HOW CAN DEEP LEARNING PERFORMS DEEP (HIERARCHICAL) LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning is also known as hierarchical learning, where the learner *learns* to represent a complex target function by decomposing it into a sequence of simpler functions to reduce sample and time complexity. This paper formally analyzes how multi-layer neural networks can perform such hierarchical learning *efficiently* and *automatically* by applying stochastic gradient descent (SGD) or its variants.

On the conceptual side, we present a characterizations of how certain deep (i.e. super-constantly many layers) neural networks can still be sample and time efficiently trained on hierarchical learning tasks, when no known existing algorithm (including layer-wise training, kernel method, etc) is efficient. We establish a new principle called "backward feature correction", where *the errors in the lower-level features can be automatically corrected when training together with the higher-level layers*. We believe this is a key behind how deep learning is performing deep (hierarchical) learning, as opposed to layer-wise learning or simulating some known non-hierarchical method. [1]

1 INTRODUCTION

Deep learning is also known as hierarchical (feature) learning.[2] The term hierarchical learning can be defined as learning to represent the *complex* target function $g(x)$ using a composition of *much simpler* functions: $g(x) = h_L(h_{L-1}(\cdots h_1(x)\cdots))$. In deep learning, for example, each $h_\ell(\cdot)$ is usually represented by a linear operator followed with activation function. Empirically, the training process of deep learning is done by stochastic gradient descent (SGD) or its variants. After training, one can verify that the complexity of the learned *features* (i.e., $h_\ell(h_{\ell-1}(\cdots x \cdots))$ indeed increases as ℓ goes deeper — see (Zeiler & Fergus, 2014) or Figure 2. It has also been discovered for a long time that hierarchical learning, in many applications, requires fewer training examples (Bouvrie, 2009) when compared with non-hierarchical methods that learn $g(x)$ in one shot.

Hierarchical learning from a theoretical perspective. It is well-known that neural networks can *represent* a wide range of complicated functions using the composition of much simpler layers. Instead of learning a degree 2^L function from scratch, can hierarchical learning learn to represent it as a composition of L-quadratic functions, and thus learning one quadratic function at a time? The **main difficulty** is that being able to *represent* a complex target function in a hierarchical network does not necessarily guarantee *efficient* learning. For example, L layers of quadratic networks can represent all parity functions up to degree 2^L; but in the deep $L = \omega(1)$ setting, it is unclear if one can learn parity functions over $x \in \{-1, 1\}^d$ with noisy labels via any efficient $\mathsf{poly}(d)$-time algorithm (Feldman et al., 2006), not to say via training neural networks.

So, for what type of functions can we formally *prove* that deep neural networks can hierarchically learn them? And, *how* can deep learning perform hierarchical learning to greatly improve learning efficiency in these cases?

[1]Note: although this paper has been circulated for a while, this is our first time to submit to an ML venue. This is a theory paper but we tried to make it suitable for a wider audience. We included many figures to support the connection between our theory and practice, but due to space limitation, many of the figures are deferred to Appendix A starting on Page 14.

[2]Quoting Bengio (2009), "deep learning aim at *learning feature hierarchies* with features from higher levels of the hierarchy formed by the composition of lower level features." Quoting Goodfellow et al. (2016) "the hierarchy of concepts allows the computer to learn complicated concepts by *building them out of simpler ones*."

Hierarchical learning and layer-wise learning. Motivated by the large body of theory works for two-layer networks, a tentative approach to analyze hierarchical learning in deep learning is via layer-wise training. Consider the example of using a multi-layer network with quadratic activation, to learn the following target function.

$$g(x) = \underbrace{x_1^2 + 2x_2^2}_{\text{low-complexity signal}} + 0.1\underbrace{(x_1^2 + 2x_2^2 + x_3)^2}_{\text{high-complexity signal}} . \tag{1.1}$$

In this example, one may hope that the first train a two-layer quadratic network to learn simple, quadratic features (x_1^2, x_2^2), and the train another two-layer quadratic network *on top of the first one* learns a quadratic function over (x_1^2, x_2^2, x_3). In this way, one can hope for never needing to learn a degree-4 polynomial in one shot, but simply learning two quadratic functions in two steps. Is hierarchical learning in deep learning really this simple?

In fact, layer-wise training is known to perform poorly in practical deep learning, see Figure 7. A common sense is that when we train lower-level layers, it might over-fit to higher-level features. Using the example of (1.1), if one uses a quadratic network to fit $g(x)$, then the first-layer features may be trained too greedily and over-fit to high-complexity signals: for instance, the best quadratic network to fit $g(x)$ may learn features $(x_1 + \sqrt{0.1}x_3)^2$ and x_2^2, instead of (x_1^2, x_2^2). Now, if we freeze the first layer and train a second layer quadratic network on top of it (and the input), this "error" of $\sqrt{0.1}x_3$ can no longer be fixed thus we cannot fit the target function perfectly.

Our main message. On the conceptual level, we show (both theoretically and *empirically*) although lower-level layers in a neural network indeed tend to over-fit to higher complexity signals at the beginning of training, when training all the layers together — using simple variants of SGD — the presence of higher-level layers can eventually help reduce this type of over-fitting in lower-level layers. For example, in the above case the quality of lower-level features can improve from $(x_1 + \sqrt{0.1}x_3)^2$ again to get closer and closer to x_1^2 when trained together with higher-level layers. We call this *backward feature correction*. More generally, we identify *two critical steps* in the hierarchical learning process of a multi-layer network.

- The ***forward feature learning*** step, where a higher-level layer can learn its features using the simple combinations of the learned features from lower-level layers. This is an analog of layer-wise training, but a bit different (see discussions in (Allen-Zhu & Li, 2019a)) since all the layers are still trained *simultaneously*.
- The ***backward feature correction*** step, where a lower-level layer can learn to further *improve* its feature quality with the help of the learned features in higher-level layers. We are not aware of this being recorded in the theory literature, and believe it is *a most critical reason* for why hierarchical learning goes *beyond* layer-wise training in deep learning. We shall mathematically characterize this in Theorem 2.

Remark. When all the layers of a neural network are trained together, the aforementioned two steps actually occur *simultaneously*. For interested readers, we also design experiments to separate them and visualize, see Figure 3, 5, and 10 in Section A. On the theoretical side, we also give toy examples with mathematical intuitions in Section 1.2 to further explain the two steps.

Our technical results. With the help of the discovered conceptual message, we show the following technical results. Let input dimension d be sufficiently large, there exist a non-trivial class of "well-conditioned" L-layer neural networks with $L = \omega(1)$ and quadratic activations[3] so that:

- Training such networks by a variant of SGD *efficiently* and *hierarchically* learns this concept class. Here, by "efficiently" we mean time/sample complexity is $\mathsf{poly}(d/\varepsilon)$ where ε is the generalization error; and by "hierarchically" we mean the network learns to represent the concept class by decomposing it into a composition of simple (i.e. quadratic) functions, via forward feature learning and backward feature correction, to significantly reduce sample/time complexity.
- We are unaware of existing algorithm that can achieve the same result in polynomial time. For completeness, we prove super-polynomial lower bounds for shallow learning methods such as (1) kernel method, (2) regression over feature mappings, (3) two-layer networks with degree

[3]It is easy to measure the network's growing representation power in depth using quadratic activations (Livni et al., 2014). As a separate note, quadratic networks can perform as well as ReLU networks in practice (see Figure 11 on Page 25), and has particular cryptographic advantage (Mishra et al., 2020).

$\leq 2^L$ activations, or (4) the previous three with any regularization. Although proving separation is *not our main message*, we still illustrate in Section 1.2 that neither do we believe layer-wise training, or applying kernel method multiple (even $\omega(1)$ many) times can achieve poly-time.

To this extent, we have shown, at least for this class of L-layer networks with $L = \omega(1)$, *deep learning can indeed perform efficient hierarchical learning* when trained by a variant of SGD to learn functions not known to be learnable by "shallow learners" (including layer-wise training which can be viewed as applying two-layer networks multiple times). Thus, we believe that hierarchical learning (especially with backward feature correction) is *critical* to learn this concept class.

Difference from existing theory. Many prior works have studied the theory of deep learning. We try to cover them all in Section F but we summarize our difference from them as follows.

- Starting from Jacot et al. (2018), there is a rich literature that reduces multi-layer neural networks to kernel methods (e.g. neural tangent kernels, or NTKs). They approximate neural networks by *linear models* over (hierarchically defined) random features — which are not *learned* through training. They do not show the power of deep learning beyond kernel methods.
- Many other theories focus on two-layer networks but they do not have the *deep hierarchical* structure. In particular, some have studied *feature learning* as a process (Daniely & Malach, 2020; Li et al., 2020; Allen-Zhu & Li, 2021), but still cannot cover how the features of the second layer can help backward correct the first layer; thus naively repeating them for multi-layer networks may only give rise to layer-wise training as opposed to the full hierarchical learning.
- Allen-Zhu et al. (2019a) shows that 3-layer neural networks can learn the so-called "second-order NTK," which is not a linear model; however, second-order NTK is also learnable by doing a nuclear-norm constrained linear regression, which is still not truly hierarchical.
- Allen-Zhu & Li (2019a) shows that 3-layer ResNet can learn a concept class otherwise not learnable by kernel methods (within the same level of sample complexity). We discuss more in Section F, but most importantly, that concept class is learnable by applying kernel method twice.

In sum, prior works may have only studied a simpler but already non-trivial question: "can multi-layer neural networks efficiently learn simple functions that are *already learnable* by non-hierarchical models." While the cited works shed great light on the learning process of neural networks, in the language of this paper, they cannot justify how deep learning performs *deep hierarchical feature learning*. Our work is motivated by this huge gap between theory and practice.

Admittedly, with a more ambitious goal we have to sacrifice something. Notably, we study quadratic activations while some cited works can handle ReLU. Note this may be still fine: in practice, deep learning with quadratic networks perform very closely to ReLU ones, and significantly better than two-layer networks or neural kernel methods (see Figure 11). Hence, our theoretical result may also serve as a provisional step towards understating the deep learning process in ReLU networks.

1.1 Our Theorem

We give an overview of our theoretical result. The learner networks we consider are like DenseNets:

$$G(x) = \sum_{\ell=2}^{L} \langle u_\ell, G_\ell(x) \rangle \in \mathbb{R} \qquad \text{where} \quad G_0(x) = x \in \mathbb{R}^d, \qquad G_1(x) = \sigma(x) - \mathbb{E}[\sigma(x)] \in \mathbb{R}^d$$
$$G_\ell(x) = \sigma\left(\sum_{j \in \mathcal{J}_\ell} \mathbf{M}_{\ell,j} G_j(x)\right) \qquad \text{for } \ell \geq 2 \text{ and } \mathcal{J}_\ell \subseteq \{0, 1, \cdots, \ell-1\} \tag{1.2}$$

Here, σ is the activation function and we pick $\sigma(z) = z^2$ in this paper, $\mathbf{M}_{\ell,j}$'s are weight matrices, and the final output $G(x) \in \mathbb{R}$ is a weighted summation of the outputs of all the layers. The set $\mathcal{J}_\ell$ defines the connection graph. We can handle *any* connection graph with the only restriction being there is at least one "skip link."[4] To illustrate the main idea, we focus here on a regression problem in the teacher-student setting, although our result applies to classification as well as the *agnostic learning setting* (where the target network may also have label error). In this teacher-student regression setting, the goal is to learn some unknown target function $G^\star(x)$ in some concept class given samples $(x, G^\star(x))$ where $x \sim \mathcal{D}$ follows some distribution $\mathcal{D}$. In this paper, we consider

[4]In symbols, for every $\ell \geq 3$, we require $(\ell-1) \in \mathcal{J}_\ell$, $(\ell-2) \notin \mathcal{J}_\ell$ but $j \in \mathcal{J}_\ell$ for some $j \leq \ell - 3$. As comparisons, the vanilla feed-forward network corresponds to $\mathcal{J}_\ell = \{\ell - 1\}$, while ResNet (He et al., 2016) (with skip connection) corresponds to $\mathcal{J}_\ell = \{\ell-1, \ell-3\}$ with weight sharing (namely, $\mathbf{M}_{\ell,\ell-1} = \mathbf{M}_{\ell,\ell-3}$).

the target functions $G^\star(x) \in \mathbb{R}$ coming from the *same class* as the learner network:

$$G^\star(x) = \sum_{\ell=2}^L \alpha_\ell \cdot \langle u_\ell^\star, G_\ell^\star(x) \rangle \in \mathbb{R} \qquad \text{where} \quad G_0^\star(x) = x \in \mathbb{R}^d, \qquad G_1^\star(x) = \sigma(x) - \mathbb{E}[\sigma(x)] \in \mathbb{R}^d$$
$$G_\ell^\star(x) = \sigma\left(\sum_{j \in \mathcal{J}_\ell} \mathbf{W}_{\ell,j}^\star G_j^\star(x)\right) \in \mathbb{R}^{k_\ell} \qquad \text{for } \ell \geq 2 \text{ and } \mathcal{J}_\ell \subseteq \{0, 1, \cdots, \ell-1\} \tag{1.3}$$

Since $\sigma(z)$ is degree 2-homogenous, without loss of generality we assume $\|\mathbf{W}_{\ell,j}^\star\|_2 = O(1)$, $u_\ell^\star \in \{-1,1\}^{k_\ell}$ and let $\alpha_\ell \in \mathbb{R}_{>0}$ be a scalar to control the contribution of the ℓ-th layer.

In the teacher-student setting, our main theorems can be sketched as follows:

Theorem (sketched)**.** *For every input dimension $d > 0$ and every $L = o(\log\log d)$, for certain concept class consisting of certain L-layer target networks defined in Eq. (1.3), over certain input distributions (such as standard Gaussian, certain mixture of Gaussians, etc.), we have:*

- *Within* $\mathsf{poly}(d/\varepsilon)$ *time/sample complexity, by a variant of SGD starting from random initialization, the L-layer quadratic DenseNet can learn this concept class with any generalization error ε, using* **forward feature learning + backward feature correction***. (See Theorem 1.)*
- *As side result, we show any kernel method, any linear model over prescribed feature mappings, or any two-layer neural networks with arbitrary degree-2^L activations, require $d^{\Omega(2^L)}$ sample or time complexity, to achieve non-trivial generalization error such as $\varepsilon = d^{-0.01}$. (See Section O.)*

Remark. As we shall formally introduce in Section 2, the concept class in our theorem — the class of target functions to be learned — comes from Eq. (1.3) with additional width requirement $k_\ell \approx d^{1/2^\ell}$ and *information gap* requirement $\alpha_{\ell+1} \ll \alpha_\ell$ with $\alpha_2 = 1$ and $\alpha_L \geq \frac{1}{\sqrt{d}}$. The requirement $L = o(\log\log d)$ is very natural: a quadratic network even with constant condition number can output 2^{2^L} and we need this to be at most $\mathsf{poly}(d)$ to prove any efficient training result.

Remark. We refer the assumption $\alpha_{\ell+1} \ll \alpha_\ell$ as ***information gap***. In a classification problem, it can be understood as "α_ℓ is the marginal accuracy improvement when using ℓ-layer networks to fit the target function comparing to $(\ell-1)$-layer ones." We discuss more in Section B.1. For example, in Figure 4, we see $> 75\%$ of the CIFAR-10 images can be classified correctly using a two-hidden-layer network; but going from depth 7 to depth 8 only gives $< 1\%$ accuracy gain. Information gap was also pointed out in natural language processing applications (Tenney et al., 2019).

