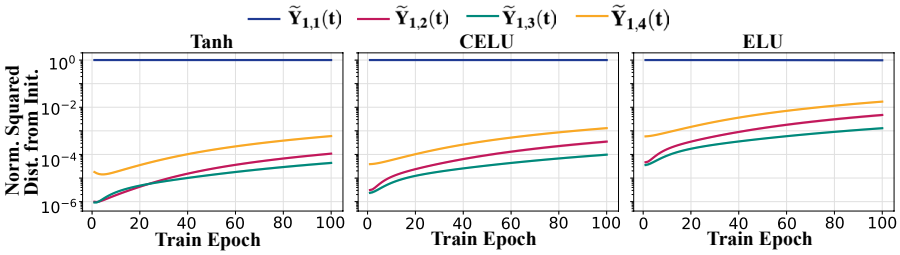


Figure 2: The updates of $Y_1(t)$ concentrate in the subspace $\widetilde{Y}_{1,1}(t)$ under theoretical settings across different activation functions

3.2 Empirical observations for deep Sandwich networks

In this section, we further examine the low-dimensional training dynamics in deeper Sandwich networks. We consider a network with three hidden Sandwich layers followed by a dense readout. We train all $\Theta := \{(X_k, Y_k)\}_{1\le k\le L}$ with full batch GD. For each layer, let $G_k(0) = \nabla_{Y_k}\mathcal{L}(\Theta(0))$ and let $L_{k,1}(0)$ and $R_{k,1}(0)$ denote its top-K left and right singular subspaces. Follow the two-layer analysis, we define the gradient concentration subspaces for each layer as:

$$\bar{L}_k = \mathrm{orth}[L_{k,1}(0), Y_k(0)R_{k,1}(0)/\rho_k], \quad \bar{R}_k = \mathrm{orth}\left[R_{k,1}(0), Y_k(0)^\top L_{k,1}(0)/\rho_k\right], \tag{20}$$

where $\bar{L}_k \in \mathbb{R}^{m\times 2K}$, $\bar{R}_k \in \mathbb{R}^{m\times 2K}$. Let $\bar{L}_{k,\perp}$ and $\bar{R}_{k,\perp}$ denote their orthogonal complement, then for each hidden layer we have the following block decomposition:

$$Y_k(t) = \begin{bmatrix} \bar{L}_k^\top \\ \bar{L}_{k,\perp}^\top \end{bmatrix}^\top \begin{bmatrix} \widetilde{Y}_{k,1}(t) & \widetilde{Y}_{k,2}(t) \\ \widetilde{Y}_{k,3}(t) & \widetilde{Y}_{k,4}(t) \end{bmatrix} \begin{bmatrix} \bar{R}_k^\top \\ \bar{R}_{k,\perp}^\top \end{bmatrix}. \tag{21}$$

For each hidden layer k, the y-axis shows the normalized squared distance of each block from initialization, $\|\widetilde{Y}_{k,j}(t)-\widetilde{Y}_{k,j}(0)\|_F^2/\|\widetilde{Y}_k(t)-\widetilde{Y}_k(0)\|_F^2$, for $j \in \{1,2,3,4\}$. In Fig. 3, we empirically observe a similar low-rank training dynamics across all hidden layers for both tanh and CELU activations. We provide additional ablations in Appendix Fig. 8, where we train deep Sandwich networks using SGD with momentum, Adam, and Muon, and also consider unwhitened data. Across these settings, we consistently observe a similar low-rank training dynamics.

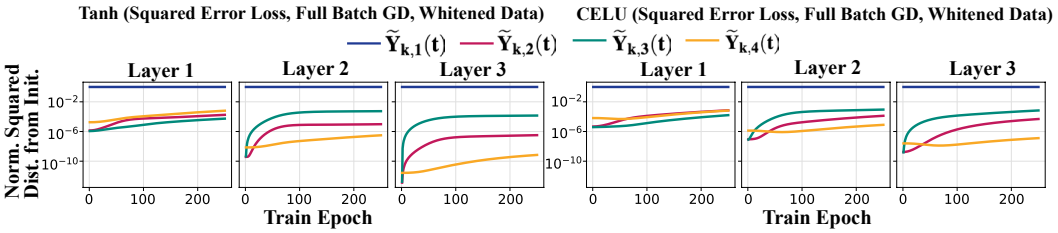


Figure 3: The updates of each hidden layer Y_k concentrate in the subspace $\widetilde{Y}_{k,1}(t)$. Empirically, the same low-rank training dynamics of Y_k exist in deep Sandwich network across tanh and CELU.

4 Low-rank parameterization of Sandwich networks

Building upon the two-layer Sandwich network analysis and empirical observations in deep Sandwich networks, we propose a low-rank parameterization for Sandwich networks that allows us to use fewer trainable parameters to match the dense counterpart while keeping the exact Lipschitz certification. For the low-rank Sandwich network, we parameterize each hidden layer as

$$Y_k = L_k M_k R_k^\top, \quad X_k - X_k^\top = R_k \Omega_k R_k^\top, \tag{22}$$

where $L_k \in \mathbb{R}^{m\times r}$ and $R_k \in \mathbb{R}^{m\times r}$ are semi-orthogonal matrices, while $M_k, \Omega_k \in \mathbb{R}^{r\times r}$ are trainable low-dimensional cores with $\Omega_k^\top = -\Omega_k$. Recall the Sandwich hidden layer in 7, the choice of using the same right subspace R_k for the skew coordinate is directly motivated by the Sandwich structure. Since $Y_k^\top Y_k = R_k M_k^\top L_k^\top L_k M_k R_k^\top$, choose $S_k = X_k - X_k^\top = R_k \Omega_k R_k^\top$ yields

$$X_k - X_k^\top + Y_k^\top Y_k = R_k P_k R_k^\top, \quad P_k := \Omega_k + M_k^\top L_k^\top L_k M_k \in \mathbb{R}^{r\times r}, \tag{23}$$

and therefore $C_k = \left(I + S_k + Y_k^\top Y_k\right)^{-1} = \left(I + R_k P_k R_k^\top\right)^{-1}$. Consequently,

$$C_k^\top Y_k^\top h_k = R_k \left(I + P_k^\top R_k^\top R_k\right)^{-1} M_k^\top L_k^\top h_k, \tag{24}$$

$$2C_k - I = I - 2R_k \left(I + P_k R_k^\top R_k\right)^{-1} P_k R_k^\top. \tag{25}$$

Thus, each Sandwich hidden layer is confined to the r-dimensional subspace $\mathcal{R}(R_k)$, reducing the required matrix inversion to an $r \times r$ system. To connect the low-dimensional coordinates across layers and avoid enforcing orthogonality during training, we choose a semi-orthogonal interface subspace $F_k \in \mathbb{R}^{m\times r}$ and then freeze,

$$F_k := R_k = L_{k+1}, \quad F_k^\top F_k = I_r, \quad \text{for } k = 1, \ldots, L-2. \tag{26}$$

This gives a common r-dimensional coordinate system for adjacent layers and avoids redundant degrees of freedom. We keep the internal shared subspaces fixed and semi-orthogonal, while the two endpoint subspaces L_1 and R_{L-1} remain trainable. Since these endpoint subspaces are not shared across hidden-layer interfaces, they need not preserve a common orthonormal coordinate

system during training. The readout remains a fully trainable dense Sandwich layer as form in 7 and does not share any interface subspace with the hidden layers.

Initialization: To initialize two endpoint subspaces $L_1(0) \in \mathbb{R}^{d\times r}$ and $R_{L-1}(0) \in \mathbb{R}^{m\times r}$, we first construct a full-rank network with $Y_k^{\text{full}}(0) = \rho_k Q_k,\ Q_k \in \mathbb{R}^{m\times m}$ and compute the initial gradient of the first hidden layer $G_1(0) = \nabla_{Y_1}\mathcal{L}\big|_{t=0}$. Let $L_{1,1}(0) \in \mathbb{R}^{d\times K}$ and $R_{1,1}(0) \in \mathbb{R}^{m\times K}$ denote top-K singular subspaces of $G_1(0)$. For $r = 2K$, we initialize endpoint subspace $L_1(0)$ as

$$L_1(0) = \text{orth}\left[L_{1,1}(0),\, Q_1 R_{1,1}(0)\right] \in \mathbb{R}^{d\times r}. \tag{27}$$

We then propagate this subspace through the full rank hidden layers to initialize $R_{L-1}(0)$ via

$$F_1^{\text{full}} := L_1(0) \in \mathbb{R}^{d\times r}, \tag{28a}$$

$$F_{k+1}^{\text{full}} := \text{orth}\big(Q_k^\top F_k\big) \in \mathbb{R}^{m\times r}, \quad k = 1, \ldots, L-1, \tag{28b}$$

$$R_{L-1}(0) := F_L^{\text{full}} \in \mathbb{R}^{m\times r}. \tag{28c}$$

The internal low-rank interface subspaces are initialized according to equation 26. For each hidden layer, we initialize the core as $M_k(0) = \rho_k \widehat{Q}_k$, where $\widehat{Q}_k \in \mathbb{R}^{r\times r}$ is orthogonal, and set the skew core to $\Omega_k(0) = 0$ for $k = 1, \ldots, L-1$. The dense readout is initialized independently as $Y_L(0) = \rho_L Q_L,\ Q_L \in \mathbb{R}^{m\times K},\ Q_L^\top Q_L = I_K$.

5 Experiments

In this section, we show that when the endpoint subspaces $L_1(0)$ and $R_{L-1}(0)$ are initialized according to 28, our **low-rank Sandwich networks can achieve nearly the same performance, in terms of test accuracy and test loss, as their dense counterparts using fewer trainable parameters while preserving the Lipschitz certification**. We use the same hyperparameters across all model variants. All experiments are implemented in PyTorch and run on NVIDIA A40 GPUs.

We first train a low-rank sandwich network and a full-rank one on Fashion MNIST (Xiao et al., 2017). Fig. 4 shows that a low-rank model with proper initialization (41K paras) will closely match the dense model (2.77M paras) on test accuracy and test loss trajectory. The low-rank model with randomly initialized endpoint subspaces remains at much lower test accuracy, while its test loss stays high throughout training. Training details and ablation results are in Appendix A.5.

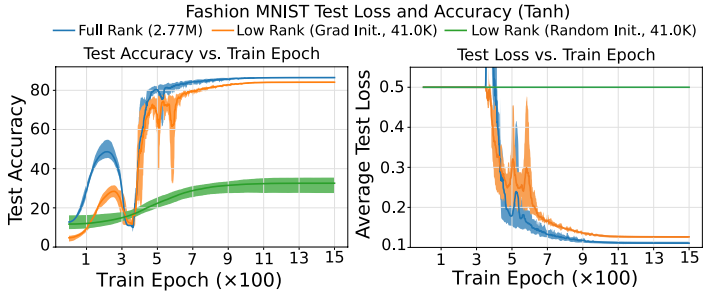


Figure 4: Comparison of the full-rank model, the low-rank model (gradient-informed init.), and the low-rank model (random init.) on Fashion MNIST. With gradient-informed init., the low-rank model closely matches the dense model in both test accuracy and test-loss trajectory while randomly init. low-rank model has much worse accuracy and high test loss.

We then test the low-rank sandwich network Lipschitz tightness on fitting a square wave using networks with Lipschitz bound of 10. Left panel of Fig. 5 shows empirical versus prescribed Lipschitz bounds, where γ_{lb} denotes the empirical bound estimated by projected gradient descent. Across hidden layers, the low-rank model matches the Lipschitz-bound tightness of the original Sandwich model, while effectively use the Lipschitz budget. Right panel shows the spectral norms of the effective weight matrices (see Appendix equation 47). The low-rank model exhibits behaviors similar to the original Sandwich model, allowing norms larger than 1, unlike methods that explicitly constrain weight norm at most 1.

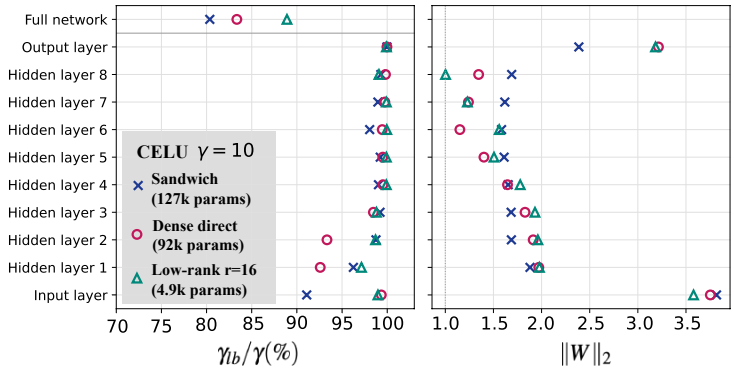


Figure 5: **Left**: empirical versus prescribed Lipschitz bounds, where γ_{lb} denotes the empirical bound estimated by projected gradient descent. The low-rank model shows similar Lipschitz-bound tightness. **Right**: spectral norms of the effective weight matrices (see Appendix equation 47). The low-rank Sandwich network allows norms larger than 1, unlike methods that overly constrain individual weight norm at most 1.