1.2 High-Level Intuitions

Intuitively, learning a *single* quadratic function is easy, but our concept class consists of a sufficiently rich set of degree $2^L = 2^{\omega(1)}$ polynomials over d dimensions. Using non-hierarchical learning methods, typical sample/time complexity is $d^{\Omega(2^L)} = d^{\omega(1)}$ — and we prove such lower bound for kernel (and some other) methods, even when all $k_\ell = 1$. This *is <u>not</u> surprising*, since kernel methods *do not* perform hierarchical learning so have to essentially "write down" all the monomials of degree 2^{L-1}, which suffers a lot in the sample complexity. Even if the learner performs kernel method $O(1)$ times, since the target function has width $k_\ell = d^{\Omega(1)}$ for any constant ℓ, this cannot avoid learning in one level a degree-$\omega(1)$ polynomial that depends on $d^{\Omega(1)}$ variables, resulting again in sample/time complexity $d^{\omega(1)}$.

Now, the *hope* for training a quadratic DenseNet with $\mathsf{poly}(d)$ time, is because it may decompose a degree-2^L polynomial into learning one quadratic function at a time. Easier said than done, let us provide intuition by considering an extremely simplified example: $L = 3$, $d = 4$, and

$$G^\star(x) = x_1^4 + x_2^4 + \alpha((x_1^4 + x_3)^2 + (x_2^4 + x_4)^2) \quad \text{for some } \alpha = o(1).$$

(Recall $L = 3$ refers to having two trainable layers that we refer to as the second and third layers.)

Forward feature learning: richer representation by over-parameterization. Since $\alpha \ll 1$, one may hope for the second layer $G_2(x)$ to learn x_1^4 and x_2^4 — which is quadratic over $G_1(x)$ — through some representation of its neurons; then feed this as input to the third layer. If so, the third layer $G_3(x)$ could learn a quadratic function over x_1^4, x_2^4, x_3, x_4 to fit the remainder $\alpha((x_1^4 + x_3)^2 + (x_2^4 + x_4)^2)$ in the objective. This logic has a critical flaw:

- *Instead of learning x_1^4, x_2^4, the second layer may as well learn $\frac{1}{5}(x_1^2 + 2x_2^2)^2$, $\frac{1}{5}(2x_1^2 - x_2^2)^2$.*

Indeed, $\frac{1}{5}(x_1^2+2x_2^2)^2+\frac{1}{5}(2x_1^2-x_2^2)^2=x_1^4+x_2^4$; however, *no* quadratic function over $\frac{1}{5}(x_1^2+2x_2^2)^2, \frac{1}{5}(2x_1^2-x_2^2)^2$ and x_3, x_4 can produce $(x_1^4+x_3)^2+(x_2^4+x_4)^2$. Therefore, the second layer needs to learn not only how to fit $x_1^4+x_2^4$ but also the "correct basis" x_1^4, x_2^4 for the third layer.

To achieve this goal, we let the learner network to use (quadratically-sized) over-parameterization with random initialization. Instead of having only two hidden neurons, we will let the network have $m>2$ hidden neurons. We show a critical lemma that the neurons in the second layer of the network can learn *a richer representation* of the same function $x_1^4+x_2^4$, given by:

$$\{(\alpha_i x_1^2+\beta_i x_2^2)^2\}_{i=1}^m$$

In each hidden neuron, the coefficients α_i, β_i behave like i.i.d. Gaussians. Indeed, $\mathbb{E}[(\alpha_i x_1^2+\beta_i x_2^2)^2]\approx x_1^4+x_2^4$, and w.h.p. when $m\geq 3$, we can show that a quadratic function of $\{(\alpha_i x_1^2+\beta_i x_2^2)^2\}_{i=1}^m, x_3, x_4$ can be used to fit $(x_1^4+x_3)^2+(x_2^4+x_4)^2$, so the algorithm can proceed. Note this is a completely different view comparing to prior works: here over-parameterization is not to make training easier in the current layer; instead, it enforces the network to learn a richer set of hidden features (to represent the same target function) that can be better used for higher layers.

Backward feature correction: improvement in lower layers after learning higher layers. The second obstacle in this toy example is that the second layer might not even learn the function $x_1^4+x_2^4$ *exactly*. It is possible to come up with a distribution where the best quadratic over $G_1(x)$ (i.e., $x_1^2, x_2^2, x_3^2, x_4^2$) to fit $G^\star(x)$ is instead $(x_1^2+\alpha x_3^2)^2+(x_2^2+\alpha x_4^2)^2$, which is only of magnitude α close to the ideal function $x_1^4+x_2^4$. This is *over-fitting*, and the error $\alpha x_3^2, \alpha x_4^2$ *cannot* be corrected by over-parameterization. (More generally, this error in the lower-level features can propagate layer after layer, if one keeps performing forward feature learning without going back to correct them. This why we do not believe applying kernel method sequentially even $\omega(1)$ times can possibly learn our concept class in poly-time. We discuss more in Section 3.)

Let us proceed to see how this over-fitting on the second layer can be corrected by learning the third layer together. Say the **second layer has an "α-error"** and feeds the over-fit features $(x_1^2+\alpha x_3^2)^2, (x_2^2+\alpha x_4^2)^2$ to the third layer. The third layer can therefore use $\Delta'=\alpha((x_1^2+\alpha x_3^2)^2+x_3)^2+\alpha((x_2^2+\alpha x_4^2)^2+x_4)^2$ to fit the remainder term $\Delta=\alpha((x_1^4+x_3)^2+(x_2^4+x_4)^2)$ in $G^\star(x)$.

A very neat observation is that Δ' is only of magnitude α^2 away from Δ. Therefore, when the second and third layers are trained together, this "α^2-error" remainder Δ' will be subtracted from the training objective, so the **second layer can learn up to accuracy α^2, instead of α.** In other words, the amount of over-fitting is now reduced from α to α^2. We call this "backward feature correction" (see Figure 3, 5, and 10 in Section A). (This is also consistent with what we discover on *ReLU* networks in real-life experiments, see Figure 5 where we *visualize* such "over-fitting.")

In fact, this process $\alpha\to\alpha^2\to\alpha^3\to\cdots$ keeps going and the second layer can feed better and better features *to* the third layer (forward learning), via the reduction of over-fitting *from* the third layer (via backward correction). We can eventually learn $G^\star$ to arbitrarily small error $\varepsilon>0$. When there are more than two trainable layers, the process is slightly more involved, and we summarize this *hierarchical* learning process in Figure 6 on Page 15.

Hierarchical learning in deep learning goes *beyond* layer-wise training. Our results also shed lights on the following observation in practice: typically layer-wise training (i.e. train layers one by one starting from lower levels) performs much worse than training all the layers together, see Figure 7. The fundamental reason is due to the missing piece of "backward feature correction."

2 Target Network and Learner Network

Target network. Recall we have defined the layers $G_2^\star(x),\dots,G_L^\star(x)$ of target networks in (1.3). The weights $\mathbf{W}_{\ell,j}^\star\in\mathbb{R}^{k_\ell\times k_j}$ for $j\in\mathcal{J}_\ell$ and we write $\mathbf{W}_{\ell,j}^\star=0$ for $j\notin\mathcal{J}_\ell$. Our concept class to be learned consists of functions $G^\star\colon\mathbb{R}^d\to\mathbb{R}$ written as coordinate summation of each layer:[5]

$$G^\star(x)=\sum_{\ell=2}^L\alpha_\ell\cdot\mathbf{Sum}(G_\ell^\star(x)):=\sum_{\ell=2}^L\alpha_\ell\sum_{i\in[k_\ell]}G_{\ell,i}^\star(x)$$

[5] Our result trivially extends to the case when $\mathbf{Sum}(v)$ is replaced with $\sum_i p_i v_i$ where $p_i\in\{\pm1\}$ for half of the indices. We refrain from proving that version for notational simplicity.

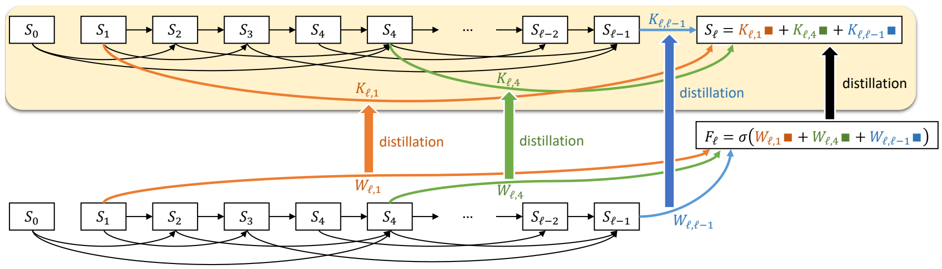


Figure 1: learner network structure with distillation

where $\mathbf{Sum}(v) := \sum_i v_i$, and it satisfies $\alpha_2 = 1$ and $\alpha_{\ell+1} < \alpha_\ell$. We will provide more explanation of the meaningfulness and necessity of information-gap $\alpha_{\ell+1} < \alpha_\ell$ in Section B.1.

It is convenient to define $S^\star_\ell(x)$ as the *hidden features* of target network (and $G^\star_\ell(x) = \sigma(S^\star_\ell(x))$).

$$S^\star_0(x) = G^\star_0(x) = x, \quad S^\star_1(x) = G^\star_1(x), \quad S^\star_\ell(x) := \sum_{j=0}^{\ell-1} \mathbf{W}^\star_{\ell,j} G^\star_j(x) \quad \forall \ell \geq 2$$

Learner network. Recall we have constructed the learner network to be of the same structure (with over-parameterization, see (1.2)) as $G^\star$. We choose $\mathbf{M}_{\ell,0}, \mathbf{M}_{\ell,1} \in \mathbb{R}^{\binom{k_\ell+1}{2}\times d}$ and $\mathbf{M}_{\ell,j} \in \mathbb{R}^{\binom{k_\ell+1}{2}\times\binom{k_j+1}{2}}$ for every $2 \leq j \leq \ell-1$. In other words, the amount of over-parameterization is quadratic (i.e., from $k_j \to \binom{k_j+1}{2}$) per layer. Using samples $(x, G^\star(x))$ from an unknown target network $G^\star(x)$, our goal is to learn weight matrices $\mathbf{M}_{\ell,j}$ to satisfy

$$G(x) := \sum_{\ell=2}^{L} \alpha_\ell \cdot \mathbf{Sum}(G_\ell(x)) \approx G^\star(x) \enspace .$$

2.1 Learner Network Re-parameterization

In this paper, for theoretical efficient training purpose, we work on a re-parameterization of the learner network. We use the following function to fit the target $G^\star(x)$:

$$F(x) = \sum_{\ell=2}^{L} \alpha_\ell \cdot \mathbf{Sum}(F_\ell(x))$$

where the layers are defined as: $S_0(x) = G^\star_0(x)$, $S_1(x) = G^\star_1(x)$, and for $\ell \geq 2$:

$$S_\ell(x) = \sum_{j\in\mathcal{J}_\ell, j\geq 2} \mathbf{K}_{\ell,j}\sigma\left(\mathbf{R}_j S_j(x)\right) + \sum_{j\in\{0,1\}\cap\mathcal{J}_\ell} \mathbf{K}_{\ell,j} S_j(x) \in \mathbb{R}^{k_\ell} \tag{2.1}$$

$$F_\ell(x) = \sigma\Big(\sum_{j\in\mathcal{J}_\ell, j\geq 2} \mathbf{W}_{\ell,j}\sigma\left(\mathbf{R}_j S_j(x)\right) + \sum_{j\in\{0,1\}\cap\mathcal{J}_\ell} \mathbf{W}_{\ell,j} S_j(x)\Big) \in \mathbb{R}^m \tag{2.2}$$

Above, we shall choose m to be polynomially large and let

- $\mathbf{R}_\ell \in \mathbb{R}^{\binom{k_\ell+1}{2}\times k_\ell}$ be randomly initialized for every layer ℓ, not changed during training; and
- $\mathbf{W}_{\ell,j} \in \mathbb{R}^{m\times q}, \mathbf{K}_{\ell,j} \in \mathbb{R}^{k_\ell\times q}$ be trainablefor every ℓ and $j \in \mathcal{J}_\ell$, and the dimension $q = \binom{k_j+1}{2}$ for $j \geq 2$ and $q = d$ for $j = 0, 1$.

It is easy to verify that when $\mathbf{R}_\ell^\top \mathbf{R}_\ell = \mathbf{I}$ and when $\mathbf{W}_{\ell,j} = \mathbf{K}_{\ell,j}$, by defining $\mathbf{M}_{\ell,j} = \mathbf{R}_\ell \mathbf{K}_{\ell,j}$ we have $F_\ell(x) = G_\ell(x)$ and $F(x) = G(x)$. We remark that the hidden dimension k_ℓ can also be learned during training, see Algorithm 1 in Section C.[6]

Why this re-parameterization. We work with this re-parameterization $F(x)$ for *efficient training purpose*. It is convenient to think of $S_\ell(x)$ as the "*hidden features*" used by the learner network. Since $S_\ell(x)$ is of the same dimension k_ℓ as $S^\star_\ell(x)$, our goal becomes to prove that the hidden features $S_\ell(x)$ and $S^\star_\ell(x)$ are close up to unitary transformation (i.e. Theorem 2).

One may also consider $F_\ell(x) = \sigma(\mathbf{W}\cdots)$ and treat the pre-activation part $(\mathbf{W}\cdots) \in \mathbb{R}^m$ in (2.2) — instead of $S_\ell(x) \in \mathbb{R}^{k_\ell}$ — as the "over-parameterized hidden features." This over-parameterization is used to make the training *provably efficient*. As we shall see, we will impose regularizers during training to enforce $\mathbf{K}^\top\mathbf{K} \approx \mathbf{W}^\top\mathbf{W}$; and this idea of using a larger unit $\mathbf{W}$

[6]From this definition, it seems the learner needs to know $\{\alpha_\ell\}_\ell$ and $\{\mathcal{J}_\ell\}_\ell$; as we point out in Section C, performing grid search over them is efficient in $\mathsf{poly}(d)$ time. This can be viewed as neural architecture search. As a consequence, in the agnostic setting, our theorem can be understood as: *"the learner network can fit the labeling function using the* ***best*** *$G^\star$ from the concept class as well as the* ***best*** *choices of $\{\alpha_\ell\}_\ell$ and $\{\mathcal{J}_\ell\}_\ell$."*

for training and using a smaller unit $\mathbf{K}$ to learn the larger one can be viewed as *knowledge distillation*. One can then argue that the "over-parameterized hidden features" are also close to $S^\star_\ell(x)$ up to knowledge distillation and unitary transformation. Knowledge distillation is commonly used in practice (Hinton et al., 2015), and we illustrate this by Figure 1.

Truncated quadratic activation. To make our theory simpler, during training, it would be easier to work with an activation function that has bounded derivatives in the entire space (recall the derivative $|\sigma'(z)| = |z|$ is unbounded). We make a theoretical choice of a *truncated quadratic activation* $\widetilde{\sigma}(z)$ that is sufficiently close to $\sigma(z)$. Accordingly, we rewrite $F(x)$, $F_\ell(x)$, $S_\ell(x)$ as $\widetilde{F}(x)$, $\widetilde{F}_\ell(x)$, $\widetilde{S}_\ell(x)$ whenever we replace $\sigma(\cdot)$ with $\widetilde{\sigma}(\cdot)$. (For completeness we still include the formal definition in Appendix H.1.) Our lemma — see Appendix J.1 — shall ensure that $F(x) \approx \widetilde{F}(x)$ and $S_\ell(x) \approx \widetilde{S}_\ell(x)$. Thus, our *final learned network $F(x)$ is truly quadratic*. In practice, people use batch/layer normalizations to make sure activations stay bounded, but truncation is more theory-friendly.

Notation simplification. We concatenate the weight matrices used in the same layer ℓ as follows:

$$\mathbf{W}_\ell = (\mathbf{W}_{\ell,j})_{j\in\mathcal{J}_\ell} \qquad \mathbf{K}_\ell = (\mathbf{K}_{\ell,j})_{j\in\mathcal{J}_\ell} \qquad \mathbf{W}^\star_\ell = \left(\mathbf{W}^\star_{\ell,j}\right)_{j\in\mathcal{J}_\ell}$$
$$\mathbf{W}_{\ell\triangleleft} = (\mathbf{W}_{\ell,j})_{j\in\mathcal{J}_\ell, j\neq\ell-1} \qquad \mathbf{K}_{\ell\triangleleft} = (\mathbf{K}_{\ell,j})_{j\in\mathcal{J}_\ell, j\neq\ell-1} \qquad \mathbf{W}^\star_{\ell\triangleleft} = \left(\mathbf{W}^\star_{\ell,j}\right)_{j\in\mathcal{J}_\ell, j\neq\ell-1}$$

2.2 Training Objective

We focus our notation for the regression problem in the realizable case. We will introduce notations for the agnostic case and for classification in Section B.1 when we need them.

As mentioned earlier, to perform knowledge distillation, we add a regularizer to ensure $\mathbf{W}_\ell^\top \mathbf{W}_\ell \approx \mathbf{K}_\ell^\top \mathbf{K}_\ell$ so that $\mathbf{K}_\ell^\top \mathbf{K}_\ell$ is a low-rank approximation of $\mathbf{W}_\ell^\top \mathbf{W}_\ell$. (This also implies $\mathbf{Sum}(F_\ell(x)) \approx \mathbf{Sum}(\sigma(S_\ell(x)))$.) Specifically, we use the following training objective:

$$\widetilde{\mathbf{Obj}}(x; \mathbf{W}, \mathbf{K}) = \widetilde{\mathbf{Loss}}(x; \mathbf{W}, \mathbf{K}) + \mathbf{Reg}(\mathbf{W}, \mathbf{K})$$

where the ℓ_2 loss is $\widetilde{\mathbf{Loss}}(x; \mathbf{W}, \mathbf{K}) = (G^\star(x) - \widetilde{F}(x))^2$ and

$$\mathbf{Reg}(\mathbf{W}, \mathbf{K}) = \textstyle\sum_{\ell=2}^L \lambda_{3,\ell} \left\| \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft} - \mathbf{W}_{\ell,\ell-1}^\top \mathbf{W}_{\ell\triangleleft} \right\|_F^2 + \sum_{\ell=2}^L \lambda_{4,\ell} \left\| \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} - \mathbf{W}_{\ell,\ell-1}^\top \mathbf{W}_{\ell,\ell-1} \right\|_F^2$$
$$+ \textstyle\sum_{\ell=2}^L \lambda_{5,\ell} \left\| \mathbf{K}_\ell^\top \mathbf{K}_\ell - \mathbf{W}_\ell^\top \mathbf{W}_\ell \right\|_F^2 + \sum_{\ell=2}^L \lambda_{6,\ell} \left(\|\mathbf{K}_\ell\|_F^2 + \|\mathbf{W}_\ell\|_F^2 \right) .$$

For a given set $\mathcal{Z}$ consisting of N i.i.d. samples from the true distribution $\mathcal{D}$, the *training process* minimizes the following objective ($x \sim \mathcal{Z}$ denotes x is uniformly sampled from the training set $\mathcal{Z}$)

$$\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) = \mathop{\mathbb{E}}_{x\sim\mathcal{Z}} [\widetilde{\mathbf{Obj}}(x; \mathbf{W}, \mathbf{K})] \tag{2.3}$$

The regularizers we used are just (squared) Frobenius norm on the weight matrices, which are common in practice. The regularizers associated with $\lambda_{3,\ell}, \lambda_{4,\ell}, \lambda_{5,\ell}$ are for *knowledge distillation propose* to make sure $\mathbf{K}$ is close to $\mathbf{W}$ (they are simply zero when $\mathbf{K}_\ell^\top \mathbf{K}_\ell = \mathbf{W}_\ell^\top \mathbf{W}_\ell$). They play *no role* in backward feature corrections (since layers ℓ and ℓ' for $\ell' \neq \ell$ are optimized *independently* in these regularizers). These corrections are done solely by SGD automatically.

For the original, non-truncated quadratic activation network, we also denote by

$$\mathbf{Loss}(x; \mathbf{W}, \mathbf{K}) = (G^\star(x) - F(x))^2 \text{ and } \mathbf{Obj}(x; \mathbf{W}, \mathbf{K}) = \mathbf{Loss}(x; \mathbf{W}, \mathbf{K}) + \mathbf{Reg}(\mathbf{W}, \mathbf{K}).$$

3 Statements of Main Result

We assume the *input distribution* $x \sim \mathcal{D}$ satisfies random properties such as isotropy and hyper-contractivity. We defer the details to Section D, while pointing out that not only standard Gaussian but even some mixtures of non-spherical Gaussians satisfy these properties (see Proposition D.1). For simplicity, the readers can think of $\mathcal{D} = \mathcal{N}(0, \mathbf{I})$ in this section.

We consider a concept class consisting of target networks satisfying the following parameters

1. (monotone) $d \geq k := k_2 \geq k_3 \geq \cdots \geq k_L$.

2. (normalized) $\mathbb{E}_{x\sim\mathcal{D}}[\mathbf{Sum}(G^\star_\ell(x))] \leq B_\ell$ for some $B_\ell \geq 1$ for all ℓ and $B := \max_\ell\{B_\ell\}$.
3. (well-conditioned) the singular values of $\mathbf{W}^\star_{\ell,j}$ are between $\frac{1}{\kappa}$ and κ for all $\ell, j \in \mathcal{J}_\ell$ pairs.

Remark 3.1. Properties $1, 3$ are satisfied for many practical networks; in fact, many practical networks have weight matrices close to unitary, see (Huang et al., 2018). For property 2, although there may exist some worst case $\mathbf{W}^\star_{\ell,j}$, at least when each $\mathbf{W}^\star_{\ell,j}$ is of the form $\mathbf{U}_{\ell,j}\mathbf{\Sigma}\mathbf{V}_{\ell,j}$ for $\mathbf{U}_{\ell,j}, \mathbf{V}_{\ell,j}$ being random orthonormal matrices, with probability at least 0.9999, it holds $B_\ell = \kappa^{2^{O(\ell)}} k_\ell$ for instance for standard Gaussian inputs — this is small since $L \leq o(\log\log d)$. Another view is that practical networks are equipped with batch/layer normalizations, which ensure $B_\ell = O(k_\ell)$.

Our results. In the main body of this paper, we state a special case of our main (positive result) Theorem 1 which is sufficiently interesting and has simpler notations. The full Theorem 1' is in Appendix H. In this special case, we assume there are absolute integer constants $C > C_1 \geq 2$ such that, the concept class consists of target networks $G^\star(x)$ satisfies the above three properties with parameters $\kappa \leq 2^{C_1^L}, B_\ell \leq 2^{C_1^\ell} k_\ell, k_\ell \leq d^{\frac{1}{C^{\ell+C_1}}}$ and there is an information gap $\frac{\alpha_{\ell+1}}{\alpha_\ell} \leq d^{-\frac{1}{C^\ell}}$ for $\ell \geq 2$; furthermore, suppose in the connection graph $\{2, 3, \cdots, \ell - C_1\} \cap \mathcal{J}_\ell = \varnothing$, meaning that the skip connections do not go very deep, unless directly connected to the input.

Theorem 1 (special case of Theorem 1'). *In the special case as defined above, for every sufficiently large $d > 0$, every $L = o(\log\log d)$, every $\varepsilon \in (0, 1)$, consider any target network $G^\star(x)$ satisfying the above parameters. Then, given $N = \mathsf{poly}(d/\varepsilon)$ i.i.d. samples x from $\mathcal{D}$ with corresponding labels $G^\star(x)$, by applying Algorithm 1 (a variant of SGD) with over-parameterization $m = \mathsf{poly}(d/\varepsilon)$ and learning rate $\eta = \frac{1}{\mathsf{poly}(d/\varepsilon)}$ over the training objective (2.3), with probability at least 0.99, we can find a learner network F in time $\mathsf{poly}(d/\varepsilon)$ such that:*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}} \big(G^\star(x) - F(x)\big)^2 \leq \varepsilon^2 \quad \textit{and} \quad \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \big(G^\star(x) - \widetilde{F}(x)\big)^2 \leq \varepsilon^2 \ .$$

We defer the detailed pseudocode of Algorithm 1 to Section C but make several remarks:

- Note $\alpha_{\ell+1} = \alpha_\ell d^{-\frac{1}{C^\ell}}$ implies $\alpha_L \geq d^{-\frac{1}{C}} \geq \frac{1}{\sqrt{d}}$. Hence, to achieve for instance $\varepsilon \leq \frac{1}{d^4}$ error, the learning algorithm has to truly learn *all* the layers of $G^\star(x)$, as opposed to for instance ignoring the last layer which will incur error $\alpha_L \gg \varepsilon$.
- The reason we focus on $L = o(\log\log d)$ and well-conditioned target networks should be natural. Since the target network is of degree 2^L, we wish to have $\kappa^{2^L} \leq \mathsf{poly}(d)$ so the output of the network is bounded by $\mathsf{poly}(d)$ for efficient learning.

The main conceptual and technical contribution of our paper is the "backward feature correction" process. To illustrate this, we highlight a critical lemma in our proof and state it as a theorem:

Backward Feature Correction Theorem

Theorem 2 (highlight of Corollary L.3d). *In the setting of Theorem 1, during the training process, suppose the first ℓ-layers of the learner network* has achieved *ε generalization error,*

$$\textit{or in symbols,} \quad \mathbb{E}\left[\left(G^\star(x) - \textstyle\sum_{\ell'\leq\ell} \alpha_{\ell'} \mathbf{Sum}(F_{\ell'}(x))\right)^2\right] \leq \varepsilon^2 \ , \tag{3.1}$$

then for every $\ell' \leq \ell$, there is unitary matrix $\mathbf{U}_{\ell'} \in \mathbb{R}^{k_{\ell'}\times k_{\ell'}}$ such that (we write $\alpha_{L+1} = 0$)

$$\mathbb{E}\left[\alpha_{\ell'}^2 \left\|S^\star_{\ell'}(x) - \mathbf{U}_{\ell'} S_{\ell'}(x)\right\|^2\right] \lesssim (\alpha_{\ell+1}^2 + \varepsilon^2) \ .$$

In other words, once we have trained the first ℓ layers well enough, for some lower-level layer $\ell' \leq \ell$, the "error in the learned features $S_{\ell'}(x)$ comparing to $S^\star_{\ell'}(x)$" is *proportional to* $\alpha_{\ell+1}$. Recall for fixed ℓ', as we increase ℓ the value $\alpha_{\ell+1}$ decreases, thus Theorem 2 suggests that

the lower-level features can actually get improved when we train higher-level layers together.

Remark 3.2. Theorem 2 is not a "representation" theorem. There might be other networks F such that (3.1) is satisfied but $S_{\ell'}(x)$ is not close to $S^\star_{\ell'}(x)$ at all. Theorem 2 implies *during the training process*, as long as we following carefully the training process of SGD, such "bad F" will be automatically avoided. We give more details in our intuition and sketched proof Section E.

Comparing to sequential kernel methods. Recall we have argued in Section 1.2 that our concept class is not likely to be efficiently learnable, if one applies kernel method $O(1)$ times sequentially. Even if one applies kernel method for $\omega(1)$ rounds, this is similar to *layer-wise training* and misses "backward feature correction." As we pointed out using examples in Section 1.2, this is unlikely to learn the target function to good accuracy either. In fact, one may consider "sequential kernel" together with "backward feature correction", but even this may not always work, since small generalization error does not necessarily imply sufficient accuracy on intermediate features *if we do not follow the SGD training process* (see Remark 3.2).[7]

Importance of hierarchical learning. To the best of our knowledge, for the concept class considered in this paper, we do not know any other simple algorithm to learn it in polynomial time, and the only simple learning algorithm we are aware of is to train a neural network to perform hierarchical learning. In other words, we believe we have presented a setting where we prove that training a neural network via a simple SGD variant can perform hierarchical learning, to solve an underlying problem that is not known solvable by existing algorithms, including applying kernel methods sequentially multiple times, tensor decomposition methods, sparse coding.

3.1 Backward Feature Correction: How deep? How much?

How deep does it need for the neural network to perform backward feature correction? In our theoretical result, we studied an extreme case in which training the L-th layer can even backward correct the learned weights on the first layer for $L = \omega(1)$ (see Theorem 2). In practice, we demonstrate that backward feature correction may indeed need to be deep. For the 34-layer WideResNet architecture on CIFAR tasks, see Figure 8 on Page 16, we show that backward feature correction happens for *at least* 8 *layers*, meaning that if we first train all the $\leq \ell$ layers for some large ℓ (say $\ell = 21$), the features in layer $\ell - 8, \ell - 7, \cdots, \ell$ *still need to be (locally) improved* in order to become the best features comparing to training all the layers together.

We also give a characterization on **how much** the features need to be backward corrected using theory and experiments. On the empirical side, we measure the changes given by backward feature correction in Figure 8 and 9. We detect that these changes are *local*: meaning although the lower layers need to change when training with higher layers together to obtain the highest accuracy, they *do not change by much* (the correlation of layer weights before and after backward correction is more than 0.9). In Figure 10, we also visualize the neurons at different layers, so that one can easily see backward feature correction is indeed a *local correction process in practice*.

This is consistent with our theory. Theorem 2 shows at least for our concept class, backward feature correction is a local correction, meaning that the amount of feature change to the lower-level layers (when trained together with higher-level layers) is only little-$o(1)$ due to $\alpha_{\ell+1} \ll \alpha_{\ell'}$.

Intuitively, the locality comes from "information gap", which asserts that the lower layers in $G^\star$ can already fit a majority of the labels. When the lower layers in G are trained, their features will already be close to those "true" lower-level features in $G^\star$ and only a *local correction* is needed.

We believe that *the need for only local backward feature corrections is one of the main reasons that deep learning works in practice* on performing efficient (deep) hierarchical learning. We refer to (Allen-Zhu & Li, 2019a) for empirical evidence that deep learning fails to perform hierarchical learning when information gap is removed and the correction becomes non-local, even in the teacher-student setting with a hierarchical target network exactly generating the labels. The main contribution of our theoretical result is to show that such local "backward feature correction" can be done automatically when applying (a variant of) SGD to the training objective.

What's in Supplementary Materials. We include an Appendix I to cover some missing details: including missing Figures 2-13, and experiments to support the connection between our theory and practice, theorem statements in the agonistic or classification settings, formal specification of the SGD training algorithm, formal specification of the data distribution, as well as a sketched proof. Detailed detailed proofs to Appendix II.