Last, we test the low-rank model on CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009). We use AutoAttack (Croce & Hein, 2020) to evaluate empirical robustness. For CIFAR-10, we use the pretrained CNN backbone from the released KWL-Sandwich-small model (Wang & Manchester, 2023) with Lipschitz bound $\gamma = 1$, and replace its classifier with our low-rank Sandwich classifier then freeze the backbone and only fine-tune the low-rank classifier. We compare the lo-rank classifier against the corresponding dense classifier. The results show that, with gradient-informed initialization, the low-rank model improves consistently as the rank increases, with higher-rank variants closely approaching the dense model in both clean accuracy and AutoAttack robustness. In contrast, randomly initialized low-rank models remain at much lower clean and robust accuracy. For CIFAR-100, we use the pretrained CNN backbone from the released KWL-Sandwich-large model with Lipschitz bound $\gamma = 2$. We observe the same overall trend as on CIFAR-10. See Appendix A.7 for rank ablation test accuracy and loss trajectory.

Table 1: CIFAR-10 and CIFAR-100 AutoAttack results with fine-tuned low-rank Sandwich classifier. Gradient-initialized low-rank networks match the performance of the dense network as the rank grows, while the randomly initialized model stays at low accuracy. * denotes random initialization.

CIFAR-10

Rank (Params)	Clean	AA@36	AA@72	AA@108
$r = 10$ (52K)	60.7	53.4	44.1	35.4
$r = 20$ (99K)	64.2	55.3	45.9	37.3
$r = 40$ (194K)	64.6	55.9	46.6	37.2
$r = 80$ (393K)	65.2	56.3	47.7	38.5
$r = 20^*$ (99K)	16.0	8.2	4.5	2.1
Dense (2.63M)	66.3	57.8	47.9	38.7

CIFAR-100

Rank (Params)	Clean	AA@36	AA@72	AA@108
$r = 200$ (4.02M)	52.8	43.9	35.7	28.7
$r = 400$ (8.06M)	54.2	45.3	36.7	30.8
$r = 800$ (16.87M)	54.8	45.8	37.9	30.6
$r = 200^*$ (4.02M)	5.6	2.9	1.3	0.8
Dense (42.15M)	55.4	46.1	37.4	30.8

6 Conclusion

We study the training dynamics of a Lipschitz-constrained deep neural network and showed that their training dynamics concentrate in a low-dimensional structure. We theoretically investigated that in a shallow network, the gradient updates for certain variables concentrated in a low-dimensional subspaces that depends on the initialization under gradient descent. We further observed that this gradient updates concentration persists empirically in deep sandwich networks under gradient descent. We then proposed a low-rank parameterization of the Sandwich network and evaluated across the classification tasks. The low-rank model closely match their dense counterparts in terms of test accuracy, loss trajectory and AutoAttack robustness. These results suggest that Lipschitz-guaranteed networks can be made more parameter-efficient without sacrificing their robustness guarantees.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the verification of mathematical arguments, notation design, figure formatting, hyperparameter search, numerical-stability improvements in code, and debugging. We also used generative AI tools to improve the clarity and readability of the manuscript. All AI-assisted content was reviewed and verified by the authors. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed network architectures, initialization procedures, training hyperparameters, and evaluation protocols in the main text and Appendix. Complete proofs of the theoretical results are provided in the Appendix.

REFERENCES

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International conference on machine learning*, pp. 291–301. PMLR, 2019.

Alexandre Araujo, Aaron J Havens, Blaise Delattre, Alexandre Allauzen, and Bin Hu. A unified algebraic perspective on lipschitz neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018.

Jimmy Ba, Murat A Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems*, 35:37932–37946, 2022.

Laura Balzano, Tianjiao Ding, Benjamin D. Haeffele, Soo Min Kwon, Qing Qu, Peng Wang, Zhangyang Wang, and Can Yaras. An overview of low-rank structures in the training and adaptation of large models: From implicit low-dimensionality to efficient training and fine-tuning [special issue on the mathematics of deep learning]. *IEEE Signal Processing Magazine*, 43(3): 52–70, 2026. doi: 10.1109/MSP.2026.3666749.

Yajie Bao, Amarda Shehu, and Mingrui Liu. Global convergence analysis of local sgd for two-layer neural network without overparameterization. *Advances in Neural Information Processing Systems*, 36:24610–24660, 2023.

Lukas Braun, Clémentine Dominé, James Fitzgerald, and Andrew Saxe. Exact learning dynamics of deep linear networks with prior knowledge. *Advances in Neural Information Processing Systems*, 35:6615–6629, 2022.

Rudy Bunel, Jingyue Lu, Ilker Turkaslan, Philip HS Torr, Pushmeet Kohli, and M Pawan Kumar. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21(42):1–39, 2020.

Dmitry Chistikov, Matthias Englert, and Ranko Lazic. Learning a neuron by a shallow relu network: Dynamics and implicit bias for correlated inputs. *Advances in Neural Information Processing Systems*, 36:23748–23760, 2023.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Clementine Domine, Nicolas Anguita, Alexandra M Proca, Lukas Braun, Daniel Kunin, Pedro Mediano, and Andrew Saxe. From lazy to rich: Exact learning dynamics in deep linear networks. In *International Conference on Learning Representations*, volume 2025, pp. 102485–102536, 2025.

Farzan Farnia, Jesse M Zhang, and David Tse. Generalizable adversarial training via spectral normalization. *arXiv preprint arXiv:1811.07457*, 2018.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Spencer Frei, Gal Vardi, Peter L Bartlett, Nathan Srebro, and Wei Hu. Implicit bias in leaky relu networks trained on high-dimensional data. *arXiv preprint arXiv:2210.07082*, 2022.

Pulkit Gopalani and Anirbit Mukherjee. Global convergence of sgd on two layer neural nets. *Information and Inference: A Journal of the IMA*, 14(1):iaae035, 2025.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Kai Hu, Haoqi Hu, and Matt Fredrikson. Lipnext: Scaling up lipschitz-based certified robustness to billion-parameter models. *arXiv preprint arXiv:2601.18513*, 2026.

Ajay Kumar Jaiswal, Yifan Wang, Lu Yin, Shiwei Liu, Runjin Chen, Jiawei Zhao, Ananth Grama, Yuandong Tian, and Zhangyang Wang. From low rank gradient subspace stabilization to low-rank weights: Observations, theories, and applications. In *Forty-second International Conference on Machine Learning*, 2025.

Yiwen Kou, Zixiang Chen, and Quanquan Gu. Implicit bias of gradient descent for two-layer relu and leaky relu networks on nearly-orthogonal data. *Advances in Neural Information Processing Systems*, 36:30167–30221, 2023.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Souvik Kundu, Massoud Pedram, and Peter A. Beerel. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5189–5198, 2021. doi: 10.1109/ICCV48922.2021.00516.

Soo Min Kwon, Zekai Zhang, Dogyoon Song, Laura Balzano, and Qing Qu. Efficient low-dimensional compression of overparameterized models. In *International Conference on Artificial Intelligence*. PMLR, 2024.

Bo-Han Lai, Pin-Han Huang, Bo-Han Kung, and Shang-Tse Chen. Enhancing certified robustness via block reflector orthogonal layers and logit annealing loss. In *Forty-second International Conference on Machine Learning*, 2025.

Brett W Larsen, Stanislav Fort, Nic Becker, and Surya Ganguli. How many degrees of freedom do we need to train deep networks: a loss landscape perspective. In *International Conference on Learning Representations*, 2022.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Linyi Li, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. In *2023 IEEE symposium on security and privacy (SP)*, pp. 1289–1310. IEEE, 2023.

Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Jörn-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. *Advances in neural information processing systems*, 32, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations (ICLR)*, 2018.

Alexandre Megretski and Anders Rantzer. System analysis via integral quadratic constraints. *IEEE transactions on automatic control*, 42(6):819–830, 1997.

Laurent Meunier, Blaise J Delattre, Alexandre Araujo, and Alexandre Allauzen. A dynamical system perspective for lipschitz neural networks. In *International Conference on Machine Learning*, pp. 15484–15500. PMLR, 2022.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pp. 4951–4960. PMLR, 2019.

Samet Oymak and Mahdi Soltanolkotabi. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1):84–105, 2020.

Vardan Papyan, Xiao Y Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Patricia Pauli, Ruigang Wang, Ian R Manchester, and Frank Allgöwer. Lipkernel: Lipschitz-bounded convolutional neural networks via dissipative layers. *Automatica*, 188:112959, 2026.

Bernd Prach and Christoph H Lampert. Almost-orthogonal layers for efficient general-purpose lipschitz networks. In *European Conference on Computer Vision*, pp. 350–365. Springer, 2022.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

Sahil Singla and Soheil Feizi. Skew orthogonal convolutions. In *International Conference on Machine Learning*, pp. 9756–9766. PMLR, 2021.

Rishi Sonthalia, Michael Murray, and Guido Montufar. Low rank gradients and where to find them. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Asher Trockman and J Zico Kolter. Orthogonalizing convolutional layers with the cayley transform. *arXiv preprint arXiv:2104.07167*, 2021.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Peng Wang, Xiao Li, Can Yaras, Zhihui Zhu, Laura Balzano, Wei Hu, and Qing Qu. Understanding deep representation learning via layerwise feature compression and discrimination. *Journal of machine learning research*, 26(220):1–61, 2025.

Ruigang Wang and Ian Manchester. Direct parameterization of lipschitz-bounded deep networks. In *International Conference on Machine Learning*, pp. 36093–36110. PMLR, 2023.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International conference on machine learning*, pp. 36246–36263. PMLR, 2023.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Hermann Weyl. Inequalities between the two kinds of eigenvalues of a linear transformation. *Proceedings of the national academy of sciences*, 35(7):408–411, 1949.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Alec S Xu, Can Yaras, Matthew Asato, Qing Qu, and Laura Balzano. Emergent low-rank training dynamics in mlps with smooth activations. *arXiv preprint arXiv:2602.06208*, 2026.

Xiaojun Xu, Linyi Li, and Bo Li. Lot: Layer-wise orthogonal training on improving l2 certified robustness. *Advances in Neural Information Processing Systems*, 35:18904–18915, 2022.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International conference on machine learning*, pp. 10693–10705. PMLR, 2020.

Can Yaras, Peng Wang, Wei Hu, Zhihui Zhu, Laura Balzano, and Qing Qu. The law of parsimony in gradient descent for learning deep linear networks. *arXiv preprint arXiv:2306.01154*, 2023.

Can Yaras, Peng Wang, Laura Balzano, and Qing Qu. Compressible dynamics in deep overparameterized low-rank learning & adaptation. *arXiv preprint arXiv:2406.04112*, 2024.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Jinxin Zhou, Chong You, Xiao Li, Kangning Liu, Sheng Liu, Qing Qu, and Zhihui Zhu. Are all losses created equal: A neural collapse perspective. *Advances in Neural Information Processing Systems*, 35:31697–31710, 2022.

A Appendix

A.1 Sandwich layers

Here, we briefly summarize the Sandwich layer parameterization and more details can be found in (Wang & Manchester, 2023). Follow 1, stacking all hidden unit $z_1, \ldots, z_L$ together and obtain a

compact form as follows:

$$\overbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{bmatrix}}^{z} = \sigma\left(\overbrace{\begin{bmatrix} 0 & & & \\ W_2 & \ddots & & \\ \vdots & \ddots & 0 & \\ 0 & \cdots & W_{L-1} & 0 \end{bmatrix}}^{W}\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{bmatrix} + \overbrace{\begin{bmatrix} W_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}^{U} x + \overbrace{\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{L-1} \end{bmatrix}}^{b_z}\right), \tag{29}$$

$$y = \overbrace{\begin{bmatrix} 0 & \cdots & 0 & W_L \end{bmatrix}}^{Y}\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{bmatrix} + \overbrace{b_L}^{b_y}.$$

Let $z := \mathrm{col}(z_1, \ldots, z_L)$ and $v := Wz + Ux + b_z$, we can rewrite the above equation by

$$v = Wz + Ux + b_z, \quad z = \sigma(v), \quad y = Yz + b_y. \tag{30}$$

Then, we proceed to introduce incremental quadratic constraints (iQCs) (Megretski & Rantzer, 1997), which helps to build the Lipschitz certificate for this network.

Lemma A.1 (Incremental quadratic constraint)**.** *If Assumption 2.1 holds, then for any $\Lambda \in \mathbb{D}^n_{++}$ the following iQC holds for any pair of (v^a, z^a) and (v^b, z^b) satisfying $z^a = \sigma(v^a)$ and $z^b = \sigma(v^b)$:*

$$\begin{bmatrix} \Delta v^\top \\ \Delta z^\top \end{bmatrix}^\top \begin{bmatrix} 0 & \Lambda \\ \Lambda & -2\Lambda \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta z \end{bmatrix} \geq 0 \tag{31}$$

where $\Delta v = v^b - v^a$ and $\Delta z = z^b - z^a$.

Proof. Let $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ with $\lambda_j > 0$. For each j, Assumption 2.1 gives

$$2\Delta z_j(\Delta v_j - \Delta z_j) \geq 0. \tag{32}$$

Multiplying by $\lambda_j > 0$ and summing over j yields

$$2\sum_{j=1}^{n} \lambda_j \Delta z_j(\Delta v_j - \Delta z_j) = 2\Delta z^\top \Lambda(\Delta v - \Delta z) \geq 0. \tag{33}$$

Rewriting the above inequality yields 31. □

Following (Fazlyab et al., 2019) and (Wang & Manchester, 2023), LMI certification for network 1 can be characterized as follows.