[7] One may also want to connect this to (Allen-Zhu & Li, 2019a): according to Footnote 16, the analysis from (Allen-Zhu & Li, 2019a) is analogous to doing "sequential kernel" for 2 rounds, but even if one wants to backward correct the features of the first hidden layer, its error remains to be α and cannot be improved to arbitrarily small.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. LazySVD: even faster SVD decomposition yet without agonizing pain. In *NeurIPS*, pp. 974–982, 2016.

Zeyuan Allen-Zhu and Yuanzhi Li. What Can ResNet Learn Efficiently, Going Beyond Kernels? In *NeurIPS*, 2019a. Full version available at `http://arxiv.org/abs/1905.10337`.

Zeyuan Allen-Zhu and Yuanzhi Li. Can SGD Learn Recurrent Neural Networks with Provable Generalization? In *NeurIPS*, 2019b. Full version available at `http://arxiv.org/abs/1902.01028`.

Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *FOCS*, 2021. Full version available at `http://arxiv.org/abs/2005.10190`.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers. In *NeurIPS*, 2019a. Full version available at `http://arxiv.org/abs/1811.04918`.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the convergence rate of training recurrent neural networks. In *NeurIPS*, 2019b. Full version available at `http://arxiv.org/abs/1810.12065`.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *ICML*, 2019c. Full version available at `http://arxiv.org/abs/1811.03962`.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *International Conference on Machine Learning*, pp. 584–592, 2014.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019a.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *CoRR*, abs/1901.08584, 2019b. URL `http://arxiv.org/abs/1901.08584`.

Ainesh Bakshi, Rajesh Jayaram, and David P Woodruff. Learning two layer rectified neural networks in polynomial time. *arXiv preprint arXiv:1811.01885*, 2018.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International Conference on Machine Learning*, pp. 583–593, 2019.

Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

Digvijay Boob and Guanghui Lan. Theoretical properties of the global optimizer of two layer neural network. *arXiv preprint arXiv:1710.11241*, 2017.

Jacob V Bouvrie. *Hierarchical learning: Theory with applications in speech and vision*. PhD thesis, Massachusetts Institute of Technology, 2009.

Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. *arXiv preprint arXiv:1702.07966*, 2017.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 10835–10845, 2019.

Amit Daniely. Sgd learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2017.

Amit Daniely and Eran Malach. Learning parities with neural networks. *arXiv preprint arXiv:2002.07400*, 2020.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2253–2261, 2016.

Simon S Du and Wei Hu. Width provably matters in optimization for deep linear neural networks. *arXiv preprint arXiv:1901.08572*, 2019.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, November 2018a.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018b.

Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on learning theory*, pp. 907–940, 2016.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 563–574. IEEE, 2006.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pp. 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Learning one-hidden-layer neural networks with landscape design. *arXiv preprint arXiv:1711.00501*, 2017.

Rong Ge, Rohith Kuditipudi, Zhize Li, and Xiang Wang. Learning two-layer neural networks with symmetric inputs. *arXiv preprint arXiv:1810.06793*, 2018.

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. *arXiv preprint arXiv:1909.08156*, 2019.

Lei Huang, Xianglong Liu, Bo Lang, Adams Wei Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in neural information processing systems*, pp. 103–112, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Kenji Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pp. 586–594, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Yuanzhi Li and Zehao Dou. When can wasserstein gans minimize wasserstein distance? *arXiv preprint arXiv:2003.04033*, 2020.

Yuanzhi Li and Yingyu Liang. Provable alternating gradient descent for non-negative matrix factorization with strong correlations. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2062–2070. JMLR. org, 2017.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, 2018.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pp. 597–607.

http://arxiv.org/abs/1705.09886, 2017.

Yuanzhi Li, Yingyu Liang, and Andrej Risteski. Recovery guarantee of non-negative matrix factorization via alternating updates. In *Advances in neural information processing systems*, pp. 4987–4995, 2016.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *COLT*, 2018.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*, 2019a.

Yuanzhi Li, Tengyu Ma, and Hongyang R Zhang. Learning over-parametrized two-layer relu neural networks beyond ntk. *arXiv preprint arXiv:2007.04596*, 2020.

Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019b.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*, 2020a.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*, 2020b.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in Neural Information Processing Systems*, pp. 855–863, 2014.

Shachar Lovett. An elementary proof of anti-concentration of polynomials in gaussian variables. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 17, pp. 182, 2010.

Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. Delphi: A cryptographic inference service for neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 2505–2522. USENIX Association, August 2020. ISBN 978-1-939133-17-5. URL `https://www.usenix.org/conference/usenixsecurity20/presentation/mishra`.

Elchanan Mossel. Deep learning and hierarchal generative models. *arXiv preprint arXiv:1612.09057*, 2016.

Ido Nachum and Amir Yehudayoff. On symmetry and initialization for neural networks. In *LATIN 2020*, pp. 401–412, 2020.

Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *arXiv preprint arXiv:1902.04674*, 2019.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pp. 11289–11300, 2019.

Warren Schudy and Maxim Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pp. 437–446. Society for Industrial and Applied Mathematics, 2012.

Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Ludwig Schmidt, Jonathan Ragan-Kelley, and Benjamin Recht. Neural kernels without tangents. *arXiv preprint arXiv:2003.02237*, 2020.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *arXiv preprint arXiv:1707.04926*, 2017.

Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

Matus Telgarsky. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. *arXiv preprint arXiv:1703.00560*, 2017.

Loc Quang Trinh. Greedy layerwise training of convolutional neural networks. Master's thesis, Massachusetts Institute of Technology, 2019.

Santosh Vempala and John Wilmes. Polynomial convergence of gradient descent for training one-hidden-layer neural networks. *arXiv preprint arXiv:1805.02677*, 2018.

Bo Xie, Yingyu Liang, and Le Song. Diversity leads to generalization in neural networks. *arXiv preprint Arxiv:1611.03131*, 2016.

Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. *arXiv preprint arXiv:1904.00687*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. *arXiv preprint arXiv:1806.07808*, 2018.

Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. *arXiv preprint arXiv:1706.03175*, 2017.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2053–2062, 2019.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.

APPENDIX I: MISSING DETAILS AND SKETCHED PROOFS

We include in Section A missing figures for the main body. We include in Section B some missing statements of the theorem, for the agnostic learning setting as well as for a classification task. We also include in Section B some detailed elaborations on the information gap assumption. We formally include the specifications of Algorithm 1 in Section C. The requirements on the input distribution $\mathcal{D}$ is given in Section D (recall standard Gaussians and certain mixture of Gaussians are permitted). We give sketched proofs in Section E, and discuss more related works in Section F. We explain our experiment setups and give additional experiments in Section G.

A MISSING FIGURES

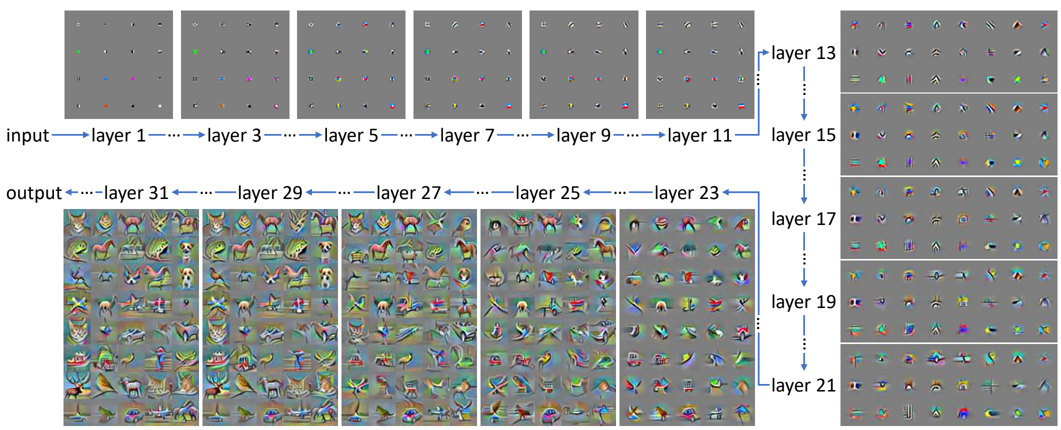


Figure 2: Illustration of the hierarchical learning process of ResNet-34 on CIFAR-10. Details see Section G.1.

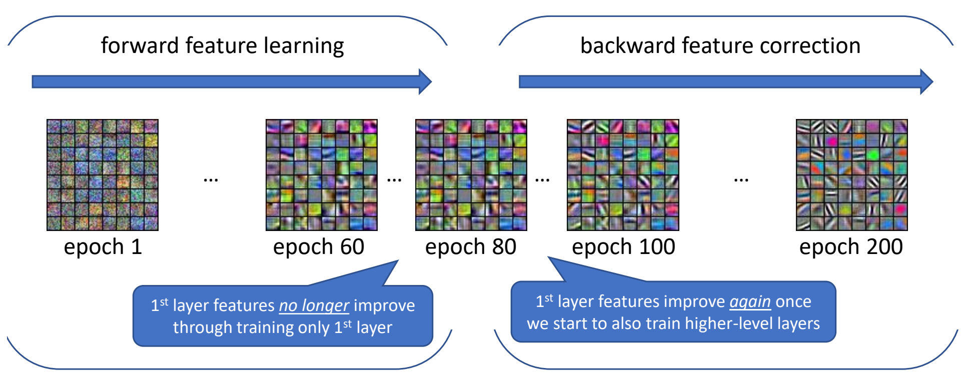


Figure 3: Convolutional features of the first layer in AlexNet. In the first 80 epochs, we train only the first layer freezing layers $2 \sim 5$; in the next 120 epochs, we train all the layers together (starting from the weights in epoch 80). Details in Appendix G.2. For visualizations of *deeper layers of ResNet*, see Figure 5 and 10.

Observation: In the first 80 epochs, when the first layer is trained until convergence, its features can already catch certain meaningful signals, but cannot get further improved. As soon as the 2nd through 5th layers are added to the training parameters, features of the first layer get improved again.

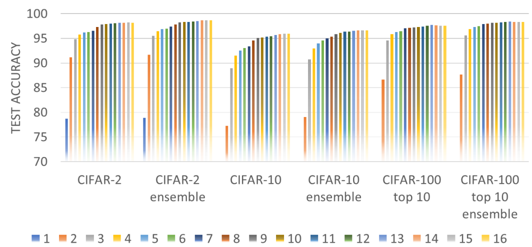


Figure 4: Justification of **information gap** on the CIFAR datasets for WRN-34-10 architecture. The 16 colors represent 16 different depths, and deeper layers have diminishing contributions to the classification accuracy. We discuss details in Section B.1 and experiment details in Appendix G.6.

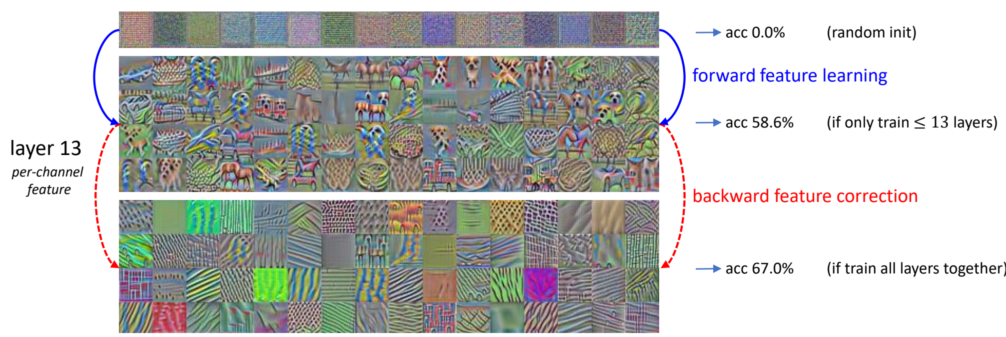


Figure 5: Visualize backward feature correction using WRN-34-5 on ℓ_2 *adversarial training*. Details in Appendix G.5.

Observation: if only training lower-level layers of a neural network, the features over-fit to higher-complexity signals of the images; while if training all the layers together, the higher-complexity signals are learned on higher-level layers and shall be "subtracted" from the lower-level features. The mathematical intuitions can be found in Section 1.2.

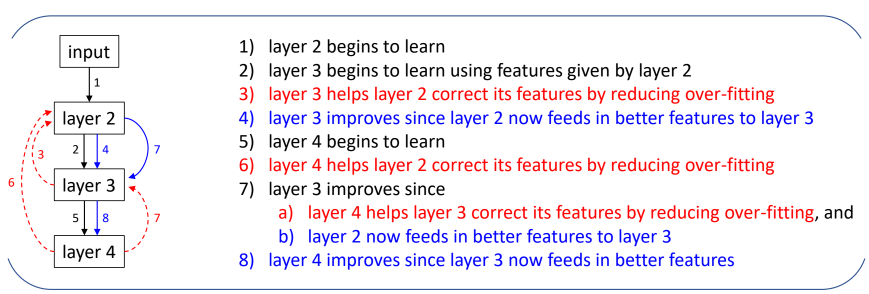
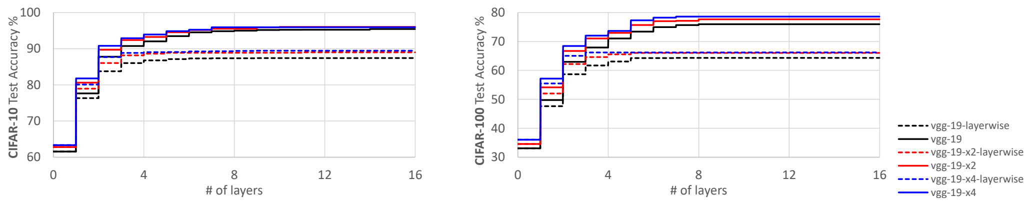
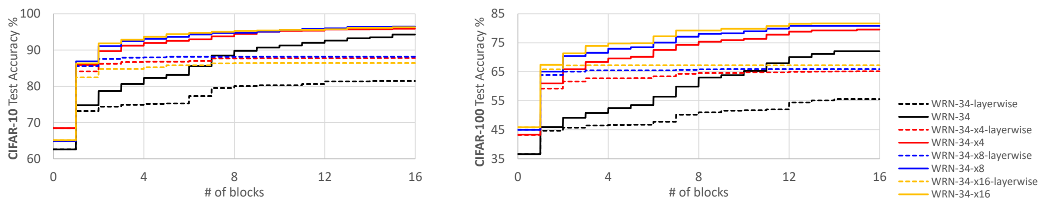


Figure 6: We explain the hierarchical learning process in a 4-layer example. The **back** and **blue** arrows correspond to "forward feature learning" (Allen-Zhu & Li, 2019a). The **red** dashed arrows correspond to "backward feature correction".
Note: In our work, we do not explicitly train the network in this order, this "back and forth" learning process happens rather implicitly when we simply train all layers in the network together.



(a) VGG19+BatchNorm, accuracy at x-axis S indicates only the first S convolutional layers are trained



(b) WideResNet-34, accuracy at x-axis S indicates only the first S convolutional blocks are trained

Figure 7: **Layerwise training vs Training all layers together.**
xN means widen factor N. Architecture is optimized so that layerwise training obtains its best performance. For details and more experiments on VGG-13 and ResNet-22, see Appendix G.4.

Take-away messages: During layer-wise training, lower layers are trained too greedily and over-fit to higher-complexity signals. This leads to worse final accuracy comparing to hierarchical learning (i.e., training all the layers together), even at the *second hidden layer*. Going deeper cannot increase accuracy anymore due to the low-quality features at lower layers that are already fixed. In fact, for moderately wide (e.g. width=64) architectures such as VGG or wide ResNet, layer-wise training stops improving test accuracy even after depth three *without Backward Feature Correction*.

	CIFAR-100 accuracy	$\ell=1$	$\ell=3$	$\ell=5$	$\ell=7$	$\ell=9$	$\ell=11$	$\ell=13$	$\ell=15$	$\ell=17$	$\ell=19$	$\ell=21$	
ensemble	train only $\le \ell$	16.0%	43.1%	61.5%	67.9%	70.7%	71.5%	75.9%	78.6%	79.8%	80.6%	80.9%	→ no BFC
	fix $\le \ell$, train the rest	83.1%	78.9%	78.4%	76.9%	75.9%	75.6%	77.6%	79.6%	80.5%	81.0%	81.2%	→ no BFC
	fix $\le \ell-2$, train the rest	-	83.4%	81.5%	79.8%	78.4%	77.4%	78.7%	80.3%	80.7%	81.0%	81.3%	→ BFC for 2 layers
	fix $\le \ell-4$, train the rest	-	-	83.1%	81.9%	81.2%	80.1%	80.8%	82.6%	82.2%	81.0%	81.3%	→ BFC for 4 layers
	fix $\le \ell-6$, train the rest	-	-	-	83.2%	82.3%	82.0%	81.8%	82.4%	82.7%	82.0%	81.8%	→ BFC for 6 layers
	fix $\le \ell-8$, train the rest	-	-	-	-	83.4%	82.2%	82.4%	83.1%	83.2%	82.7%	81.9%	→ BFC for 8 layers
	train all the layers	83.2%	83.2%	83.0%	82.9%	82.8%	83.0%	83.1%	83.1%	82.9%	83.2%	83.0%	→ full BFC
	average weight correlations (for "train $\le \ell$" vs "rand init")	*0.131*	*0.081*	*0.070*	*0.054*	*0.051*	*0.037*	*0.036*	*0.034*	*0.034*	*0.032*	*0.031*	→ training neural nets is far from the NTK regime
	average weight correlations (for "train $\le \ell$" vs "train all")	*0.927*	*0.956*	*0.966*	*0.958*	*0.965*	*0.960*	*0.950*	*0.968*	*0.967*	*0.959*	*0.948*	→ correlation between with vs. without BFC
single model	train only $\le \ell$	16.0%	41.1%	58.5%	64.0%	67.3%	68.0%	72.0%	74.8%	76.0%	76.1%	76.9%	→ no BFC
	fix $\le \ell$, train the rest	79.4%	72.9%	72.7%	70.6%	69.8%	69.9%	72.1%	74.2%	75.2%	75.2%	76.1%	→ no BFC
	fix $\le \ell-2$, train the rest	-	79.6%	77.5%	74.5%	72.3%	71.5%	72.8%	74.7%	75.0%	75.1%	75.8%	→ BFC for 2 layers
	fix $\le \ell-4$, train the rest	-	-	79.9%	78.1%	76.2%	74.8%	75.2%	77.3%	76.9%	75.8%	75.9%	→ BFC for 4 layers
	fix $\le \ell-6$, train the rest	-	-	-	80.0%	78.6%	77.1%	76.8%	78.0%	78.9%	77.4%	76.8%	→ BFC for 6 layers
	fix $\le \ell-8$, train the rest	-	-	-	-	79.9%	78.4%	78.3%	78.7%	79.2%	79.0%	77.6%	→ BFC for 8 layers
	train all the layers	79.7%	79.9%	79.6%	79.4%	79.3%	79.4%	79.4%	79.1%	79.0%	79.3%	79.1%	→ full BFC

Figure 8: CIFAR-100 accuracy difference on WideResNet-34-5 with vs. without backward feature correction (BFC).

In the table, "train $\le \ell$" means training only the first ℓ convolutional layers; average weight correlation is the average of $\langle \frac{w_i}{\|w_i\|}, \frac{w_i'}{\|w_i'\|} \rangle$ where w_i and w_i' are the neuron weight vectors before and after BFC. More experiments on CIFAR-10 and on adversarial training see Appendix G.5.

Observation: (1) at least 8 layers of backward feature correction is necessary for obtaining the best accuracy; (2) BFC is indeed a *local feature correction* process because neuron weights strongly correlate with those before BFC; and (3) neural tangent kernel (NTK) approach is insufficient to explain neural network training because neuron correlations with the random initialization is small.

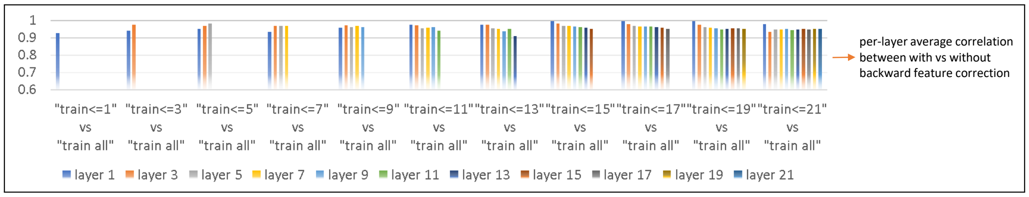


Figure 9: A more refined version of Figure 8 to show the *per-block* average weight correlations.

Observation: BFC is *local correction* because neuron weights strongly correlate with those before BFC.

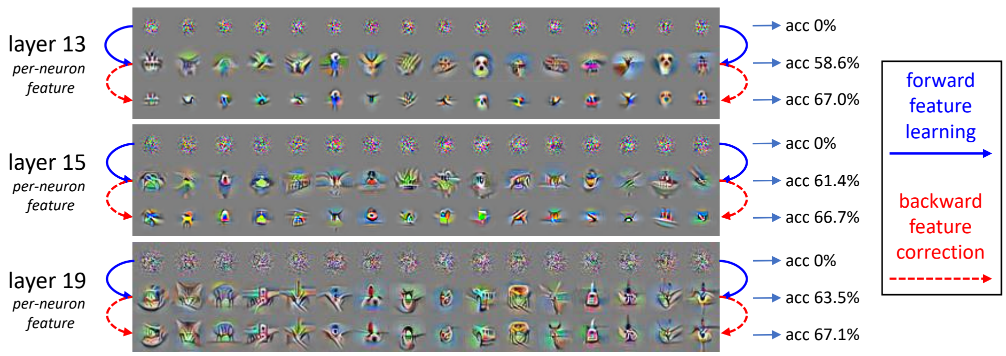


Figure 10: Visualize backward feature correction (per-neuron features) using WRN-34-5 on ℓ_2 *adversarial training*. Details in Appendix G.5.

Observation: backward feature correction is a *local correction* but is necessary for the accuracy gain.

B Statements of Main Result (continued)

Agnostic learning. Our theorem also works in the agnostic setting, where the labeling function $Y(x)$ satisfies $\mathbb{E}_{x\sim\mathcal{D}}(G^\star(x) - Y(x))^2 \le \mathsf{OPT}$ and $|G^\star(x) - Y(x)| \le \mathsf{poly}(d)$ for some *unknown* $G^\star(x)$. The SGD algorithm can learn a function $F(x)$ with error at most $(1+\gamma)\mathsf{OPT} + \varepsilon^2$ for any constant $\gamma > 1$ given i.i.d. samples of $\{x, Y(x)\}$. Thus, the learner can *compete* with the performance of the best target network. We present the result in Appendix H.5 and state its special case below.

Theorem 3 (special case of Theorem 3'). *For every constant $\gamma > 0$, in the same setting Theorem 1, given $N = \mathsf{poly}(d/\varepsilon)$ i.i.d. samples $\mathcal{Z}$ from $\mathcal{D}$ and their corresponding labels $\{Y(x)\}_{x\in\mathcal{Z}}$, by applying Algorithm 1 (a variant of SGD) over the agnostic training objective $\mathbb{E}_{x\sim\mathcal{Z}}\left(Y(x) - \widetilde{F}(x)\right)^2 + \mathbf{Reg}(\mathbf{W},\mathbf{K})$, with probability ≥ 0.99, it finds a learner network F in time $\mathsf{poly}(d/\varepsilon)$ s.t.*

$$\mathbb{E}_{x\sim\mathcal{D}}\left(F(x) - Y(x)\right)^2 \le \varepsilon^2 + (1+\gamma)\mathsf{OPT} \ .$$

B.1 More on Information Gap and Classification Problem

We have made a gap assumption $\frac{\alpha_{\ell+1}}{\alpha_\ell} \le d^{-\frac{1}{C^{\ell+1}}}$, which says in the target function $G^\star(x)$, *higher levels contribute less to its output*. This is typical for tasks such as image classification on CIFAR-10, where the first convolutional layer can already be used to classify $> 75\%$ of the data and higher-level layers have diminishing contributions to the accuracy (see Figure 4 on Page 15). For such classification tasks, researchers do *fight for* even the final 0.1% performance gain by going for (much) larger networks, so those higher-level functions *cannot be ignored*.

Information Gap: Empirically. We point out that *explicitly* setting higher levels in the network to contribute less to the output has also been used empirically to improve the performance of training deep neural networks, such as training very deep transformers (Liu et al., 2020a;b; Huang et al., 2019).

To formally justify information gap, it is beneficial to consider a *classification* problem. W.l.o.g. scale $G^\star(x)$ so that $\mathbf{Var}_x[G^\star(x)] = 1$, and consider a two-class labeling function $Y(x_0, x)$:

$$Y(x_0, x) = \mathsf{sgn}(x_0 + G^\star(x)) \in \{-1, 1\} \ ,$$

where $x_0 \sim \mathcal{N}(-\mathbb{E}_x[G^\star(x)], 1)$ is a Gaussian random variable independent of x. Here, x_0 can be viewed either a coordinate of the entire input $(x_0, x) \in \mathbb{R}^{d+1}$, or more generally as linear direction $x_0 = w^\top \widehat{x}$ for the input $\widehat{x} \in \mathbb{R}^{d+1}$. For notation simplicity, we focus on the former view.

Using probabilistic arguments, one can derive that except for α_ℓ fraction of the input $(x_0, x) \sim \mathcal{D}$, the label function $Y(x_0, x)$ is fully determined by the target function $G^\star(x)$ up to layer $\ell - 1$; or in symbols,

$$\mathop{\mathbf{Pr}}_{(x_0,x)\sim\mathcal{D}}\left[Y(x_0, x) \neq \mathsf{sgn}\left(x_0 + \textstyle\sum_{s\leq\ell-1} \alpha_s \mathbf{Sum}(G^\star_s(x))\right)\right] \approx \alpha_\ell \ .$$

In other words, for *binary classification*:

α_ℓ is (approximately) the increment in classification accuracy

when we use an ℓ-layer network comparing to $(\ell - 1)$-layer ones

Therefore, information gap is equivalent to saying that harder data (which requires deeper networks to learn) are fewer in the training set, which can be *very natural* . For instance, around 70% images of the CIFAR-10 data can be classified correctly by merely looking at their rough colors and patterns using a one-hidden-layer network; the final $< 1\%$ accuracy gain requires much refined arguments such as whether there is a beak on the animal face which can only be detected using very deep networks. As another example, humans use much more training examples to learn counting, than to learn basic calculus, than to learn advanced calculus.

For *multi-class classification*, information gap can be further relaxed. On CIFAR-100, a three-hidden layer network can already achieve 86.64% top-10 accuracy (see Figure 4 on Page 15), and the remaining layers only need to pick labels from these ten classes instead of the original 100 classes.

In this classification regime, our Theorem 1 still applies as follows. Recall the cross entropy (i.e., logistic loss) function $\mathsf{CE}(y, z) = -\log \frac{1}{1+e^{-yz}}$ where $y \in \{-1, 1\}$ is the label and $z \in \mathbb{R}$ is the prediction. In this regime, we can choose a training loss function

$$\begin{aligned}\widetilde{\mathbf{Loss}}^{\mathsf{xE}}(x_0, x; \mathbf{W}, \mathbf{K}) &:= \mathsf{CE}(Y(x_0, x), v(x_0 + \widetilde{F}(x; \mathbf{W}, \mathbf{K})))\\ &= \log\left(1 + e^{-Y(x_0,x)\cdot v(x_0+\widetilde{F}(x;\mathbf{W},\mathbf{K}))}\right)\end{aligned}$$

where the parameter v is around $\frac{1}{\varepsilon}$ is for proper normalization and the training objective is

$$\widetilde{\mathbf{Obj}}^{\mathsf{xE}}(x_0, x; \mathbf{W}, \mathbf{K}) = \widetilde{\mathbf{Loss}}^{\mathsf{xE}}(x_0, x; \mathbf{W}, \mathbf{K}) + v\mathbf{Reg}(\mathbf{W}, \mathbf{K}) \tag{B.1}$$

We have the following corollary of Theorem 1:

Theorem 4 (classification)**.** *In the same setting Theorem 1, and suppose additionally $\varepsilon > \frac{1}{d^{100\log d}}$. Given $N = \mathsf{poly}(d/\varepsilon)$ i.i.d. samples $\mathcal{Z}$ from $\mathcal{D}$ and given their corresponding labels $\{Y(x_0, x)\}_{(x_0,x)\in\mathcal{Z}}$, by applying a variant of SGD (Algorithm 1) over the training objective $\widetilde{\mathbf{Obj}}^{\mathsf{xE}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$, with probability at least 0.99, we can find a learner network F in time $\mathsf{poly}(d/\varepsilon)$ such that:*

$$\mathop{\mathbf{Pr}}_{(x_0,x)\sim\mathcal{D}}[Y(x_0, x) \neq \mathsf{sgn}(x_0 + F(x))] \leq \varepsilon \ .$$

Intuitively, Theorem 4 is possible because under the choice $v = 1/\varepsilon$, up to small multiplicative factors, "ℓ_2-loss equals ε^2" becomes near identical to "cross-entropy loss equals ε". This is why we need to add a factor v in from of the regularizers in (B.1). We make this rigorous in Appendix N.

C TRAINING ALGORITHM

We describe our algorithm in Algorithm 1. It is almost the vanilla SGD algorithm: in each innermost iteration, it gets a random sample $z \sim \mathcal{D}$, computes (stochastic) gradient in $(\mathbf{W}, \mathbf{K})$, and moves in the negative gradient direction with step length $\eta > 0$.

To make our analysis simpler, we made several *minor modifications only for theory purpose* on Algorithm 1 so that it may not appear immediate like the vanilla SGD at a first reading.