Theorem A.2. *Assume Assumption 2.1 holds, the network 1 is γ-Lipschitz if there exists a positive diagonal multiplier $\Lambda \in \mathbb{D}^n_{++}$ such that*

$$H := \begin{bmatrix} \gamma I & -U^\top \Lambda & 0 \\ -\Lambda U & 2\Lambda - \Lambda W - W^\top \Lambda & -Y^\top \\ 0 & -Y & \gamma I \end{bmatrix} \succeq 0 \tag{34}$$

Proof. Because $\gamma I_{n_{L+1}}$ is positive definite, applying the Schur complement to H gives

$$\begin{bmatrix} \gamma I_{n_0} & -U^\top \Lambda \\ -\Lambda U & 2\Lambda - \Lambda W - W^\top \Lambda - \frac{1}{\gamma} Y^\top Y \end{bmatrix} \succeq 0. \tag{35}$$

Multiplying equation 35 on the left and right by $[\Delta x^\top, \Delta z^\top]^\top$ yields

$$\begin{aligned} 0 &\leq \gamma \left\|\Delta x\right\|_2^2 - 2\Delta z^\top \Lambda U \Delta x + \Delta z^\top (2\Lambda - \Lambda W - W^\top \Lambda)\Delta z - \frac{1}{\gamma}\Delta z^\top Y^\top Y \Delta z \\ &= \gamma \left\|\Delta x\right\|_2^2 - \frac{1}{\gamma}\left\|\Delta y\right\|_2^2 - 2\Delta z^\top \Lambda(\Delta h - \Delta z), \end{aligned}$$

which implies,

$$\gamma \|\Delta x\|_2^2 - \frac{1}{\gamma}\|\Delta y\|_2^2 \geq 2\Delta z^\top \Lambda(\Delta h - \Delta z).$$

By Lemma A.1, the right-hand side is nonnegative. Therefore,

$$\gamma \|\Delta x\|_2^2 - \frac{1}{\gamma}\|\Delta y\|_2^2 \geq 0 \implies \|\Delta y\|_2^2 \leq \gamma^2 \|\Delta x\|_2^2 \implies \|\Delta y\|_2 \leq \gamma \|\Delta x\|_2 .$$

Thus, the network 1 is γ-Lipschitz. □

Given a network with fixed W, U, Y in 29, let $\tilde{U} = \Lambda U, \tilde{W} = \Lambda W$, 34 become

$$H = \begin{bmatrix} \gamma I & -\hat{W}_0^\top & & & \\ -\hat{W}_0 & 2\Lambda_0 & -\hat{W}_1^\top & & \\ & \ddots & \ddots & \ddots & \\ & & -\hat{W}_{L-1} & 2\Lambda_{L-1} & -\hat{W}_L^\top \\ & & & -\hat{W}_L & \gamma I \end{bmatrix} \succeq 0 \tag{36}$$

where $\hat{W}_k = \Lambda_k W_k$ for $0 \leq k < L$ and $\hat{W}_L = W_L$.

In (Wang & Manchester, 2023), they enforce $H \succeq 0$ by $H = PP^\top$, where P is defined as

$$P = \begin{bmatrix} D_{-1} & & \\ -V_0 & D_0 & \\ & \ddots & \ddots \end{bmatrix}.$$

Therefore, the diagonal and off-diagonal blocks satisfy

$$H_{kk} = V_k V_k^\top + D_k D_k^\top, \quad H_{k-1,k} = -V_k D_{k-1}^\top. \tag{37}$$

Let

$$D_k = \Psi_k A_k, \qquad V_k = \Psi_k B_k, \qquad \Psi_k = \mathrm{diag}(e^{d_k}), \tag{38}$$

where

$$\begin{bmatrix} A_k^\top \\ B_k^\top \end{bmatrix} = \mathrm{Cayley}\left(\begin{bmatrix} X_k \\ Y_k \end{bmatrix}\right) = \begin{bmatrix} (I + Z_k)^{-1}(I - Z_k) \\ -2Y_k(I + Z_k)^{-1} \end{bmatrix}, \quad Z_k = X_k - X_k^\top + Y_k^\top Y_k. \tag{39}$$

The Cayley map ensures $A_k A_k^\top + B_k B_k^\top = I$, therefore

$$H_{kk} = \Psi_k^2, \quad H_{k-1,k} = -\Psi_k B_k A_{k-1}^\top \Psi_{k-1}. \tag{40}$$

Comparing these terms with the corresponding blocks of the LMI 36 gives

$$\Lambda_k = \frac{1}{2}\Psi_k^2, \quad W_k = 2\Psi_k^{-1} B_k A_{k-1}^\top \Psi_{k-1}. \tag{41}$$

Notice that free variables are coupled across layers in 41, change of hidden coordinates yields the Sandwich layer

$$h_{k+1} = \sqrt{2} A_k^\top \Psi_k \sigma\left(\sqrt{2}\Psi_k^{-1} B_k h_k + b_k\right), \tag{42}$$

which is 1-Lipschitz by construction (Wang & Manchester, 2023).

A.2 Multi-layer Sandwich network low-rank training dynamics experiment

In this section, we discuss the task setup and training details for Figure 1 and more ablation study results with different smooth slope-restricted activation functions (CELU, ELU).

Training task: Each experiment uses a $K = 4$ class Gaussian mixture in $d = 32$ dimensions. Total training examples are $N = 2000$, so that each class has 500 training examples. The data is generated as follows: 1.) class centers are sampled from $\mu_k \sim \mathcal{N}(0, I_d)$ and normalized to unit norm; 2.) for each class, 500 samples are generated through $x_{k,i} \sim \mathcal{N}(\mu_k, \sigma^2)$ with $\sigma = 0.35$. Then

we concatenated the training samples as $\Xi \in \mathbb{R}^{d\times N}$ and whitened the input so that $\Xi\Xi^\top/N = I_d$. We use one-hot vectors for labels $T \in \mathbb{R}^{K\times N}$, the loss is the squared error averaged over samples.

Sandwich network: We trained a 4-layer Sandwich network with 3 hidden layers and a linear readout of the form 7, input and hidden dimensions are set to $d = m = 32$. We split the Lipschitz budget as $g_k = 0.5, \text{for } 0 \le k < 3$ and $g_2 = 8$, so that $\left(\prod_{k=0}^{3} g_k\right) g_r = \gamma = 1$. For hidden layers $1 \le k < 3$, each Y_k is initialized as $Y_k = \rho_k Q_k, \rho_k = \epsilon/(2\sqrt{2}g_k)$ with $\epsilon = 0.1$. For final linear readout, Y_2 is initialized as $Y_2 = \rho_2 Q_2, \rho_2 = 0.9$. Here, Q_k and Q_2 are random orthogonal matrices. With this initialization, every hidden layer starts with a flat spectrum and small pre-activation of order ϵ. All X_k are initialized at 0.

Singular subspace metric: We perform SVD to $Y_k(t)$ in each hidden layer that yields:

$$Y_k(t) = [L_{k,1}(t) \quad L_{k,2}(t) \quad L_{k,3}(t)] \begin{bmatrix} s_{k,1}(t) & 0 & 0 \\ 0 & s_{k,2}(t) & 0 \\ 0 & 0 & s_{k,3}(t) \end{bmatrix} [R_{k,1}(t) \quad R_{k,2}(t) \quad R_{k,3}(t)]^\top,$$

where $L_{k,i}(t)$ and $R_{k,i}(t)$ with $1 \le i \le 3$ represent the left/right singular subspaces of top-K, middle $d-2K$ and bottom-K during training, respectively. We track the top-K and bottom-K left singular subspaces changes using

$$\|\sin\Phi\left(L_{k,1}(t), L_{k,1}(0)\right)\|_2^2 \quad \text{and} \quad \|\sin\Phi\left(L_{k,3}(t), L_{k,3}(0)\right)\|_2^2, \tag{43}$$

similar for right subspaces and $\sin\Phi$ is defined in Definition 3.5. This metric tracks the largest direction change from initialization. For the middle $d-2K$ singular subspaces, we use the following metric,

$$\|\sin\Phi\left(L_{k,2}(t), U_{k,2}\right)\|_F^2 \quad \text{and} \quad \|\sin\Phi\left(R_{k,2}(t), V_{k,2}\right)\|_F^2, \tag{44}$$

which measures the average singular subspace direction drift from initialization. The reference subspaces $U_{k,2}$ and $V_{k,2}$ are calculated once at initialization from the loss gradient with respect to each Y_k. Let

$$\nabla_{Y_k}\mathcal{L}(0) \approx U_K \Sigma_K V_K^\top$$

denote its rank-K SVD and recall $Y_k(0) = \rho_k Q_k$ denote the initial weight. Let $U_{k,2}$ and $V_{k,2}$ be the orthogonal complements of

$$\text{span}(U_K, Q_k V_K) \quad \text{and} \quad \text{span}(V_K, Q_k^\top U_K), \tag{45}$$

respectively. Theoretically, our analysis predicts that the middle left/right singular subspaces remain close to $U_{k,2}$ and $V_{k,2}$ throughout training.

Here we show that similar trends exist in the second layer singular subspace with Tanh and ReLU activation functions 6, as well as additional ablations of smooth activation functions 7. For CELU, we set the "alpha=4" in PyTorch function and use the default ELU. Across different activation functions and layers, the low-rank training dynamics emerge, and smooth functions exhibit stronger low-rank dynamics.

A.3 Two-layer Sandwich network experiment setup

Here we summarize the setup of the task and network structure used in plotting Fig.2.

Training task: Similar task set up as A.2. Though in this figure, we set Gaussian mixture in $d = 64$ dimensions and total train examples are $N = 1000$, so that each class has 250 training examples. We trained the network with full-batch GD with step size 0.01 for 100 epochs over 10 seeds.

Sandwich network: We trained a 2-layer Sandwich network of the form 7, input and hidden dimensions are set to $d = m = 64$. We split the Lipschitz budget as $g_1 = 0.5$ and $g_2 = 2$, so that $g_1 g_2 = 1$. X_1 and X_2 are initialized as zero and held fixed. $Y_1 = \rho_1 Q_1, \rho = \epsilon/(2\sqrt{2}g_1)$ with $\epsilon = 0.1$. For the linear readout, Y_2 is initialized as $Y_2 = \rho_2 Q_2, \rho_2 = 0.9$. Here, Q_1 and Q_2 are random orthogonal matrices. Linear readout $W_2(0) = -2g_2 C_2 Y_2(0)^\top$ is held fixed throughout training. For CELU, we set the "alpha=4" in PyTorch function and use the default ELU.

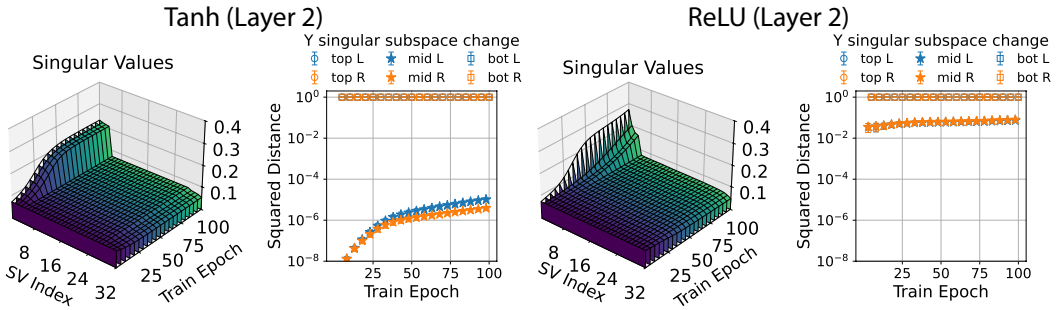


Figure 6: The second layer Y_2 singular subspaces change with slope-restricted smooth and slope-restricted non-smooth activation functions. Similar to the first layer, middle singular subspaces have much smaller changes with smooth activation function versus non-smooth activation function throughout training.