- We added a target error ε_0 which is initially large, and when the empirical objective $\widetilde{\mathbf{Obj}}$ falls below $\frac{1}{4}(\varepsilon_0)^2$ we set $\varepsilon_0 \leftarrow \varepsilon_0/2$. This lets us gradually decrease the weight decay factor $\lambda_{6,\ell}$.
- We divided Algorithm 1 into stages, where in each stage a deeper layer is added to the set of trainable variables. (When $\widetilde{\mathbf{Obj}}$ falls below $\mathsf{Thres}_{\ell,\triangle}$, we add $\mathbf{W}_\ell$ to the set; when it falls below $\mathsf{Thres}_{\ell,\triangledown}$, we add $\mathbf{K}_\ell$ to the set.) This is known as *layerwise pre-training* and we use it to simplify analysis. In practice, even when all the layers are trainable from the beginning, higher-level layers will not learn high-complexity signals until lower-level ones are sufficiently trained. "Layerwise pre-training" yields *almost identical performance* to "having all the layers trainable from the beginning" (see Figure 7 and Appendix G.4), and sometimes has advantage (Karras et al., 2018).
- When $\mathbf{K}_\ell$ is added to the set of trainable variables (which happens only *once* per layer ℓ), we apply a low-rank SVD decomposition to obtain a warm-start for distilling $\mathbf{K}_\ell$ using $\mathbf{W}_\ell$ for theoretical purpose. This allows us to compute k_ℓ without knowing it in advance; it also helps avoid singularities in $\mathbf{K}_\ell$ which will make the analysis messier. This SVD warm-start is invoked only L times and is only for theoretical purpose. It *serves little role* in learning $G^\star$, and essentially all of the learning is done by SGD.[8]

We specify the choices of thresholds $\mathsf{Thres}_{\ell,\triangle}$ and $\mathsf{Thres}_{\ell,\triangledown}$, and the choices of regularizer weights $\lambda_{3,\ell}, \lambda_{4,\ell}, \lambda_{5,\ell}$ in full in Appendix H. Below, we calculate their values in the special case of Theorem 1.

$$\mathsf{Thres}_{\ell,\triangle} = \frac{\alpha_{\ell-1}^2}{d^{\frac{1}{3C^{\ell-1}}}} \quad , \quad \mathsf{Thres}_{\ell,\triangledown} = \frac{\alpha_\ell^2}{d^{\frac{1}{6C^\ell}}} \quad , \quad \lambda_{3,\ell} \leftarrow \frac{\alpha_\ell^2}{d^{\frac{1}{6C^\ell}}} \quad , \quad \lambda_{4,\ell} \leftarrow \frac{\alpha_\ell^2}{d^{\frac{1}{3C^\ell}}} \quad , \quad \lambda_{5,\ell} = \frac{\alpha_\ell^2}{d^{\frac{1}{2C^\ell}}} \tag{C.1}$$

As mentioned above, our algorithm does not require knowing k_ℓ but learns it on the air. In Line 21 of Algorithm 1, we define $\mathsf{rank}_b(\mathbf{M})$ as the number of singular values of $\mathbf{M}$ with value $\geq b$, and use this to compute k_ℓ. Similarly, α_ℓ and the connection graph $\mathcal{J}_\ell$ can be learned as well, at the expense of complicating the algorithm; but grid searching suffices for theoretical purpose.[9]

D GENERAL DISTRIBUTIONS

Here we define the general distributional assumptions of our work. Given any degree-q homogenous polynomial $f(x) = \sum_{I \in \mathbb{N}^n} a_I \prod_{j \in [n]} x_j^{I_j}$, define $\mathcal{C}_x(f) := \sum_{I \in \mathbb{N}^n} a_I^2$ as the sum of squares of its coefficients.

Input Distribution. We assume the input distribution $\mathcal{D}$ has the following property:

1. (isotropy). There is an absolute constant $c_6 > 0$ such that for every w, we have that

$$\mathbb{E}_{x\sim\mathcal{D}}[|\langle w, x\rangle|^2] \leq c_6 \|w\|_2^2 \quad \text{and} \quad \mathbb{E}_{x\sim\mathcal{D}}[|\langle w, S_1(x)\rangle|^2] \leq c_6 \|w\|_2^2 \tag{D.1}$$

[8]For instance, after $\mathbf{K}_\ell$ is warmed up by SVD, the objective is still around α_ℓ^2 (because deeper layers are not trained yet). It still requires SGD to update each $\mathbf{K}_\ell$ in order to eventually decrease the objective to ε^2.

[9]It suffices to know α_ℓ up to a constant factor α'_ℓ since one can scale the weight matrices as if $G^\star$ uses precisely α'_ℓ. This increases B_ℓ by at most $2^{2^{O(\ell)}}$ so does not affect our result. Gird searching for α'_ℓ takes time $O(\log(1/\varepsilon))^L < \mathsf{poly}(d/\varepsilon)$. Moreover, searching the neural architecture (the connections $\mathcal{J}_\ell$) takes time $2^{O(L^2)} < \mathsf{poly}(d)$.

Algorithm 1 A variant of SGD for DenseNet

Input: Data set $\mathcal{Z}$ of size $N = |\mathcal{Z}|$, network size m, learning rate $\eta > 0$, target error ε.

current target error $\varepsilon_0 \leftarrow B^2$; $\eta_\ell \leftarrow 0$; $\lambda_{3,\ell}, \lambda_{4,\ell}, \lambda_{5,\ell}, \lambda_{6,\ell} \leftarrow 0$; $[\mathbf{R}_\ell]_{i,j} \leftarrow \mathcal{N}(0, 1/(k_\ell)^2)$;
$\mathbf{K}_\ell, \mathbf{W}_\ell \leftarrow \mathbf{0}$ for every $\ell = 2, 3, \dots, L$.
while $\varepsilon_0 \geq \varepsilon$ **do**
  while $\widetilde{\mathbf{Obj}} := \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \geq \frac{1}{4}(\varepsilon_0)^2$ **do**
    for $\ell = 2, 3, \cdots, L$ **do**
      if $\eta_\ell = 0$ and $\widetilde{\mathbf{Obj}} \leq \mathsf{Thres}_{\ell,\triangle}$ **then** — setup learning rate and weight decay
        $\eta_\ell \leftarrow \eta$, $\lambda_{6,\ell} = \frac{(\varepsilon_0)^2}{(\overline{k}_\ell \cdot L \cdot \kappa)^8}$. ◇ $\overline{k}_\ell := \max\{k_j : j \in \mathcal{J}_\ell \wedge j \geq 2\}$
      if $\lambda_{3,\ell} = 0$ and $\widetilde{\mathbf{Obj}} \leq \mathsf{Thres}_{\ell,\triangledown}$ **then**
        set $\lambda_{3,\ell}, \lambda_{4,\ell}, \lambda_{5,\ell}$ according to (C.1)
        $\mathbf{K}_\ell \leftarrow$ INITIAL-DISTILL$_\ell(\mathbf{W}_\ell)$;
    end for
    $x \leftarrow$ a random sample from $\mathcal{Z}$ — stochastic gradient descent (SGD)
    for $\ell = 2, 3, \cdots, L$ **do**
      $\mathbf{K}_\ell \leftarrow \mathbf{K}_\ell - \eta_\ell \nabla_{\mathbf{K}_\ell} \widetilde{\mathbf{Obj}}(x; \mathbf{W}, \mathbf{K})$.
      $\mathbf{W}_\ell \leftarrow \mathbf{W}_\ell - \eta_\ell \nabla_{\mathbf{W}_\ell} \widetilde{\mathbf{Obj}}(x; \mathbf{W}, \mathbf{K}) + \mathsf{noise}$ ◇ *any poly-small Gaussian noise;*
    end for ◇ noise *is for theory purpose to escape saddle points (Ge et al., 2015).*
  end while
  $\varepsilon_0 \leftarrow \varepsilon_0/2$ and $\lambda_{6,\ell} \leftarrow \lambda_{6,\ell}/4$ for every $\ell = 2, 3, \dots, L$.
end while
return $\mathbf{W}$ and $\mathbf{K}$, representing $F(x; \mathbf{W}, \mathbf{K})$.

procedure INITIAL-DISTILL$_\ell(\mathbf{W}_\ell)$ — warm-up for $\mathbf{K}_\ell$, called **only once** for each ℓ=$2, 3, \dots, L$
  $k_\ell \leftarrow \mathrm{rank}_{1/(10\kappa^2)}(\mathbf{W}_{\ell,\triangleleft}^\top \mathbf{W}_{\ell,\ell-1})$.
  $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow k_\ell$-SVD$(\mathbf{W}_{\ell,\triangleleft}^\top \mathbf{W}_{\ell,\ell-1})$,
  return $\mathbf{K}_\ell$ where $\mathbf{K}_{\ell,\triangleleft}^\top = \mathbf{U}\mathbf{\Sigma}^{1/2}$ and $\mathbf{K}_{\ell,\ell-1} = \mathbf{\Sigma}^{1/2}\mathbf{V}$.

2. (hyper-contractivity). There exists absolute constant $c_2 > 0$ such that, for every integer $q \in [1, 2^L]$, there exists value $c_4(q) \geq q$ such that, for every degree q polynomial $f(x)$.

$$\Pr_x \left[|f(x) - \mathbb{E}[f(x)]| \geq \lambda \right] \leq c_4(q) \cdot e^{-\left(\frac{\lambda^2}{c_2 \cdot \mathbf{Var}[f(x)]}\right)^{1/c_4(q)}} \tag{D.2}$$

   If $\mathcal{D} = \mathcal{N}(0, \mathbf{I})$, we have $c_4(q) = O(q)$ (see Lemma P.2b). Note Eq. (D.2) implies there exists value $c_3(q) \geq 1$ such that, for every degree q polynomial $f(x)$, for every integer $p \leq 6$,

$$\mathbb{E}_{x \sim \mathcal{D}} \left[(f(x))^{2p} \right] \leq c_3(q)\, \mathbb{E}\left[(f(x))^2 \right]^p \tag{D.3}$$

   If $\mathcal{D} = \mathcal{N}(0, \mathbf{I})$, we have $c_3(q) \leq O((6q)!)$; and more generally we have $c_3(q) \leq O(c_4(q))^{c_4(q)}$.
3. (degree-preserving). For every integer $q \in [1, 2^L]$, there exists $c_1(q) \geq 1$ such that for every polynomial $P(x)$ with max degree q, let $P_q(x)$ be the polynomial consisting of only the degree-q part of P, the following holds

$$\mathcal{C}_x(P_q) \leq c_1(q) \mathbb{E}_{x \sim \mathcal{D}} P(x)^2 \tag{D.4}$$

   For $\mathcal{D} = N(0, \mathbf{I})$, such inequality holds with $c_1(q) \leq q!$.

Assumptions (isotropy) and (hyper-contractivity) are very common and they are satisfied for sub-gaussian distributions or even heavy-tailed distributions such as $p(x) \propto e^{-x^{0.1}}$. Assumption (degree-preserving) says that data has certain variance along every degree q direction, which is also typical for distributions such like Gaussians or heavy-tailed distributions.

We point out that it is possible to have a distribution to be a mixture of C-distributions satisfying (D.4), where none of the individual distributions satisfies (D.4). For example, the distribution can be a mixture of d-distributions, the i-th distribution satisfies that $x_i = 0$ and other coordinates are i.i.d. standard Gaussian. Thus, non of the individual distribution is degree-preserving, however, the mixture of them is as long as $q \leq d-1$.

It is easy to check some simple distributions satisfy the following parameters.

Proposition D.1. *Our distributional assumptions are satisfied for $c_6 = O(1), c_1(q) = O(q)^q, c_4(q) = O(q), c_3(q) = q^{O(q)}$ when $\mathcal{D} = \mathcal{N}(0, \mathbf{\Sigma}^2)$, where $\mathbf{\Sigma}$ has constant singular values (i.e., in between $\Omega(1)$ and $O(1)$), it is also satisfied for a mixture of arbitrarily many $\mathcal{D}_i = \mathcal{N}(0, \mathbf{\Sigma}_i^2)$'s as long as each $\mathbf{\Sigma}_i$ has constant singular values and for each j, the j-th row: $\|[\mathbf{\Sigma}_i]_j\|_2$ has the same norm for every i.*

In the special case of the main theorem stated in Theorem 1, we work with the above parameters. In our full Theorem 1', we shall make the dependency of those parameters transparent.

E SKETCHED PROOF

Our goal in this section is to make the high level intuitions in Section 1.2 concrete. In this sketched proof let us first ignore the difference between truncated activations and the true quadratic activation. We explain at the end why we need to do truncation.

Let us now make the intuition concrete. We plan to prove by induction, so let us assume for now that the regression error is ε^2 and for every layer $\ell' \leq \ell$, the function $S_{\ell'}$ is already learned correct up to error $\varepsilon/\alpha_{\ell'} \leq \varepsilon/\alpha_\ell$. Let us now see what will happen if we continue to decrease the regression error to $(\widehat{\varepsilon})^2$ for some $\widehat{\varepsilon} < \varepsilon$. We want to show

- $S_{\ell+1}$ can be learned to error $\frac{\widehat{\varepsilon}}{\alpha_{\ell+1}}$ (forward feature learning),
- $S_{\ell'}$ can be backward corrected to error $\frac{\widehat{\varepsilon}}{\alpha_{\ell'}}$ for each $\ell' \leq \ell$ (backward feature correction).

Note that due to error between $S^\star_{\ell'}$ and $S_{\ell'}$ for $\ell' \leq \ell$, when we use them to learn the $(\ell+1)$-th layer, namely $\alpha_{\ell+1} G^\star_{\ell+1} = \alpha_{\ell+1}\sigma\big(\mathbf{W}^\star_{\ell+1,\ell}\sigma(S^\star_\ell) + \cdots\big)$, we cannot learn it correct for any error better than $\varepsilon/\alpha_\ell \times \alpha_{\ell+1}$. Fortunately, using information gap, we have $\varepsilon/\alpha_\ell \times \alpha_{\ell+1} < \varepsilon$, so if we continue to decrease the regression loss to $(\widehat{\varepsilon})^2$, we can at least "hope for" learning some $\alpha_{\ell+1} F_{\ell+1} \approx \alpha_{\ell+1} G^\star_{\ell+1}$ up to error $\widehat{\varepsilon}$ as long as $\widehat{\varepsilon} > \varepsilon/\alpha_\ell \times \alpha_{\ell+1}$. (This implies $S_{\ell+1} \approx S^\star_{\ell+1}$ up to error $\frac{\widehat{\varepsilon}}{\alpha_{\ell+1}}$.) Moreover, if we have learned $\alpha_{\ell+1} G^\star_{\ell+1}$ to error $\widehat{\varepsilon}$ and the regression error is $(\widehat{\varepsilon})^2$, then the sum of the lower-order terms $\sum_{\ell' \leq \ell} \alpha_{\ell'} G^\star_{\ell'}$ is also of error $\widehat{\varepsilon} < \varepsilon$, so by induction the lower-level features also get improved.

There are several major obstacles for implementing the above intuition, as we summarized blow.

Function value v.s. coefficients. To actually implement the approach, we first notice that $F_{\ell+1}$ is a polynomial of *maximum* degree $2^{\ell+1}$, however, it also has a lot of lower-degree monomials. Obviously, the monomials up to degree 2^ℓ can *also* be learned in lower layers such as F_ℓ. As a result, it is *impossible* to derive $F_{\ell+1} \approx G^\star_{\ell+1}$ simply from $F \approx G^\star$. Using a concrete example, the learner network could instead learn $F_{\ell+1}(x) \approx G^\star_{\ell+1}(x) - F'(x)$ for some error function $F'(x)$ of degree 2^ℓ, while satisfying $F_\ell(x) \approx G^\star_\ell(x) + \frac{\alpha_{\ell+1}}{\alpha_\ell} F'(x)$.