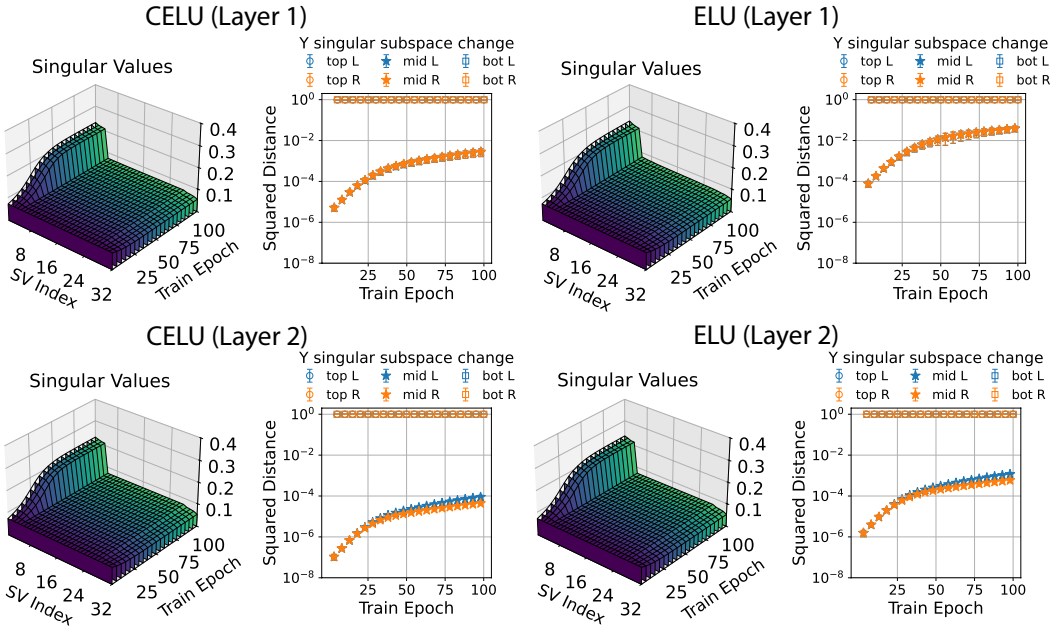


Figure 7: The first and the second layer Y_1 and Y_2 singular subspaces change with different slope-restricted smooth activation functions ablation. The slow middle singular subspace change is consistent across different smooth slope-restricted activation functions.

A.4 Multi-layer Sandwich network experiment setup

In this section, we summarize the details of deep Sandwich networks ablation experiments in Fig. 3.

Training task: Similar task set up as A.2. For Fig. 3 and Fig. 8, however, we set the Gaussian mixture in $d = 64$ dimensions with a total of $N = 1000$ training examples, so that each class has 250 training examples. We trained the network with full-batch GD with step size 0.01 for 250 epochs over 10 seeds. For the loss and optimizer ablations we fixed the model, initialization, data and replace GD on the squared error by mini-batch SGD ($\eta = 10^{-2}$), Adam ($\eta = 10^{-5}$) or Muon ($\eta = 10^{-5}$) on the cross-entropy loss. All with batches of 100 samples and unwhitened inputs.

Sandwich network: Same architecture and initialization as A.2, with hidden width set to $m = 72$. All parameters, X_k and Y_k for every layer including the readout, are trained, while each X_k is initialized to zero. For CELU, we set the "alpha=4".

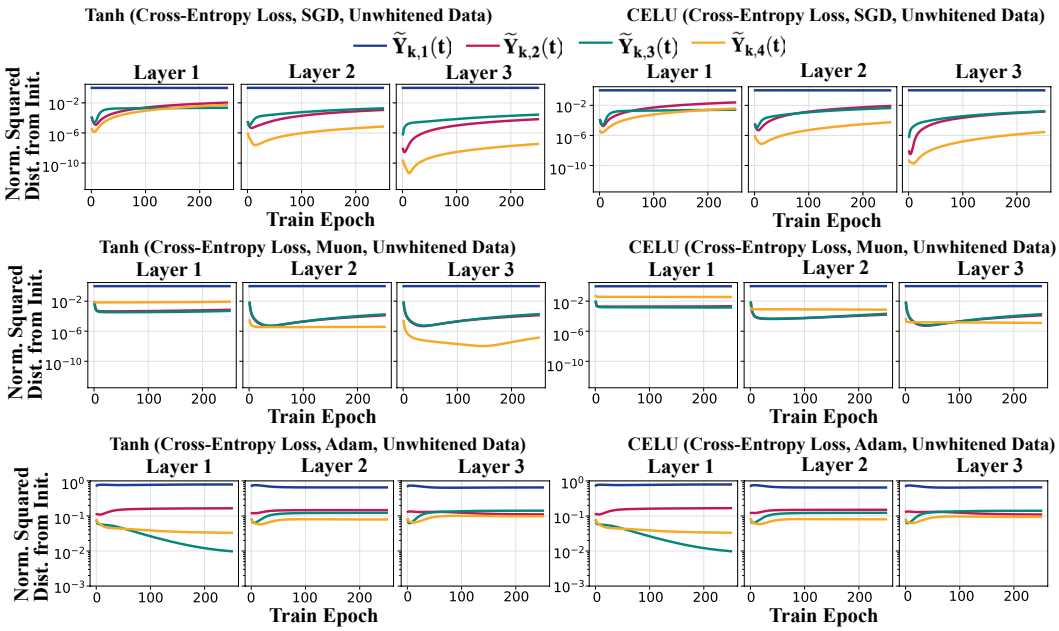


Figure 8: Deep Sandwich network ablation study. We replace the squared-error loss with cross-entropy, consider different optimizers including SGD with momentum, Adam, and Muon, and additionally evaluate training on unwhitened data. Across these settings, we consistently observe similar low-rank training dynamics in Y_k.

A.5 Fashion MNIST training details and ablation results

In this section, we provide training details and ablation results for Fashion-MNIST experiments.

Data pre-processing: Each image is flattened and scaled to [0, 1].

Sandwich Networks:

Dense model: we trained a 4-layer Sandwich network of the form 7. Three hidden layers with input and hidden dimensions $d = m = 784$ and a linear Sandwich readout to $K = 10$ classes. All hidden gains g_k and readout gain g_L are set to one, so the prescribed Lipschitz bound is 1. Each hidden layer carries a full $Y_k \in \mathbb{R}^{784\times784}$ and a skew part S_k with X_k is initialized as zero and $Y_k = \rho_k Q_k$ with Q_k is a semi-orthogonal matrix, $\rho_k = \epsilon/(\sqrt{2} + \sqrt{2-\epsilon^2}) \approx \epsilon/(2\sqrt{2})$, $\epsilon = 0.01$. The readout is $Y_L = \rho_L Q_L$ with $Q_L \in \mathbb{R}^{784\times10}$ random semi-orthogonal, $\rho_o = \epsilon/(g_l + \sqrt{g_L^2 - \epsilon^2}) \approx \epsilon/2$ and X_L is set to zero.

Low-rank model: we adapt the same architecture as the full-rank dense model but initialize the network following section 4. For the random low-rank model, we adapt the same low-rank structure as specified in section 4, but the endpoint subspaces $V \in \mathbb{R}^{784\times20}$ and $U \in \mathbb{R}^{784\times20}$ are instead initialized independently as random semi-orthogonal matrices.

Training: All models are trained for 1500 epochs using full-batch SGD with MSE loss, without momentum or weight decay. We use a triangular learning-rate schedule with a peak learning rate of $\eta = 3 \times 10^{-5}$.

Fig. 9 shows the similar results for CELU activation function. For CELU we set "alpha=5". Low-rank model (41K) with proper initialization closely match dense model (2.77M) test accuracy and test loss trajectory while the randomly initialized low-rank model has much worse test accuracy and high loss values throughout the training.

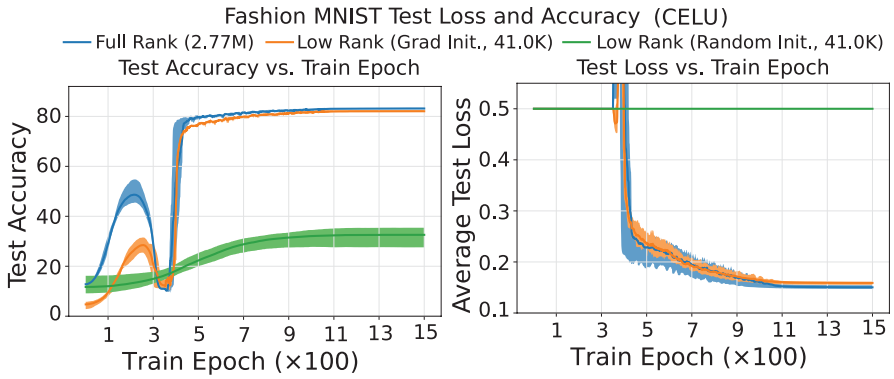


Figure 9: Fashion-MNIST results with CELU. The properly initialized low-rank model closely matches the performance of the dense model, while the randomly initialized low-rank model achieves much lower accuracy and its test loss remains high throughout training.

A.6 Lipschitz tightness

Training task: We use a square-wave task to test Sandwich network Lipschitz tightness, similar to (Wang & Manchester, 2023). We sample $N = 300$ training points from $x_i \sim \mathcal{U}[-2, 2]$ and use 200 uniformly sampled test points on $[-2, 2]$. The targets are $y = 1$ for $x \leq -1$ and $0 < x \leq 1$, and $y = 0$ otherwise. We used mean squared error loss for this task.

Sandwich networks: We use a 10-layer Sandwich network with nine hidden layers and a linear readout. To fit this square wave, we added a trainable bias in the readout layer. The low-rank model is initialized following section 4.

We swept the core rank $r \in \{4, 8, 16\}$ and train for 400 epochs using SGD with momentum 0.9, weight decay 5×10^{-4}. We used a batch size 50 with the triangular learning-rate scheduler with peak at 0.03 over three seeds. We estimate the empirical Lipschitz lower bound γ_{lb} following Wang & Manchester (2023) and evaluate tightness $\gamma_{\text{lb}}/\gamma$ with the fitted slope at the $x = 0$. Fig. 10 shows that the low-rank Sandwich networks have a tight empirical Lipschitz bound, although smooth slope-restricted activations can introduce more contraction compared with norm-preserving activations such as GroupSort (Anil et al., 2019), which are generally better choices for Lipschitz networks.

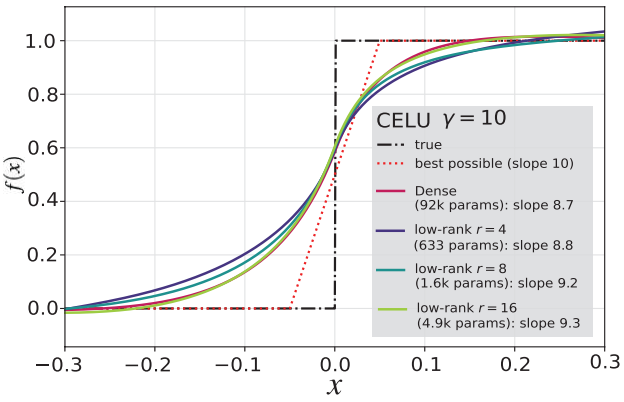


Figure 10: Fitting a square wave using original sandwich layer, our full rank model and our low-rank models with different ranks with Lipschitz bound of 10.

Effective weight norms: Recall single Sandwich layer has the following form

$$h_{k+1} = \sqrt{2}(2C_k - I)\sigma\left(-2\sqrt{2}g_k C_k^\top Y_k^\top h_k\right), \tag{46}$$

Combine the output map of one hidden layer with the input map of the next, the effective weights are

$$-2\sqrt{2}g_1C_1^\top Y_1^\top, \quad -4g_kC_k^\top Y_k^\top(2C_{k-1}-I), \text{ for } 2 \le k < L, \tag{47}$$

with final readout

$$-2\sqrt{2}g_LC_L^\top Y_L^\top(2C_{L-1}-I). \tag{48}$$

Fig. 5 shows that the low-rank Sandwich networks have similar property of its dense counterpart, which allows each effective weight norms to exceed 1 while preserving the prescribed Lipschitz bound.

A.7 CIFAR-10 AND CIFAR-100 EXPERIMENTS

Note that the model and hyperparameters are not tuned for best performance, the primary goal is to study the low-rank training dynamics.

Training task: For image classification tasks, we also applied the data augmentation used by (Wang & Manchester, 2023; Araujo et al., 2023). AutoAttack accuracy at $\epsilon \in 36, 72, 108/255$ on the first 1000 test images.

CIFAR-10:

Sandwich network: We use the pretrained CNN backbone from the released KWL-Sandwich-small model with Lipschitz bound $\gamma = 1$ (Wang & Manchester, 2023). The backbone outputs a 4096-dimensional feature vector, we then freeze the backbone and fine-tuned a three layers Sandwich classifier consisting of two hidden layers $4096 \to 512 \to 512$ and a dense readout which maps to 10 classes. For the low-rank classifier, only the two hidden layers are factorized through an r-dimensional core follow section 4, while the readout remains dense and fully trainable. We sweep $r \in \{10, 20, 40, 80, 160\}$ and compare gradient-informed initialization against a random semi-orthogonal initialization of the endpoint subspaces, all other training settings are identical. We use CELU with $\alpha = 0.5$. We train the model using SGD with momentum 0.9, weight decay 5×10^{-4}, batch size 256 for 100 epochs with a cosine learning-rate schedule from 10^{-2}. We use multi-margin loss with margin 0.5 aligned with (Wang & Manchester, 2023) over 3 seeds. The dense model uses the same setup, with the hidden layers parameterized at full rank.

Fig. 11 shows CIFAR-10 test loss trajectory and test accuracy for low-rank Sandwich rank ablation study. The properly initialized low-rank Sandwich classifier closely matches the dense model in both test accuracy and test-loss trajectory, whereas the randomly initialized low-rank classifier remains at low accuracy. The rank-10 model also shows visibly worse performance than higher-rank variants, consistent with theoretical analysis

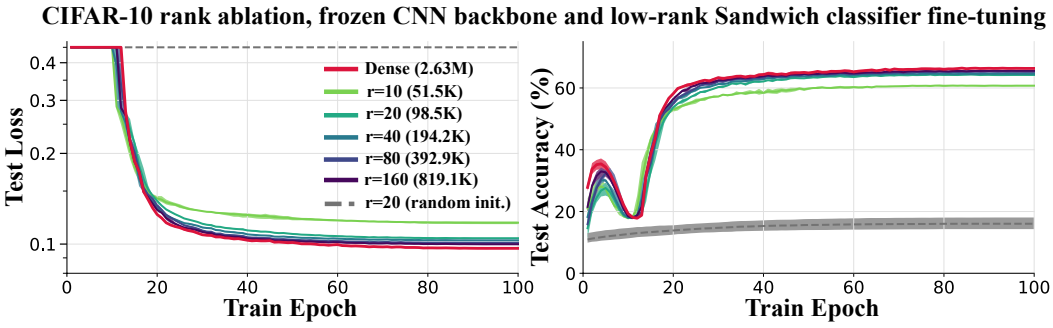


Figure 11: CIFAR-10 test loss and accuracy for the rank ablation of the fine-tuned low-rank Sandwich classifier.

CIFAR-100:

Sandwich network: We use the pretrained CNN backbone from the released KWL-Sandwich-large model with Lipschitz bound $\gamma = 2$ (Wang & Manchester, 2023). Similar to CIFAR-10 experiment, we fine-tuned the low-rank classifier, only the two hidden layers are factorized through

an r-dimensional core following Section 4, while the readout remains dense and fully trainable. $r \in \{200, 400, 800\}$ and compare gradient-informed initialization against a random semi-orthogonal initialization of the endpoint subspaces, all other training settings are identical. We use CELU with $\alpha = 0.5$. We train the model using SGD with momentum 0.9, weight decay 5×10^{-4}. Batch size is set to 256 for 100 epochs with a cosine learning-rate schedule from 10^{-2}. We use offset cross-entropy loss (offset 1.5, temperature 0.25) over 3 seeds for CIFAR-100 follow (Wang & Manchester, 2023). The dense model used the same setup.

Fig. 12 shows that CIFAR-100 test loss trajectory and test accuracy for the low-rank Sandwich rank ablation study. Similar to CIFAR-10 results.

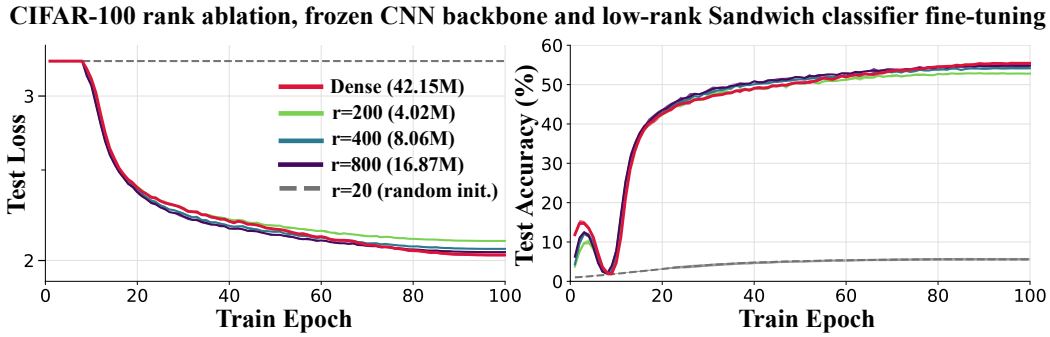


Figure 12: CIFAR-100 test loss and accuracy for the rank ablation of the fine-tuned low-rank Sandwich classifier.

A.8 Discussion on Assumption 3.4

In this section, we empirically validate Assumption 3.4. Following Xu et al. (2026), we examine whether the gradient norm and its top-K singular values decay during training. We use the same experimental setup as Fig. 2. As in Figure 13, $\frac{\|G_{Y_1}(t)\|_F}{\|G_{Y_1}(0)\|_F}$ and $\frac{s_i(G_{Y_1}(t))}{s_i(G_{Y_1}(0))}$, for $i \in [K]$ decreases throughout training. We also set $c_2 = 2$ to show that $\frac{s_i(G_1(t))}{s_i(G_1(0))} \leq c_2 \frac{\|G_1(t)\|_F}{\|G_1(0)\|_F}$, for $i \in [K]$. These observations empirically support assumptions on the trajectory bound $\frac{\|G_{Y_1}(t)\|_F}{\|G_{Y_1}(0)\|_F} \leq c_1 \lambda^t, \lambda = e^{-c\eta}$.

Then we also plot the K-th singular value of the analytical gradient at initialization $s_K(\frac{4ag_1}{N} \Xi T^\top W_2)$ versus $(K+1)$s-th real gradient $s_{K+1}(G_{Y_1}(t))$ over different $\epsilon \in [0.075, 0.125]$ across 10 trials. Throughout the training, $s_K(\frac{4ag_1}{N} \Xi T^\top W_2)$ is much larger than $s_{K+1}(G_{Y_1}(t))$, which justifies our assumption that $s_K(M) - s_{K+1}(G_1(t)) \geq c_3 s_K(\frac{4ag_1}{N} \Xi T^\top W_2)$. The trends behave similarly across tanh and CELU activation functions.

A.9 Theoretical analysis on two-layer Sandwich network

In this section, we proceed to discuss the details of the subspace characterization. We first introduce the two-layer Sandwich network and the notation used to derive its analytical gradient.

Assumption A.3 (Two-layer Sandwich network)**.** We consider the network in equation 7 with $L = 2$, consisting of a single Sandwich layer followed by a *frozen* linear readout. Throughout training, Y_1 is the only trainable parameter, whereas X_1, X_2, and the readout parameters are held fixed. In particular,

$$S_1 := X_1 - X_1^\top = 0, \quad C_1 := \left(I + S_1 + Y_1^\top Y_1\right)^{-1} = \left(I + Y_1^\top Y_1\right)^{-1} \in \mathbb{R}^{m \times m}, \tag{49a}$$

$$S_2 := X_2 - X_2^\top = 0, \quad C_2 := \left(I + S_2 + Y_2^\top Y_2\right)^{-1} = \left(I + Y_2(0)^\top Y_2(0)\right)^{-1} \in \mathbb{R}^{m \times m}. \tag{49b}$$

The readout is initialized from $Y_2(0)$ and then frozen:

$$W_2(0) := -2g_2 C_2 Y_2(0)^\top, \qquad W_2(t) \equiv W_2(0). \tag{50}$$

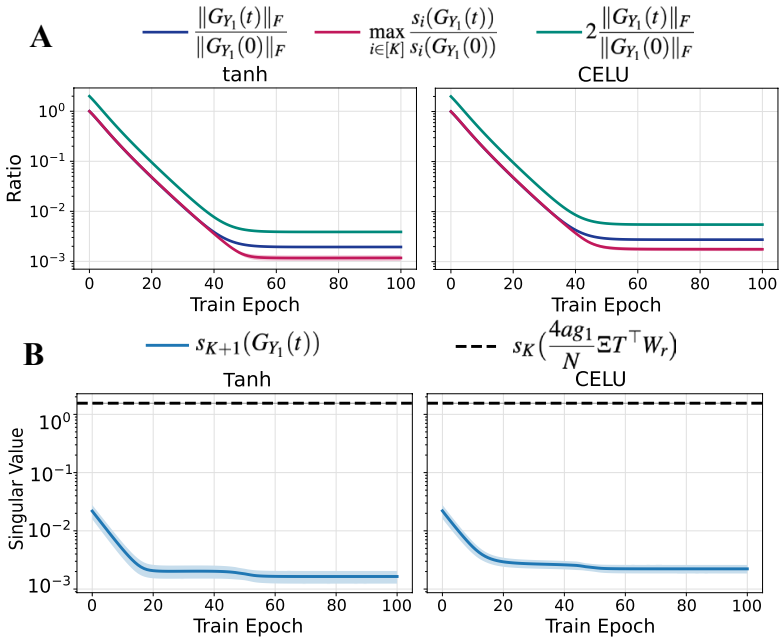


Figure 13: **A.** Empirical validation of Assumption 3.4. The gradient norm $\|G_{Y_1}(t)\|_F$ decreases relative to $\|G_{Y_1}(0)\|_F$, and the top-K singular values of the gradient, $s_i(G_{Y_1}(t))$ for $i \in [K]$, decrease relative to $s_i(G_{Y_1}(0))$ throughout training. **B.** The K^{th} singular value of the analytical gradient at initialization, $s_{K+1}\left(\frac{4ag_1}{N}\Xi T^\top W_2\right)$ remains substantially larger than the actual gradient tail $s_{K+1}(G_{Y_1}(t))$. Together, these trends empirically support Assumption 3.4.

For inputs $\Xi \in \mathbb{R}^{d\times N}$, the globally γ-Lipschitz two-layer Sandwich network is defined as:

$$\begin{aligned} H_0 &:= \Xi, \\ Z_1 &:= -2\sqrt{2}g_1 C_1^\top Y_1^\top H_0, \\ H_1 &:= \sigma(Z_1), \\ Z_2 &:= \sqrt{2}W_2(2C_1 - I)H_1, \end{aligned} \tag{51}$$

where $g_1 g_2 = \gamma$. $Z_1, H_1 \in \mathbb{R}^{m\times N}$ and $Z_2 = f_{Y_1}(\Xi) \in \mathbb{R}^{K\times N}$.

A.9.1 Analytical gradients

We start to derive the loss gradients on Y_1,i.e., $G_1(t) = \nabla_{Y_1}\mathcal{L}(Y_1(t))$. Let

$$\Delta_2 := \nabla_{Z_2}\mathcal{L} = \frac{Z_2 - T}{N}, \quad \Delta_1 := \nabla_{Z_1}\mathcal{L} = 2\sqrt{2}g_1(2C_1 - I)W_2^\top \Delta_2 \odot \sigma' Z_1. \tag{52}$$

We first calculate the loss Jacobian with respect to C_1 which affects loss either via Z_2 or Z_1 in 51:

$$d\mathcal{L} = \langle \Delta_2, dZ_2\rangle_F = 2\sqrt{2}\langle W_2^\top \Delta_2 H_1^\top, dC_1\rangle_F, d\mathcal{L} = \langle \Delta_1, dZ_1\rangle_F = -2\sqrt{2}g_1\langle \Delta_1 H_0^\top Y_1, dC_1\rangle_F.$$

Thus,

$$J_{C_1} := \frac{\partial \mathcal{L}}{\partial C_1} = 2\sqrt{2}(W_2^\top \Delta_2 H_1^\top - g_1 \Delta_1 H_0^\top Y_1). \tag{53}$$

Loss gradient with respect to Y_1 either directly coming from Y_1 via Z_1:

$$d\mathcal{L} = \langle \Delta_1, dZ_1\rangle_F = -2\sqrt{2}g_1\langle H_0 \Delta_1^\top C_1, dY_1\rangle_F,$$

or coming from C_1:

$$d\mathcal{L}_{C_1} = \left\langle J_{C_1}, -C_1\left[(dY_1^\top)Y_1 + Y_1^\top(dY_1)\right]C_1\right\rangle_F = \left\langle -Y_1 C_1(J_{C_1} + J_{C_1}^\top)C_1, dY_1\right\rangle_F.$$

Therefore, the gradient on Y_1 is:

$$G_1(t) = -2\sqrt{2}g_1 H_0 \Delta_1^\top C_1 - Y_1 C_1(J_{C_1} + J_{C_1}^\top)C_1. \tag{54}$$

A.9.2 GRADIENT BOUNDS

In this section, we derive bounds on gradient norm $\|G_1(t)\|_F$, which will be used later to characterize changes in the dominant singular subspaces.

Lemma A.4 (Gradient bounds)**.** *Given the input data whitened scale v and input dimension d, for every Y_1, the following inequality holds:*

$$\|Z_2 - T\|_F \le E_F, \quad E_F := 2g_1\|W_2\|_2\sqrt{vd} + \|T\|_F. \tag{55}$$

Along the training trajectory, gradient $G_1(t)$ is bounded:

$$\|G_1(t)\|_F \le J_F\sqrt{\frac{2\mathcal{L}(t)}{N}}, \quad J_F := 8g_1\|W_2\|_2\sqrt{vd}. \tag{56}$$

Proof. Compare 5a and 51 shows the following relations to the Cayley variables:

$$A_1 = 2C_1 - I, \quad B_1 = -2Y_1C_1 \tag{57}$$

Since A_1 and B_1 satisfy Cayley transformation $A_1A_1^\top + B_1B_1^\top = I$,

$$\|A_1\|_2 \le 1, \quad \|B_1\|_2 \le 1. \tag{58}$$

Therefore,

$$\|Z_1\|_F = \sqrt{2}g_1\|B_1^\top H_0\|_F \le \sqrt{2}g_1\|B_1\|_2\|H_0\|_F = \sqrt{2}g_1\sqrt{vd}. \tag{59}$$

Moreover, activation function is slope-restricted $0 \le \sigma'(\cdot) \le 1$, then:

$$\|Z_2\|_F \le 2g_1\|W_2\|_2\sqrt{vd}. \tag{60}$$

Thus,

$$\|Z_2 - T\|_F \le 2g_1\|W_2\|_2\sqrt{vd} + \|T\|_F.$$

This proves 55.