Our critical lemma (see Theorem 2 or Lemma L.1) proves that this *cannot* happen when we train the network using SGD. We prove it by first focusing on all the monomials in $F_{\ell+1}$ of degree $2^\ell + 1, \dots, 2^{\ell+1}$, which are not learnable at lower-level layers. One might hope to use this observation to show that it must be the case $\widehat{F}_{\ell+1}(x) \approx \widehat{G^\star}_{\ell+1}(x)$, where the $\widehat{F}_{\ell+1}$ contains all the monomials in $F_{\ell+1}$ of degree $2^\ell + 1, \dots, 2^{\ell+1}$ and similarly for $\widehat{G^\star}_{\ell+1}$.

Unfortunately, this approach fails again. Even in the ideal case when we already have $F_{\ell+1} \approx G^\star_{\ell+1} \pm \varepsilon'$, it still *does not* imply $\widehat{F}_{\ell+1} \approx \widehat{G^\star}_{\ell+1} \pm \varepsilon'$. One counterexample is the polynomial $\sum_{i\in[d]} \frac{\varepsilon'}{\sqrt{d}}(x_i^2 - 1)$ where $x_i \sim \mathcal{N}(0,1)$. This polynomial is ε'-close to zero, however, its degree-2 terms $\frac{\varepsilon'}{\sqrt{d}} x_i^2$ when added up is actually $\sqrt{d}\varepsilon' \gg \varepsilon'$. In worst case, such difference leads to complexity $d^{\Omega(2^L)}$ for learning the degree 2^L target function, leading to an unsatisfying bound.

To correct this, *as a first step*, we count the monomial *coefficients* instead of the actual function value. The main observation is that, if the regression error is already $(\widehat{\varepsilon})^2$, then[10]

- (Step 1). The top-degree (i.e., degree-$2^{\ell+1}$) coefficients of the monomials in $F_{\ell+1}$ is ε' close to that of $G^\star_{\ell+1}$ in terms of ℓ_2-norm, for $\varepsilon' = \frac{\widehat{\varepsilon}}{\alpha_{\ell+1}}$,

without sacrificing a dimension factor (and only sacrificing a factor that depends on the degree). Taking the above example, the ℓ_2 norm of the coefficients of $\frac{\varepsilon'}{\sqrt{d}} x_i^2$ is indeed ε', which does not grow with the dimension d.

Symmetrization. As a *second step*, one would like to show that Step 1 — namely, $F_{\ell+1}$ is learned so that its coefficients of degree $2^{\ell+1}$ monomials match $G^\star_{\ell+1}$ — implies $\mathbf{W}_{\ell+1,\ell}$ is close to $\mathbf{W}^\star_{\ell+1,\ell}$ in some measure.

Indeed, all of the top-degree (i.e., degree $2^{\ell+1}$) monomials in $F_{\ell+1}$ come from $\sigma(\mathbf{W}_{\ell+1,\ell}\sigma(\mathbf{R}_\ell \widehat{S}_\ell))$, where $\widehat{S}_\ell$ consists of all the top-degree (i.e., degree-$2^{\ell-1}$) monomials in S_ℓ. At the same time, inductive assumption says S_ℓ is close to $S^\star_\ell$, so the coefficients of $\widehat{S}_\ell$ are also close to $\widehat{S}^\star_\ell$. In other words, we arrive at the following question:

*If (1) the coefficients of $\widehat{S}_\ell(x)$, in ℓ_2-norm, are ε'-close to that of $\widehat{S}^\star_\ell(x)$, and (2) the coefficients of $\sigma(\mathbf{W}_{\ell+1,\ell}\sigma(\mathbf{R}_\ell \widehat{S}_\ell))$, in ℓ_2-norm, are ε'-close to that of $\sigma(\mathbf{W}^\star_{\ell+1,\ell}\sigma(\widehat{S}^\star_\ell))$, **then**, does it mean that $\mathbf{W}_{\ell+1,\ell}$ is ε'-close to $\mathbf{W}^\star_{\ell+1,\ell}$ in some measure?*

The answer to this question is very delicate, due to the huge amount of "symmetricity" in a degree-4 polynomial. Note that both the following two quantities

$$\sigma(\mathbf{W}^\star_{\ell+1,\ell}\sigma(\widehat{S}^\star_\ell)) = \left\|\mathbf{W}^\star_{\ell+1,\ell}(\mathbf{I}\otimes\mathbf{I})(\widehat{S}^\star_\ell \otimes \widehat{S}^\star_\ell)\right\|^2$$
$$\sigma(\mathbf{W}_{\ell+1,\ell}\sigma(\mathbf{R}_\ell\widehat{S}_\ell)) = \left\|\mathbf{W}_{\ell+1,\ell}(\mathbf{R}_\ell\otimes\mathbf{R}_\ell)(\widehat{S}_\ell \otimes \widehat{S}_\ell)\right\|^2$$

are degree-4 polynomials over $\widehat{S}^\star_\ell$ and $\widehat{S}_\ell$ respectively.

In general, when $x \in \mathbb{R}^d$ and $\mathbf{M}, \mathbf{M}' \in \mathbb{R}^{d^2\times d^2}$, suppose $(x\otimes x)^\top \mathbf{M}(x\otimes x)$ is ε'-close to $(x\otimes x)^\top \mathbf{M}'(x\otimes x)$ in terms of coefficients when we view them as degree 4 polynomials, this *does not* imply that $\mathbf{M}$ is close to $\mathbf{M}'$ at all. Indeed, if we increase $\mathbf{M}_{(1,2),(3,4)}$ by 10^{10} and decrease $\mathbf{M}_{(1,3),(2,4)}$ by 10^{10}, then $(x\otimes x)^\top \mathbf{M}(x\otimes x)$ remains the same.

One may consider a simple fix: define a symmetric version of tensor product — the "$*$ product" in Definition I.2 — which makes sure $x * x$ only has $\binom{d+1}{2}$ dimensions, each corresponding to the $\{i,j\}$-th entry for $i \leq j$. This makes sure $\mathbf{M}_{\{1,2\},\{3,4\}}$ is the same entry as $\mathbf{M}_{\{2,1\},\{4,3\}}$. Unfortunately, this simple fix *does not* resolve all the "symmetricity": for instance, $\mathbf{M}_{\{1,2\},\{3,4\}}$ and $\mathbf{M}_{\{1,3\},\{2,4\}}$ are still difference entries.

For reasons explained above, we *cannot* hope to derive $\mathbf{W}_{\ell+1,\ell}$ and $\mathbf{W}^\star_{\ell+1,\ell}$ are ε'-close. However, they should still be close after "twice symmetrizing" their entries. For this purpose, we introduce a "twice symmetrization" operator $\mathbf{Sym}$ on matrices, and eventually derive that:[11]

- (Step 2). $\mathbf{W}_{\ell+1,\ell}$ and $\mathbf{W}^\star_{\ell+1,\ell}$ are close under the following notation (for $\varepsilon' \approx \frac{\widehat{\varepsilon}}{\alpha_{\ell+1}}$)

$$\mathbf{Sym}\Big((\mathbf{R}_\ell * \mathbf{R}_\ell)^\top (\mathbf{W}_{\ell+1,\ell})^\top \mathbf{W}_{\ell+1,\ell}(\mathbf{R}_\ell * \mathbf{R}_\ell)\Big) \approx \mathbf{Sym}\Big((\mathbf{I}*\mathbf{I})^\top (\mathbf{W}_{\ell+1,\ell})^\top \mathbf{W}^\star_{\ell+1,\ell}(\mathbf{I}*\mathbf{I})\Big) \pm \varepsilon' \tag{E.1}$$

We then use (E.1) to non-trivially derive that $\sigma(\mathbf{W}_{\ell+1,\ell}\sigma(\mathbf{R}_\ell S_\ell))$ is close to $\sigma(\mathbf{W}^\star_{\ell+1,\ell}\sigma(S^\star_\ell))$, since S_ℓ is close to $S^\star_\ell$ as we have assumed. This implies the monomials in $F_{\ell+1}$ of degree $2^\ell + 2^{\ell-1} + 1, \dots, 2^{\ell+1}$ match that of $G^\star_{\ell+1}$. It is a good start, but there are lower-degree terms to handle.

Low-degree terms. Without loss of generality, we assume the next highest degree is $2^\ell + 2^{\ell-2}$. (It cannot be $2^\ell + 2^{\ell-1}$ since we assumed skip links.) Such degree monomials must *either* come from

[10]Concretely, this can be found in (L.7) in our proof of Lemma L.1.

[11]The operator $\mathbf{Sym}(\mathbf{M})$ essentially averages out all the $\mathbf{M}_{i,j,k,l}$ entries when $\{i,j,k,l\}$ comes from the same unordered set (see Definition I.3). The formal statement of (E.1) is in Eq. (L.9) of Appendix L.3.

$\sigma(\mathbf{W}^\star_{\ell+1,\ell}\sigma(S^\star_\ell))$ — which we have just shown it is close to $\sigma(\mathbf{W}_{\ell+1,\ell}\sigma(\mathbf{R}_\ell S_\ell))$ — *or* come from the cross term

$$(S^\star_\ell * S^\star_\ell)^\top \left(\mathbf{W}^\star_{\ell+1,\ell}\right)^\top \mathbf{W}^\star_{\ell+1,\ell-2}(S^\star_{\ell-2} * S^\star_{\ell-2})$$

Using a similar analysis, we can first show that the learned function $F_{\ell+1}$ matches in coefficients the top-degree (i.e., degree $2^\ell + 2^{\ell-2}$) monomials in the above cross term. Then, we wish to argue that the learned $\mathbf{W}_{\ell+1,\ell-2}$ is close to $\mathbf{W}^\star_{\ell+1,\ell-2}$ in some measure.

In fact, this time the proof is much simpler: the matrix $(\mathbf{W}^\star_{\ell+1,\ell})^\top\mathbf{W}^\star_{\ell+1,\ell-2}$ is not symmetric, and therefore we do not have the "twice symmetrization" issue as argued above. Therefore, we can directly conclude that the non-symmetrized closeness, or in symbols,[12]

- (Step 3). $\mathbf{W}_{\ell+1,\ell-2}$ and $\mathbf{W}^\star_{\ell+1,\ell-2}$ are close in the following sense (for $\varepsilon' \approx \frac{\widehat{\varepsilon}}{\alpha_{\ell+1}}$)

$$(\mathbf{R}_{\ell-2} * \mathbf{R}_{\ell-2})^\top (\mathbf{W}_{\ell+1,\ell-2})^\top\mathbf{W}_{\ell+1,\ell} (\mathbf{R}_\ell * \mathbf{R}_\ell) \approx (\mathbf{I} * \mathbf{I})^\top (\mathbf{W}^\star_{\ell+1,\ell-2})^\top\mathbf{W}^\star_{\ell+1,\ell} (\mathbf{I} * \mathbf{I}) \tag{E.2}$$

We can continue in this fashion for all the remaining degrees until degree $2^\ell + 1$.

Moving from W to K: Part I. So far Steps 2&3 show that $\mathbf{W}_{\ell+1,j}$ and $\mathbf{W}^\star_{\ell+1,j}$ are close in some measure. We hope to use this to show that the function $S_{\ell+1}$ is close to $S^\star_{\ell+1}$ and proceed the induction. However, if we use the matrix $\mathbf{W}_{\ell+1}$ to define $S_{\ell+1}$ (instead of introducing the notation $\mathbf{K}_{\ell+1}$), then $S_{\ell+1}$ may have huge error compare to $S^\star_{\ell+1}$.

Indeed, even in the ideal case that $(\mathbf{W}_{\ell+1,\ell})^\top\mathbf{W}_{\ell+1,\ell} \approx (\mathbf{W}^\star_{\ell+1,\ell})^\top\mathbf{W}^\star_{\ell+1,\ell} + \varepsilon'$, this only guarantees that $\mathbf{W}_{\ell+1,\ell} \approx \mathbf{U}\mathbf{W}^\star_{\ell+1,\ell} + \sqrt{\varepsilon'}$ for some column orthonormal matrix $\mathbf{U}$. This is because the inner dimension m of $(\mathbf{W}_{\ell+1,\ell})^\top\mathbf{W}_{\ell+1,\ell}$ is much larger than that the inner dimension $k_{\ell+1}$ of $\mathbf{W}^\star_{\ell+1,\ell}$.[13] This $\sqrt{\varepsilon'}$ error can lie in the orthogonal complement of $\mathbf{U}$.

To fix this issue, we need to "reduce" the dimension of $\mathbf{W}_{\ell+1,\ell}$ back to $k_{\ell+1}$ to reduce error. This is why we need to introduce the $\mathbf{K}_{\ell+1,\ell}$ matrix of rank $k_{\ell+1}$, and add a regularizer to ensure that $\mathbf{K}^\top_{\ell+1,\ell}\mathbf{K}_{\ell+1,\ell}$ approximates $(\mathbf{W}_{\ell+1,\ell})^\top\mathbf{W}_{\ell+1,\ell}$. (This can be reminiscent of knowledge distillation used in practice (Hinton et al., 2015).) This knowledge distillation step decreases the error back to $\varepsilon' \ll \sqrt{\varepsilon'}$, so now $\mathbf{K}_{\ell+1,\ell}$ truly becomes ε' close to $\mathbf{W}^\star_{\ell+1,\ell}$ up to column orthonormal transformation.[14] We use this to proceed and conclude the closeness of $S_{\ell+1}$. This is done in Section L.6.

Moving from W to K: Part II. Now suppose the leading term (E.1) holds without the $\mathbf{Sym}$ operator (see Footnote 14 for how to get rid of it), and suppose the cross term (E.2) also holds. The former means "$(\mathbf{W}_{\ell+1,\ell})^\top\mathbf{W}_{\ell+1,\ell}$ is close to $(\mathbf{W}^\star_{\ell+1,\ell})^\top\mathbf{W}^\star_{\ell+1,\ell}$" and the latter means "$(\mathbf{W}_{\ell+1,\ell-2})^\top\mathbf{W}_{\ell+1,\ell}$ is close to $(\mathbf{W}^\star_{\ell+1,\ell-2})^\top\mathbf{W}^\star_{\ell+1,\ell}$". These two together, still does not imply that "$(\mathbf{W}_{\ell+1,\ell-2})^\top\mathbf{W}_{\ell+1,\ell-2}$ is close to $(\mathbf{W}^\star_{\ell+1,\ell-2})^\top\mathbf{W}^\star_{\ell+1,\ell-2}$", since the error of $\mathbf{W}_{\ell+1,\ell-2}$ can also lie on the orthogonal complement of $\mathbf{W}_{\ell+1,\ell}$. This error can be arbitrary large when $\mathbf{W}_{\ell+1,\ell}$ is not full rank.

This means, the learner network can still make a lot of error on the $\ell+1$ layer, *even when it already learns all degree $> 2^\ell$ monomials correctly*. To resolve this, we again need to use the regularizer to ensure closeness between $\mathbf{W}_{\ell,\ell-2}$ to $\mathbf{K}_{\ell,\ell-2}$. It "reduces" the error because by enforcing $\mathbf{W}_{\ell+1,\ell-2}$ being close to $\mathbf{K}_{\ell+1,\ell}$, it must be of low rank — thus the "arbitrary large error" from the orthogonal complement cannot exist. Thus, *it is important that we keep $\mathbf{W}_\ell$ being close to the low rank counterpart $\mathbf{K}_\ell$, and update them together gradually.*

[12] The formal statement of this can be found in (L.12).