Notice that

$$\|dA_1\|_2 \le 2\|dY_1\|_F, \quad \|dB_1\|_2 \le 2\|dY_1\|_F. \tag{61}$$

Therefore,

$$\|dZ_1\|_F \le 2\sqrt{2}g_1\|H_0\|_F\|dY_1\|_F. \tag{62}$$

Then consider,

$$Z_2 = \sqrt{2}W_2A_1^\top\phi(Z_1),$$

for a small perturbation on Y_1,

$$dZ_2 = \sqrt{2}W_2\left[(dA_1)^\top\sigma(Z_1) + A_1^\top\left(\sigma'(Z_1)\odot dZ_1\right)\right]. \tag{63}$$

$$\|dZ_2\| \le \sqrt{2}\|W_2\|_2\left(\|dA_1\|_2\|\sigma(Z_1)\|_F + \|A_1\|_2\|dZ_1\|_F\right) \tag{64}$$

$$\le \sqrt{2}\|W_2\|_2\left(2\|Z_1\|_F\|dY_1\|_F + \|dZ_1\|_F\right). \tag{65}$$

Substitute 59 and 62 yields

$$\|dZ_2\|_F \le \sqrt{2}\|W_2\|_2\left[2\sqrt{2}g_1\|H_0\|_F + 2\sqrt{2}g_1\|H_0\|_F\right]\|dY_1\|_F \tag{66}$$

$$= 8g_1\|W_2\|_2\|H_0\|_F\|dY_1\|_F \tag{67}$$

$$= 8g_1\|W_2\|_2\sqrt{vd}\|dY_1\|_F \tag{68}$$

$$= J_F\|dY_1\|_F. \tag{69}$$

By the definition of Δ_2 in 52,

$$\|\Delta_2\|_F = \frac{\|Z_2 - T\|_F}{N}, \quad \mathcal{L}(t) = \frac{1}{2N}\|Z_2(t) - T\|_F^2. \tag{70}$$

Moreover,

$$\|G_1\|_F = \|\nabla_{Y_1}\mathcal{L}\|_F \le J_F\|\Delta_2\|_F. \tag{71}$$

Therefore,

$$\|G_1(t)\|_F \le J_F\sqrt{\frac{2\mathcal{L}(t)}{N}}. \tag{72}$$

□

A.10 GRADIENT SMOOTHNESS

In this section, we discuss gradient smoothness and it will be used to control the rotation of the leading singular subspaces in the subsequent analysis.

Lemma A.5 (Global Lipschitz continuity of the gradient)**.** *For any $Y_1, Y_1' \in \mathbb{R}^{d\times m}$,*

$$\|G_1(Y_1) - G_1(Y_1')\|_F \le L_G \|Y_1 - Y_1'\|_F, \tag{73}$$

where

$$L_G := \frac{J_F^2 + E_F L_F}{N}, \quad L_F := 32 g_1 \|W_2\|_2 \sqrt{\nu d} + 8\sqrt{2}\mu g_1^2 \|W_2\|_2 \nu\sqrt{d}, \tag{74}$$

with E_F and J_F are defined in Lemma A.4.

Proof. Let

$$\delta := \|Y_1 - Y_1'\|_F, \quad h := \|dY_1\|_F. \tag{75}$$

For a small perturbation dY_1, Cayley variables satisfy

$$\|dA_1(Y_1) - dA_1(Y_1')\|_2 \le 4\delta h, \quad \|dB_1(Y_1) - dB_1(Y_1')\|_2 \le 4\delta h. \tag{76}$$

Since

$$Z_1(Y_1) = \sqrt{2} g_1 B_1(Y_1)^\top H_0, \tag{77}$$

we have

$$\|Z_1(Y_1) - Z_1(Y_1')\|_F \le 2\sqrt{2} g_1 \|H_0\|_F \delta, \tag{78}$$

and

$$\|dZ_1(Y_1) - dZ_1(Y_1')\|_F \le 4\sqrt{2} g_1 \|H_0\|_F \delta h. \tag{79}$$

Moreover, by 59,

$$\|Z_1(Y_1)\|_F \le \sqrt{2} g_1 \|H_0\|_F, \tag{80}$$

and

$$\|dZ_1(Y_1)\|_F \le 2\sqrt{2} g_1 \|H_0\|_F h. \tag{81}$$

Recall that

$$Z_2(Y_1) = \sqrt{2} W_2 A_1(Y_1)^\top \sigma(Z_1(Y_1)). \tag{82}$$

By 63, subtracting the expression at Y_1' and using $0 \le \sigma'(z) \le 1$, $|\sigma''(z)| \le \mu$, with the bounds above, then

$$\|dZ_2(Y_1) - dZ_2(Y_1')\|_F \le L_F \delta h, \tag{83}$$

where

$$L_F := 32 g_1 \|W_2\|_2 \sqrt{\nu d} + 8\sqrt{2}\mu g_1^2 \|W_2\|_2 \nu\sqrt{d}. \tag{84}$$

By Lemma A.4,

$$\|dZ_2(Y_1)\|_F \le J_F \|dY_1\|_F. \tag{85}$$

Therefore,

$$\|Z_2(Y_1) - Z_2(Y_1')\|_F \le J_F \|Y_1 - Y_1'\|_F. \tag{86}$$

Recall that

$$\Delta_2(Y_1) = \frac{Z_2(Y_1) - T}{N}. \tag{87}$$

Thus,

$$\|\Delta_2(Y_1) - \Delta_2(Y_1')\|_F \le \frac{J_F}{N}\delta. \tag{88}$$

Also, by Lemma A.4,

$$\|\Delta_2(Y_1')\|_F \le \frac{E_F}{N}. \tag{89}$$

Therefore,

$$\begin{aligned} \langle G_1(Y_1) - G_1(Y_1'), dY_1\rangle_F = {} & \langle \Delta_2(Y_1) - \Delta_2(Y_1'), dZ_2(Y_1)\rangle_F \\ & + \langle \Delta_2(Y_1'), dZ_2(Y_1) - dZ_2(Y_1')\rangle_F . \end{aligned} \tag{90}$$

By Cauchy–Schwarz,

$$\begin{aligned}|\langle G_1(Y_1) - G_1(Y_1'), dY_1\rangle_F| &\le \|\Delta_2(Y_1) - \Delta_2(Y_1')\|_F \|dZ_2(Y_1)\|_F \\ &\quad + \|\Delta_2(Y_1')\|_F \|dZ_2(Y_1) - dZ_2(Y_1')\|_F \\ &\le \frac{J_F^2 + E_F L_F}{N}\delta h.\end{aligned} \tag{91}$$

Thus,

$$|\langle G_1(Y_1) - G_1(Y_1'), dY_1\rangle_F| \le L_G \|Y_1 - Y_1'\|_F \|dY_1\|_F. \tag{92}$$

Therefore, for every perturbation dY_1, the following bound holds:

$$\|G_1(Y_1) - G_1(Y_1')\|_F \le L_G \|Y_1 - Y_1'\|_F. \tag{93}$$

□

A.11 Initial gradient singular value

In this section, we discuss the gradient at initialization and we show that the gradient $G_1(0)$ at initialization is approximately rank-K.

Lemma A.6. *Assume assumptions 3.1, 3.2, 3.3 and 3.4 hold and initialize Y_1 as follows:*

$$Y_1(0) = \rho Q, \quad QQ^\top = I_d, \quad \rho > 0, \tag{94}$$

where $Q \in \mathbb{R}^{d\times m}$ is row-orthogonal. Then

$$\|G_1(0) - M\|_2 \le r_\rho, \quad M := \frac{4ag_1}{N}\Xi T^\top W_2. \tag{95}$$

Consequently, the singular value of initial gradient $G_1(0)$ on Y_1 satisfies

$$\begin{cases} |s_i(G_1(0)) - s_i(M)| \le r_\rho, & i \le K, \\ s_i(G_1(0)) \le r_\rho, & i > K. \end{cases} \tag{96}$$

where $r_\rho = O(\rho)$.

Proof. We first derive important spectrum norms and then later to bound the singular value norm using Weyl's inequality(Weyl, 1949). Let

$$P_Q := Q^\top Q, \quad C_0 := C(Y_1(0)). \tag{97}$$

Since Q is row-orthogonal, P_Q satisfies

$$C_0 = (I + \rho^2 P_Q)^{-1} = I + (c_\rho - 1)P_Q, \quad c_\rho := \frac{1}{1+\rho^2}. \tag{98}$$

$P_Q^\top Q^\top = Q^\top$ implies $C_0^\top Q^\top = c_\rho Q^\top$. Thus, by 51, the first pre-activation satisfies

$$Z_1(0) = -\epsilon_\rho Q^\top \frac{\Xi}{\sqrt{\nu}}, \quad \epsilon_\rho := 2\sqrt{2} g_1 \rho c_\rho \sqrt{\nu}. \tag{99}$$

Since Q is row-orthogonal and the input data Ξ is whitened

$$QQ^\top = I_d, \quad \frac{1}{\nu}\Xi\Xi^\top = I_d, \tag{100}$$

we have

$$\|Z_1(0)\|_2 = \epsilon_\rho, \quad \|Z_1(0)\|_F = \epsilon_\rho\sqrt{d}, \quad \|Z_1(0)\|_{\max} \le \epsilon_\rho. \tag{101}$$

Since

$$(-\epsilon_\rho Q^\top)^\top(-\epsilon_\rho Q^\top) = \epsilon_\rho^2 I_d, \tag{102}$$

and $\Xi/\sqrt{\nu}$ is whitened, following Lemma D.2 in Xu et al. (2026) yields

$$\|Z_1(0) \odot Z_1(0)\|_1 \le \epsilon_\rho^2, \quad \|Z_1(0) \odot Z_1(0)\|_\infty \le \epsilon_\rho^2. \tag{103}$$

Then, consider the Taylor expansion of $\sigma(Z_1(0))$, since $\|\sigma''(\cdot)\| \le \mu$ by 3.3, we have

$$|\sigma(Z_1(0)) - aZ_1(0)| \le \frac{\mu}{2}|Z_1(0) \odot Z_1(0)|^2, \quad a := \sigma'(0), \tag{104}$$

For any $A \in \mathbb{R}^{m\times n}$ the following norm inequality holds $\|A\|_2 \le \sqrt{\|A\|_1\|A\|_\infty}$. Moreover,

$$\|\sigma(Z_1(0)) - aZ_1(0)\|_1 \le \frac{\mu}{2}\|Z_1(0) \odot Z_1(0)\|_1, \tag{105}$$

$$\|\sigma(Z_1(0)) - aZ_1(0)\|_\infty \le \frac{\mu}{2}\|Z_1(0) \odot Z_1(0)\|_\infty \tag{106}$$

Therefore, using the norm inequality with 103 yields

$$\|\sigma(Z_1(0)) - aZ_1(0)\|_2 \le \frac{\mu}{2}\epsilon_\rho^2. \tag{107}$$

Thus

$$\begin{aligned}\|H_1(0)\|_2 &= \|\sigma(Z_1(0))\|_2 &(108)\\ &\le a\|Z_1(0)\|_2 + \|\sigma(Z_1(0)) - aZ_1(0)\|_2 &(109)\\ &\le u_\rho, &(110)\end{aligned}$$

where $u_\rho = a\epsilon_\rho + \frac{\mu}{2}\epsilon_\rho^2$. Recall at initialization,

$$Z_2(0) = \sqrt{2}W_2(2C_0 - I)^\top H_1(0), \tag{111}$$

and $\|2C_0 - I\|_2 \le 1$ by 57

$$\begin{aligned}\|Z_2(0)\|_F &\le \sqrt{2}\|W_2\|_2\|H_1(0)\|_F &(112)\\ &\le \sqrt{2}\|W_2\|_2\epsilon_\rho\sqrt{d}. &(113)\end{aligned}$$

Therefore,

$$\|\Delta_2(0)\|_F = \frac{\|Z_2 - T\|_F}{N} \le v_\rho, \quad v_\rho := \frac{\sqrt{2}\|W_2\|_2\epsilon_\rho\sqrt{d} + \|T\|_F}{N}. \tag{114}$$

Now, recall we derive the gradient in A.9.1

$$G_1(t) = -2\sqrt{2}g_1H_0\Delta_1^\top C_1 - Y_1C_1(J_{C_1} + J_{C_1}^\top)C_1. \tag{115}$$

1. We now consider the first term, replacing $\sigma'(Z_1(0)) \approx a$ and substituting Δ_1 with 52 yields the following approximation:

$$\begin{aligned}-2\sqrt{2}g_1\Xi\Delta_1(0)^\top C_0 &= -4ag_1\Xi\Delta_2(0)^\top W_2(2C_0 - I)C_0 &(116)\\ &\quad - 4g_1\Xi\Big(\big[(2C_0 - I)W_2^\top\Delta_2(0)\big] \odot \big[\sigma'(Z_1(0)) - a\mathbf{1}\big]\Big)^\top C_0. &(117)\end{aligned}$$

Since $|\sigma''(\cdot)| \le \mu$ and by 101, $\|Z_1(0)\|_{\max} \le \epsilon_\rho$,

$$\|\sigma'(Z_1(0)) - \sigma'(0)\|_{\max} = \|\sigma'(Z_1(0)) - a\mathbf{1}\|_{\max} \le \mu\epsilon_\rho, \tag{118}$$

The approximation error 117 can be bounded as

$$\begin{aligned}&\left\|4g_1\Xi\Big(\big[(2C_0 - I)W_2^\top\Delta_2(0)\big] \odot \big[\sigma'(Z_1(0)) - a\mathbf{1}\big]\Big)^\top C_0\right\|_2 &(119)\\ &\le 4g_1\|\Xi\|_2\left\|\big[(2C_0 - I)W_2^\top\Delta_2(0)\big] \odot \big[\sigma'(Z_1(0)) - a\mathbf{1}\big]\right\|_F\|C_0\|_2 &(120)\\ &\le 4g_1\sqrt{\nu}\|\sigma'(Z_1(0)) - a\mathbf{1}\|_{\max}\|(2C_0 - I)W_2^\top\Delta_2(0)\|_F &(121)\\ &\le 4g_1\sqrt{\nu}\mu\epsilon_\rho\|W_2\|_2v_\rho. &(122)\end{aligned}$$

2. Now, recall

$$\Delta_2(0) = \frac{Z_2(0) - T}{N}. \tag{123}$$

Substituting $\Delta_2(0)$ into the leading term in 116, yields

$$-4ag_1\Xi\Delta_2(0)^\top W_2(2C_0 - I)C_0 = M(2C_0 - I)C_0 \tag{124}$$

$$- \frac{4ag_1}{N}\Xi Z_2(0)^\top W_2(2C_0 - I)C_0, \tag{125}$$

where $M := \frac{4ag_1}{N}\Xi T^\top W_2$.

Expression 125 satisfies

$$\left\| \frac{4ag_1}{N}\Xi Z_2(0)^\top W_2(2C_0 - I)C_0 \right\|_2 \leq \frac{4ag_1\sqrt{\nu}\|W_2\|_2}{N}\|Z_2(0)\|_2 \tag{126}$$

$$\leq \frac{4\sqrt{2}ag\sqrt{\nu}\|W_2\|_2^2}{N}u_\rho. \tag{127}$$

3. For expression 124, the following equation holds

$$\|(2C_0 - I)C_0 - I\|_2 = h_\rho, \quad h_\rho := \frac{3\rho^2 + \rho^4}{(1+\rho^2)^2}. \tag{128}$$

Therefore,

$$\|M(2C_0 - I)C_0 - M\|_2 \leq h_\rho\|M\|_2. \tag{129}$$

4. Now, we consider the second part of the initial gradient $Y_1C_1(J_{C_1} + J_{C_1}^\top)C_1$ in 115. By 53

$$J_C(0) = 2\sqrt{2}W_2^\top\Delta_2(0)H_1(0)^\top - 2\sqrt{2}g_1\Delta_1(0)\Xi^\top Y_1(0). \tag{130}$$

By 52

$$\Delta_1(0) = \sqrt{2}(2C_0 - I)W_2^\top\Delta_2(0) \odot \sigma'(Z_1(0)). \tag{131}$$

Therefore,

$$\|\Delta_1(0)\|_F \leq \sqrt{2}\|\sigma'(Z_1(0))\|_{\max}\|(2C_0 - I_m)W_2^\top\Delta_2(0)\|_F \tag{132}$$

$$\leq \sqrt{2}\|2C_0 - I\|_2\|W_2\|_2\|\Delta_2(0)\|_F \tag{133}$$

$$\leq \sqrt{2}\|W_2\|_2 v_\rho \tag{134}$$

together with

$$\|\Xi^\top Y_1(0)\|_2 \leq \sqrt{\nu}\rho, \tag{135}$$

yields

$$\|J_C(0)\|_F \leq 2\sqrt{2}\|W_2\|_2 v_\rho u_\rho + 2\sqrt{2}g_1\|\Delta_1(0)\|_F\sqrt{\nu}\rho \tag{136}$$

$$\leq 2\sqrt{2}\|W_2\|_2 v_\rho u_\rho + 4g_1\sqrt{\nu}\rho\|W_2\|_2 v_\rho. \tag{137}$$

Moreover,

$$\|Y_1(0)C_0\|_2 = \rho c_\rho \leq \rho, \quad \|C_0\|_2 \leq 1. \tag{138}$$

Therefore,

$$\left\|Y_1(0)C_0\big(J_C(0) + J_C(0)^\top\big)C_0\right\|_2 \leq 2\rho\|J_C(0)\|_F \tag{139}$$

$$\leq 4\rho\|W_2\|_2 v_\rho\left(\sqrt{2}u_\rho + 2g_1\sqrt{\nu}\rho\right). \tag{140}$$

Combining the **(1-4)**, the total approximation error is bounded by

$$\|G_1(0) - M\|_2 \leq r_\rho, \tag{141}$$

where

$$r_\rho := 4g_1\sqrt{\nu}\|W_2\|_2\mu\epsilon_\rho v_\rho + \frac{4\sqrt{2}ag_1\sqrt{\nu}\|W_2\|_2^2}{N}u_\rho \tag{142}$$

$$+ h_\rho\|M\|_2 + 4\rho\|W_2\|_2 v_\rho\left(\sqrt{2}u_\rho + 2g_1\sqrt{\nu}\rho\right). \tag{143}$$

We now use this spectral-norm perturbation bound for the singular-value spectrum by Weyl's inequality (Weyl, 1949),

$$|s_i(G_1(0)) - s_i(M)| \le r_\rho, \quad 1 \le i \le d. \tag{144}$$

Furthermore,

$$\operatorname{rank}(M) = \operatorname{rank}\left(\frac{4ag_1}{N}\Xi T^\top W_2\right) \le K. \tag{145}$$

Thus

$$s_i(M) = 0, \quad K < i \le d. \tag{146}$$

Therefore,

$$\begin{cases} |s_i(G_1(0)) - s_i(M)| \le r_\rho, & 1 \le i \le K, \\ s_i(G_1(0)) \le r_\rho, & K < i \le d. \end{cases} \tag{147}$$

Finally, since

$$\epsilon_\rho = O(\rho), \quad u_\rho = O(\rho), \quad v_\rho = O(1), \quad h_\rho = O(\rho^2). \tag{148}$$

Therefore, $r_\rho = O(\rho)$. □

A.11.1 Gradient singular subspaces during training

In this section, we discuss the gradient subspaces changes during training. Let $G_1(t) = L_1(t)\Sigma_1(t)R_1(t)$ denote the SVD of the gradient on Y_1, where $L_{1,1}(t) \in \mathbb{R}^{d\times K}$, $\Sigma_{1,1}(t) \in \mathbb{R}^{K\times K}$, $R_{1,1}(t) \in \mathbb{R}^{m\times K}$ denote the top-K component and $L_{1,2}(t) \in \mathbb{R}^{d\times(d-K)}$, $\Sigma_{1,2}(t) \in \mathbb{R}^{(d-K)\times(d-K)}$, $R_{1,2}(t) \in \mathbb{R}^{m\times(d-K)}$ denote the bottom $d-K$ components. In the following Lemma, we characterize the subspaces changes during training.

Lemma A.7. *Assume Assumptions 3.1, 3.2, 3.3 and 3.4 hold, let r_ρ be defined as 142 and choose a constant c_3 so that $r_\rho \le c_3 s_K(M)$, and define*

$$V(t) := \frac{\eta c_1 \|G_1(0)\|_F}{1-\lambda}(1-\lambda^t), \quad q(t) := s_K(M) - r_\rho - s_{K+1}(G_1(t)). \tag{149}$$

then for $0 \le t < H$,

$$\|L_{1,1}(t)^\top L_{1,2}(0)\|_F \le \frac{L_G V(t)}{q(t)}, \quad \|R_{1,1}(t)^\top R_{1,2}(0)\|_F \le \frac{L_G V(t)}{q(t)}. \tag{150}$$

Proof. Follow Assumption 3.3, we train the network using GD, therefore,

$$Y_1(t) - Y_1(0) = -\eta \sum_{\tau=0}^{t-1} G_1(\tau), \tag{151}$$

$$\|Y_1(t) - Y_1(0)\|_F \le \eta \sum_{\tau=0}^{t-1} G_1(\tau) \le \eta c_1 \|G_1(0)\|_F \sum_{\tau=0}^{t-1} \lambda^\tau = V(t), \tag{152}$$

where the last inequality follow the Assumption 3.4. By Lemma A.5, we have

$$\|G_1(Y_1) - G_1(Y_1')\|_F \le L_G \|Y_1 - Y_1'\|_F, \tag{153}$$

therefore,

$$\|G_1(Y_1(t)) - G_1(Y_1(0))\|_F \le L_G \|Y_1(t) - Y_1(0)\|_F \le L_G V(t). \tag{154}$$

Follow 144, we have

$$s_K(G_1(0)) \ge s_K(M) - r_\rho, \tag{155}$$

Therefore,

$$s_K(G_1(0)) - s_{K+1}(G_1(t)) \ge s_K(M) - r_\rho - s_{K+1}(G_1(t)) = q(t). \tag{156}$$

By Assumption 3.4, we have

$$s_K(M) - s_{K+1}(G_1(t)) \ge c_3 s_K(M), \tag{157}$$

thus,

$$q(t) \ge c_3 s_K(M) - r_\rho \ge 0. \tag{158}$$

By Wedin's Sin Theorem Wedin, 1972 (Lemma D.5 (Xu et al., 2026)), we have

$$\|L_{1,1}(t)^\top L_{1,2}(0)\|_F \le \frac{\|G_1(t) - G_1(0)\|_F}{s_K(G_1(0)) - s_{K+1}(G_1(t))} = \frac{L_G V(t)}{q(t)}, \tag{159}$$

$$\|R_{1,1}(t)^\top R_{1,2}(0)\|_F \le \frac{\|G_1(t) - G_1(0)\|_F}{s_K(G_1(0)) - s_{K+1}(G_1(t))} = \frac{L_G V(t)}{q(t)}. \tag{160}$$

□

The previous lemma shows the dominant singular subspaces of $G_1(t)$ stay close to the initialization. Next, we show that gradient leakage into the non-dominant directions remains small, which implies that the parameter updates are largely concentrated in a low-dimensional subspace during training.

Lemma A.8. *Assume assumptions 3.1, 3.2, 3.3 and 3.4 hold and let $r_\rho, V(t)$, and $q(t)$ be defined as in Lemmas A.6 and A.7. Define*

$$A(t) := \max\left\{\left\|\sin\Phi\big(L_{1,1}(t), L_{1,1}(0)\big)\right\|_2, \left\|\sin\Phi\big(R_{1,1}(t), R_{1,1}(0)\big)\right\|_2\right\}, \tag{161}$$

$$\delta(t) := \frac{L_G V(t)}{q(t)}. \tag{162}$$

There exists semi-orthogonal matrices

$$L_s \in \mathbb{R}^{d\times(d-2K)}, \quad R_s \in \mathbb{R}^{m\times(d-2K)} \tag{163}$$

such that

$$L_s^\top L_s = I_p, \qquad R_s^\top R_s = I_p, \tag{164}$$

$$Y_1(0)R_s = \rho L_s, \quad Y_1(0)^\top L_s = \rho R_s, \tag{165}$$

$$L_s^\top L_{1,1}(0) = 0, \qquad R_s^\top R_{1,1}(0) = 0 \tag{166}$$

Define

$$e(t) := \sqrt{s_1^2(G_1(t))\frac{A^2(t)}{d-2K} + s_{K+1}^2(G_1(t))} \tag{167}$$

Then, for all $0 \le t < H$,

$$\max\left\{\frac{\|Y_1(t+1)R_s - Y_1(t)R_s\|_F}{\sqrt{d-2K}}, \frac{\|Y_1(t+1)^\top L_s - Y_1(t)^\top L_s\|_F}{\sqrt{d-2K}}\right\} \le \eta e(t). \tag{168}$$

where

$$e(t) \le \begin{cases} r_\rho, & t = 0, \\ \sqrt{(c_1c_2)^2\lambda^{2t}\big(s_1(M) + r_\rho\big)^2\frac{\delta^2(t)}{d-2K} + s_{K+1}^2(G_1(t))}, & t \ge 1. \end{cases} \tag{169}$$

Proof. Recall that $Y_1(0) = \rho Q$ with $QQ^\top = I_d$. Let

$$\mathcal{S} := \mathcal{R}\big(L_{1,2}(0)\big) \cap \mathcal{R}^\perp\big(QR_{1,1}(0)\big). \tag{170}$$

Since both subspaces have dimension at least $d - K$,

$$\dim(\mathcal{S}) \ge d - 2K. \tag{171}$$

Let $L_s \in \mathbb{R}^{d\times(d-2K)}$ be an orthonormal basis for any $d-2K$-dimensional subspace of $\mathcal{S}$, and let

$$R_s := Q^\top L_s. \tag{172}$$

Since $QQ^\top = I_d$, $R_s^\top R_s = I_p$, and $Y_1(0)R_s = \rho L_s, Y_1(0)^\top L_s = \rho R_s$. By construction,

$$L_s^\top L_{1,1}(0) = 0, \qquad R_s^\top R_{1,1}(0) = 0. \tag{173}$$

Since $L_{1,2}(0)$ is the full orthogonal complement of $L_{1,1}(0)$ in $\mathbb{R}^d$, follow Lemma D.3 in (Xu et al., 2026) gives

$$\|L_s^\top L_{1,1}(t)\|_F \le \|L_{1,2}(0)^\top L_{1,1}(t)\|_F = \left\|\sin\Phi\big(L_{1,1}(t), L_{1,1}(0)\big)\right\|_2 \le A(t), \tag{174}$$

and similarly

$$\|R_{1,1}(t)^\top R_s\|_F \le \|R_{1,1}(t)^\top R_{1,2}(0)\| = \left\|\sin\Phi\big(R_{1,1}(t), R_{1,1}(0)\big)\right\|_2 \le A(t). \tag{175}$$

Then consider the SVD of the gradient

$$G_1(t) = L_{1,1}(t)s_{1,1}(t)R_{1,1}(t)^\top + L_{1,2}(t)s_{1,2}(t)R_{1,2}(t)^\top, \tag{176}$$

Thus,

$$\begin{aligned}\|G_1(t)R_s\|_F^2 &\le s_1^2(G_1(t))\|R_{1,1}(t)^\top R_s\|_F^2 + s_{K+1}^2(G_1(t))\|R_{1,2}^\top R_s\|_F^2 \\ &\le s_1^2(G_1(t))\|R_{1,1}(t)^\top R_s\|_F^2 + s_{K+1}^2(G_1(t))\|R_s\|_F^2 \\ &\le s_1^2(G_1(t))A^2(t) + (d-2K)s_{K+1}^2(G_1(t)).\end{aligned} \tag{177}$$

Similarly,

$$\|L_s^\top G_1(t)\|_F^2 \le s_1^2(G_1(t))A^2(t) + (d-2K)s_{K+1}^2(G_1(t)). \tag{178}$$

Dividing by $d-2K$ for both sides and let

$$e(t) := \sqrt{s_1^2(G_1(t))\frac{A^2(t)}{d-2K} + s_{K+1}^2(G_1(t))} \tag{179}$$

Therefore,

$$\max\left\{\frac{\|L_s^\top G_1(t)\|_F}{\sqrt{d-2K}}, \frac{\|G_1(t)R_s\|_F}{\sqrt{d-2K}}\right\} \le e(t). \tag{180}$$

Now, consider the GD update

$$Y_1(t+1) - Y_1(t) = -\eta G_1(t) \tag{181}$$

Right multiply R_s yields

$$Y_1(t+1)R_s - Y_1(t)R_s = -\eta G_1(t)R_s. \tag{182}$$

Therefore,

$$\frac{\|Y_1(t+1)R_s - Y_1(t)R_s\|_F}{\sqrt{d-2k}} = \eta\frac{\|G_1(t)R_s\|_F}{\sqrt{d-2K}} \le \eta e(t). \tag{183}$$

Thus, for all $0 \le t < H$,

$$\max\left\{\frac{\|Y_1(t+1)R_s - Y_1(t)R_s\|_F}{\sqrt{d-2K}}, \frac{\|Y_1(t+1)^\top L_s - Y_1(t)^\top L_s\|_F}{\sqrt{d-2K}}\right\} \le \eta e(t). \tag{184}$$

For $e(t)$, at $t=0$, $A(0)=0$, therefore

$$e(t) = \sqrt{s_1^2(G_1(0))\frac{A^2(0)}{d-2K} + s_{K+1}^2(G_1(0))} = s_{K+1}(G_1(0)) \le r_\rho. \tag{185}$$

where the last inequality follows lemma A.6.

For $t \ge 1$, by Assumption 3.4,

$$\frac{{}_i(G_1(t))}{{}_i(G_1(0))} \le c_2\frac{\|G_1(t)\|_F}{\|G_1(0)\|_F}, \quad i \le K. \tag{186}$$

Let $i=1$, we have

$$s_1(G_1(t)) \le c_2\frac{\|G_1(t)\|_F}{\|G_1(0)\|_F}s_1(G_1(0)) \le c_2c_1\lambda^t s_1(G_1(0)). \tag{187}$$

By 144, we have

$$s_1(G_1(0)) \le s_1(M) + r_\rho, \tag{188}$$

therefore,

$$s_1(G_1(t)) \le c_2c_1\lambda^t(s_1(G_1(M) + r_\rho). \tag{189}$$

Together with lemma A.7 $A(t) \le \delta(t)$,

$$e^2(t) = \frac{s_1^2(G_1(t))A^2(t)}{p} + s_{K+1}^2(G_1(t)) \tag{190}$$

$$\le (c_1c_2)^2\lambda^{2t}\big(s_1(M)+r_\rho\big)^2\frac{\delta^2(t)}{d-2K} + s_{K+1}^2(G_1(t)). \tag{191}$$

Taking square roots yields,

$$e(t) \le \sqrt{(c_1c_2)^2\lambda^{2t}\big(s_1(M)+r_\rho\big)^2\frac{\delta^2(t)}{p} + s_{K+1}^2(G_1(t))}, \quad t \ge 1. \tag{192}$$

□

A.12 LOW-RANK TRAINING DYNAMICS

In this section, we build upon previous lemmas and show that in the two-layer Sandwich network, the updates of the trainable parameter Y_1 lie in the subspaces.

Theorem A.9. *Assume assumptions 3.1, 3.2, 3.3 and 3.4 hold. Define*

$$M := \frac{4ag_1}{N}\Xi T^\top W_2, \quad u_\rho := a\epsilon_\rho, \quad \epsilon_\rho := 2\sqrt{2}g_1\rho c_\rho\sqrt{\nu}, \quad c_\rho := \frac{1}{1+\rho^2}, \tag{193}$$

$$v_\rho := \frac{\sqrt{2}\|W_2\|_2\epsilon_\rho\sqrt{d} + \|T\|_F}{N}, \quad h_\rho := \frac{3\rho^2+\rho^4}{(1+\rho^2)^2}, \tag{194}$$

$$r_\rho := 4g_1\sqrt{\nu}\|W_2\|_2\mu\epsilon_\rho v_\rho + \frac{4\sqrt{2}ag_1\sqrt{\nu}\|W_2\|_2^2}{N}u_\rho \tag{195}$$

$$+ h_\rho\|M\|_2 + 4\rho\|W_2\|_2 v_\rho\left(\sqrt{2}u_\rho + 2g_1\sqrt{\nu}\rho\right), \tag{196}$$

$$L_G := \frac{J_F^2 + E_F L_F}{N}, \quad E_F := 2g_1\|W_2\|_2\sqrt{vd} + \|T\|_F, \quad J_F := 8g_1\|W_2\|_2\sqrt{vd}, \tag{197}$$

$$L_F := 32g_1\|W_2\|_2\sqrt{\nu d} + 8\sqrt{2}\mu g_1^2\|W_2\|_2\nu\sqrt{d}. \tag{198}$$

For $0 \le t < H$, define

$$A(t) := \max\left\{\left\|\sin\Phi\big(L_{1,1}(t), L_{1,1}(0)\big)\right\|_2, \left\|\sin\Phi\big(R_{1,1}(t), R_{1,1}(0)\big)\right\|_2\right\}, \tag{199}$$

$$\delta(t) := \frac{L_G V(t)}{q(t)},\ V(t) := \frac{\eta c_1\|G_1(0)\|_F}{1-\lambda}(1-\lambda^t),\ q(t) := s_K(M) - r_\rho - s_{K+1}(G_1(t)). \tag{200}$$

and let,

$$Y_1(0) = \rho Q, \quad Q \in \mathbb{R}^{d\times m}, \quad QQ^\top = I_d, \tag{201}$$

then there exist orthogonal matrices $L \in \mathbb{R}^{d\times d}$, $R \in \mathbb{R}^{m\times m}$ so that for every $0 \le t < H$, Y_1 can be decomposed as:

$$Y_1(t) = L\widetilde{Y}_1(t)R^\top = L\begin{bmatrix}\widetilde{Y}_{11}(t) & \widetilde{Y}_{12}(t)\\ \widetilde{Y}_{13}(t) & \widetilde{Y}_{14}(t)\end{bmatrix}R^\top. \tag{202}$$

where $\widetilde{Y}_{12}(t) \in \mathbb{R}^{2K\times(d-2K)}$, $\widetilde{Y}_{13}(t) \in \mathbb{R}^{(d-2K)\times(m-(d-2K))}$, $\widetilde{Y}_{14}(t) \in \mathbb{R}^{(d-2K)\times(d-2K)}$, with

$$\widetilde{Y}_{12}(0) = 0, \quad \widetilde{Y}_{13}(0) = 0, \quad \widetilde{Y}_{14}(0) = \rho I_p. \tag{203}$$

and the GD updates satisfies the following inequality for $j \in \{2,3,4\}$

$$\frac{\|\widetilde{Y}_{1,j}(t+1) - \widetilde{Y}_{1,j}(t)\|_F}{\sqrt{d-2K}} \le \eta e(t). \tag{204}$$

with

$$e(t) := \sqrt{s_1^2(G_1(t))\frac{A^2(t)}{d-2K} + s_{K+1}^2(G_1(t))} \tag{205}$$

$$\le \begin{cases} r_\rho, & t = 0,\\ \sqrt{(c_1c_2)^2\lambda^{2t}\big(s_1(M)+r_\rho\big)^2\frac{\delta^2(t)}{d-2K} + s_{K+1}^2(G_1(t))}, & t \ge 1.\end{cases} \tag{206}$$

Proof. Define

$$\mathcal{S} := \mathcal{R}\big(L_{1,2}(0)\big) \cap \mathcal{R}^\perp\big(QR_{1,1}(0)\big), \tag{207}$$

by 171, $\operatorname{rank}(\mathcal{S}) \ge d - 2K$. Choose $L_s \in \mathbb{R}^{d\times(d-2K)}$ as an orthonormal basis of any $(d-2k)-$dimensional subspace of $\mathcal{S}$ and let $R_s := Q^\top L_s$. By lemma A.8,

$$Y_1(0)R_s = \rho L_s, \quad Y_1(0)^\top L_s = \rho R_s. \tag{208}$$

Moreover, by definition of $\mathcal{S}$ we have

$$L_s^\top L_{1,1}(0) = 0, \quad R_s^\top R_{1,1}(0) = 0. \tag{209}$$

Then, let

$$L = [L_a \quad L_s] \in \mathbb{R}^{d\times d}, \qquad R = [R_a \quad R_s] \in \mathbb{R}^{m\times m}, \tag{210}$$

where

$$L_a \in \mathbb{R}^{d\times 2K}, \qquad R_a \in \mathbb{R}^{m\times(m-(d-2K))}. \tag{211}$$

The block decomposition of $Y_1(t)$, therefore, is

$$\widetilde{Y}_{11}(t) = L_a^\top Y_1(t) R_a, \qquad \widetilde{Y}_{12}(t) = L_a^\top Y_1(t) R_s, \tag{212}$$

$$\widetilde{Y}_{13}(t) = L_s^\top Y_1(t) R_a, \qquad \widetilde{Y}_{14}(t) = L_s^\top Y_1(t) R_s. \tag{213}$$

Thus, by 208 and 209,

$$\widetilde{Y}_{12}(0) = L_a^\top Y_1(0) R_s = \rho L_a^\top L_s = 0, \qquad \widetilde{Y}_{13}(0) = L_s^\top Y_1(0) R_a = \rho R_s^\top R_a = 0, \tag{214}$$

$$\widetilde{Y}_{14}(0) = L_s^\top Y_1(0) R_s = \rho I_p. \tag{215}$$

Under GD, we have

$$\widetilde{Y}_{12}(t+1) - \widetilde{Y}_{12}(t) = L_a^\top \Delta Y_1(t) R_s \implies \|\widetilde{Y}_{12}(t+1) - \widetilde{Y}_{12}(t)\|_F \le \|\Delta Y_1(t) R_s\|_F. \tag{216}$$

similarly,

$$\|\widetilde{Y}_{13}(t+1) - \widetilde{Y}_{13}(t)\|_F \le \|\Delta Y_1(t)^\top L_s\|_F, \tag{217}$$

$$\|\widetilde{Y}_{14}(t+1) - \widetilde{Y}_{14}(t)\|_F \le \min\left\{\|\Delta Y_1(t) R_s\|_F, \|\Delta Y_1(t)^\top L_s\|_F\right\}. \tag{218}$$

By lemma A.8, we have

$$\max\left\{\frac{\|\Delta Y_1(t) R_s\|_F}{\sqrt{d-2K}}, \frac{\|\Delta Y_1(t)^\top L_s\|_F}{\sqrt{d-2K}}\right\} \le \eta e(t). \tag{219}$$

Therefore,

$$\frac{\|\widetilde{Y}_{1,j}(t+1) - \widetilde{Y}_{1,j}(t)\|_F}{\sqrt{d-2K}} \le \eta e(t), \quad \text{for} \quad j \in \{2,3,4\} \tag{220}$$

Follow lemma A.8, yields the bounds on $e(t)$. Since $L_a \in \mathbb{R}^{d\times 2K}$, $L_a \widetilde{Y}_{11}(t) R_a^\top$ has rank at most $2K$. □