[13] Recall that without RIP-type of strong assumptions, such over-parameterization m is somewhat necessary for a neural network with quadratic activations to perform optimization without running into saddle points, and is also used in (Allen-Zhu et al., 2019a).

[14] In fact, things are still trickier than one would expect. To show "$\mathbf{K}_{\ell+1,\ell}$ close to $\mathbf{W}^\star_{\ell+1,\ell}$," one needs to first have "$\mathbf{W}_{\ell+1,\ell}$ close to $\mathbf{W}^\star_{\ell+1,\ell}$", but we do not have that due to the twice symmetrization issue from (E.1). Instead, our approach is to first use (E.2) to derive that there exists some matrix $\mathbf{P}$ satisfying "$\mathbf{P}\mathbf{K}_{\ell+1,\ell}$ is close to $\mathbf{P}\mathbf{W}^\star_{\ell+1,\ell}$" and "$\mathbf{P}^{-1}\mathbf{K}_{\ell+1,\ell-2}$ is close to $\mathbf{P}\mathbf{W}^\star_{\ell+1,\ell-2}$". Then, we plug this back to (E.1) to derive that $\mathbf{P}$ must be close to $\mathbf{I}$. This is precisely why we need a skip connection.

Remark E.1. If we have "weight sharing", meaning forcing $\mathbf{W}_{\ell+1,\ell-2} = \mathbf{W}_{\ell+1,\ell}$, then we immediately have $(\mathbf{W}_{\ell+1,\ell-2})^\top \mathbf{W}_{\ell+1,\ell-2}$ is close to $(\mathbf{W}^\star_{\ell+1,\ell-2})^\top \mathbf{W}^\star_{\ell+1,\ell-2}$, so we do not need to rely on "$\mathbf{W}_{\ell+1,\ell-2}$ is close to $\mathbf{K}_{\ell+1,\ell}$" and this can make the proof much simpler.

To conclude, by introducing matrices $\mathbf{K}_{\ell+1}$ and enforcing the low-rank $\mathbf{K}_{\ell+1}^\top \mathbf{K}_{\ell+1}$ to stay close to $\mathbf{W}_{\ell+1}^\top \mathbf{W}_{\ell+1}$, we have distilled the knowledge from $\mathbf{W}_{\ell+1}$ and can derive that[15]

- (Step 4). Up to unitary transformations, $\mathbf{K}_{\ell+1}$ is close to $\mathbf{W}^\star_{\ell+1}$ with error $\varepsilon' \approx \frac{\widehat{\varepsilon}}{\alpha_{\ell+1}}$; and this also implies $S_{\ell+1}$ is close to $S^\star_{\ell+1}$ with error ε' as desired.

Empirical v.s. Population loss. We have given a sketched proof to our intuition focusing on the case when F is in the *population case* (i.e., under the true distribution $\mathcal{D}$), since properties such as degree preserving Property D.4 is *only* true for the population loss. Indeed, if we only have $\mathsf{poly}(d)$ samples, the empirical distribution can not be degree-preserving at all for any $2^\ell = \omega(1)$.

One would like to get around it by showing that, when F is close to $G^\star$ only on the *training* data set $\mathcal{Z}$, then the aforementioned closeness between S_ℓ and $S^\star_\ell$ still holds for the *population* case. This turns out to be a challenging task.

One naive idea would be to show that $\mathbb{E}_{x\sim\mathcal{Z}}\left(F(x) - G^\star(x)\right)^2$ is close to $\mathbb{E}_{x\sim\mathcal{D}}\left(F(x) - G^\star(x)\right)^2$ for any networks weights $\mathbf{W}, \mathbf{K}$. However, this *cannot* work at all. Since $F(x) - G^\star(x)$ is a degree 2^L polynomial, we know that for a fixed F, $\mathbb{E}_{x\sim\mathcal{Z}}\left(F(x) - G^\star(x)\right)^2 \approx \mathbb{E}_{x\sim\mathcal{D}}\left(F(x) - G^\star(x)\right)^2 \pm \varepsilon$ only holds with probability $e^{-(N\log(1/\varepsilon))^{\frac{1}{2^L}}}$, where $|\mathcal{Z}| = N$. This implies, in order for it to hold *for all* possible $\mathbf{W}, \mathbf{K}$, we need at least $N = \Omega(d^{2^L})$ many samples, which is too bad.

We took an alternative approach. We truncated the learner network from F to $\widetilde{F}$ using truncated quadratic activations (recall 2.2): if the intermediate value of some layers becomes larger than some parameter B', then we truncate it to $\Theta(B')$. Using this operation, we can show that the function output of $\widetilde{F}$ is always bounded by a small value. Using this, one could show that $\mathbb{E}_{x\sim\mathcal{Z}}\big(\widetilde{F}(x) - G^\star(x)\big)^2 \approx \mathbb{E}_{x\sim\mathcal{D}}\big(\widetilde{F}(x) - G^\star(x)\big)^2 \pm \varepsilon$.

But, why is $F(x)$ necessarily close to $\widetilde{F}(x)$, especially on the training set $\mathcal{Z}$? If some of the $x \in \mathcal{Z}$ is too large, then $\big(\widetilde{F}(x) - F(x)\big)^2$ can be large as well. Fortunately, we show during the training process, the neural network actually has *implicit self-regularization* (as shown in Corollary L.3e): the *intermediate values* such as $\|S_\ell(x)\|^2$ stay away from $2B$ for most of the $x \sim \mathcal{D}$. This ensures that $\mathbb{E}_{x\sim\mathcal{D}}(F(x) - \widetilde{F}(x))^2$ is small in the population loss.

This implicit regularization is elegantly maintained by SGD where the weight matrix does not move too much at each step, this is another place where we need gradual training instead of one-shot learning.

Using this property we can conclude that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}\Big(\widetilde{F}(x) - G^\star(x)\Big)^2 \text{ is small} \iff \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\Big(\widetilde{F}(x) - G^\star(x)\Big)^2 \text{ is small} \iff \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left(F(x) - G^\star(x)\right)^2 \text{ is small,}$$

which allows us to interchangeably apply all the aforementioned arguments both on the empirical truncated loss and on the population loss.

F More on Related Works

Historically, due to the extreme non-convexity, for theoretical studies, the hierarchical structure of a neural network is typically a *disadvantage* for efficient training. For example, multi-layer linear network (Hardt & Ma, 2016; Du & Hu, 2019) has no advantage over linear functions in representation power, but it already creates huge obstacle for analyzing the training properties.

With such difficulties, it is perhaps not surprising that existing theory in the *efficient learning* regime of neural networks, either

[15] The formal statement can be found in (L.21).

- reduce multi-layer neural networks to non-hierarchical models such as kernel methods (a.k.a. neural kernels) (Daniely, 2017; Huang & Yau, 2019; Allen-Zhu et al., 2019c; Li & Liang, 2018; Allen-Zhu & Li, 2019b; Allen-Zhu et al., 2019b; Du et al., 2018b; Arora et al., 2019b;a; Zou et al., 2018; Du et al., 2018a; Daniely et al., 2016; Jacot et al., 2018; Ghorbani et al., 2019; Cao & Gu, 2019; Li et al., 2019a; Hanin & Nica, 2019; Yang, 2019; Zou & Gu, 2019; Shankar et al., 2020; Nachum & Yehudayoff, 2020; Allen-Zhu et al., 2019a; Allen-Zhu & Li, 2019a), or
- focus on two-layer networks (Daniely & Malach, 2020; Kawaguchi, 2016; Soudry & Carmon, 2016; Xie et al., 2016; Ge et al., 2017; Soltanolkotabi et al., 2017; Tian, 2017; Brutzkus & Globerson, 2017; Zhong et al., 2017; Li & Yuan, 2017; Boob & Lan, 2017; Li et al., 2018; Vempala & Wilmes, 2018; Ge et al., 2018; Bakshi et al., 2018; Oymak & Soltanolkotabi, 2019; Yehudai & Shamir, 2019; Zhang et al., 2018; Li & Liang, 2017; Li et al., 2016; Li & Dou, 2020; Allen-Zhu & Li, 2021) which *do not have the deep hierarchical structure*.

Learning two-layer network (Kawaguchi, 2016; Soudry & Carmon, 2016; Xie et al., 2016; Ge et al., 2017; Soltanolkotabi et al., 2017; Tian, 2017; Brutzkus & Globerson, 2017; Zhong et al., 2017; Li & Yuan, 2017; Boob & Lan, 2017; Li et al., 2018; Vempala & Wilmes, 2018; Ge et al., 2018; Bakshi et al., 2018; Oymak & Soltanolkotabi, 2019; Yehudai & Shamir, 2019; Li & Yuan, 2017; Li & Liang, 2017; Li et al., 2016; Li & Dou, 2020; Allen-Zhu & Li, 2021). There is a rich history of works considering the learnability of neural networks trained by SGD. However, as we mentioned before, many of these works only focus on network with 2 layers or only one layer in the network is trained. Hence, the learning process is not hierarchical in the language of this paper. Note even those two-layer results that study feature learning as a process (such as (Daniely & Malach, 2020; Li et al., 2020; Allen-Zhu & Li, 2021)) do not cover how the features of second layer can help backward correct the first layer, not to say repeating them for multiple layers may only give rise to layer-wise training as opposed to the full hierarchical learning.

Neural tangent/compositional kernel (Allen-Zhu et al., 2019c; Li & Liang, 2018; Allen-Zhu & Li, 2019b; Allen-Zhu et al., 2019b; Du et al., 2018b; Arora et al., 2019b;a; Zou et al., 2018; Du et al., 2018a; Daniely et al., 2016; Jacot et al., 2018; Ghorbani et al., 2019; Li et al., 2019a; Hanin & Nica, 2019; Yang, 2019; Cao & Gu, 2019; Shankar et al., 2020). There is a rich literature approximating the learning process of over-parameterized networks using the neural tangent kernel (NTK) approach, where the kernel is defined by the gradient of a neural network at

		CIFAR-10 accuracy		CIFAR-100 accuracy		training time
		single model	ensemble	single model	ensemble	
Hierarchical learning	WRN-16-10	96.27%	96.8%	80.28%	83.18%	2.5 GPU hour (V100)
	WRN-22-10	96.59%	97.12%	81.43%	84.33%	3 GPU hour (V100)
	quadratic WRN-16-10	94.68%	95.65%	75.31%	79.01%	3 GPU hour (V100)
	quadratic WRN-22-10	95.08%	95.97%	75.65%	79.97%	3.5 GPU hour (V100)
Kernel methods	neural compositional kernel * with ZCA preprocessing	89.8%	89.8%	68.2%	68.2%	~1000 GPU hour
	neural tangent kernel (NTK) **	81.40%	81.40%	-	-	~1000 GPU hour
	(+ random preprocessing)	(88.36%)	-	-	-	~1000 GPU hour
	neural Gaussian process kernel **	82.20%	82.20%	-	-	~1000 GPU hour
	(+ random preprocessing)	(88.92%)	-	-	-	~1000 GPU hour
	finite-width NTK for WRN-10-10	72.33%	75.26%	-	-	20.5 GPU hour (TitanV)
	(+ ZCA preprocessing)	(76.94%)	(80.21%)	-	-	20.5 GPU hour (TitanV)

Figure 11: Comparison between ReLU networks, quadratic networks, and several optimized kernel methods (* for (Shankar et al., 2020) and ** for (Li et al., 2019b)). Details in Appendix G.3.

Take-away messages: Quadratic networks perform comparable to ReLU, and better and *much faster* than the best-known kernel methods. Finite-width NTK (Allen-Zhu et al., 2019c) accuracy is much worse than its counterparts in hierarchical learning, showing its insufficiency for understanding the ultimate power of neural networks.
Note 1: Kernel methods usually cannot benefit from ensemble since it is typically strictly convex. Random preprocessing in principle may help if one runs it multiple times; but we expect the gain to be little. Ensemble helps on finite-width NTK (linear function over random feature mappings) because the feature space is re-randomized multiple times, so ensemble actually increases the number of features.
Note 2: Our obtained accuracies using quadratic networks may be of independent interests: networks with quadratic activations have certain practical advantage especially in cryptographic applications (Mishra et al., 2020).

random initialization (Jacot et al., 2018). Others also study neural compositional kernel through a random neural network (Daniely et al., 2016; Shankar et al., 2020). One *should not confuse* these hierarchically-defined kernels with hierarchical learning. As we pointed out, see also Bengio (2009), hierarchical learning means that each layer *learns* a combination of previous *learned* layers. In these cited kernel methods, such combinations are *prescribed* by the random initialization and *not learned* during training. As our negative result shows, for certain learning tasks, hierarchical learning is superior than any kernel method, so the *hierarchically-learned* features are indeed *superior* than any (even hierarchically) prescribed features. (See also experiments in Figure 11.)

Three-layer result (Allen-Zhu et al., 2019a). This paper shows that 3-layer neural networks can learn the so-called "second-order NTK," which is not a linear model; however, second-order NTK is also learnable by doing a nuclear-norm constrained linear regression over the feature mappings defined by the initialization of a neural network. Thus, the underlying learning process is still not truly hierarchical.

Three-layer ResNet result (Allen-Zhu & Li, 2019a). This paper shows that 3-layer ResNet can at least perform some *weaker* form of implicit hierarchical learning, with better sample or time complexity than any kernel method or linear regression over feature mappings. Our result is greatly inspired by (Allen-Zhu & Li, 2019a), but with several major differences.

First and foremost, the result (Allen-Zhu & Li, 2019a) is only forward feature learning *without* backward feature correction. It is a weaker version of hierarchical learning.

Second, the result (Allen-Zhu & Li, 2019a) can also be achieved by non-hierarchical methods such as *simply* applying kernel method twice.[16]

Third, we prove in this paper a "poly vs. super-poly" running time separation, which is what one refers to as "efficient vs non-efficient" in traditional theoretical computer science. The result (Allen-Zhu & Li, 2019a) is regarding "poly vs. bigger poly" in the standard regime with constant output dimension.

Fourth, as we illustrate in Section E, the *major* technical difficulty of this paper comes from showing how the ***hidden features*** *are learned hierarchically*. In contrast, the intermediate features in (Allen-Zhu & Li, 2019a) are directly connected to the outputs so are not hidden.

Fifth, without backward feature correction, the error incurred from lower layers in (Allen-Zhu & Li, 2019a) cannot be improved through training (see Footnote 16), and thus their theory does not lead to arbitrarily small generalization error like we do. This also prevents (Allen-Zhu & Li, 2019a) from going beyond $L = 3$ layers.

Separation between multi-layer networks and shallower learners. Prior results such as (Eldan & Shamir, 2016; Telgarsky, 2016) separate the representation power of multi-layer networks from shallower learners (without efficient training guarantee), and concurrent results (Daniely & Malach, 2020; Li et al., 2020) separate the power of *two-layer* neural networks from kernel methods with efficient training guarantees. As we emphasized, proving separation is not the main message of this paper, and we focus on studying ***how*** deep learning perform ***efficient hierarchical learning*** when $L = \omega(1)$.

Other theoretical works on hierarchical learning (Arora et al., 2014; Mossel, 2016). There are other theoretical works to perform provable hierarchical learning. The cited works (Arora et al., 2014; Mossel, 2016) propose *new, discrete learning algorithms* to learn certain hierarchical representations. In contrast, the main goal of our work is to explore how deep learning (multi-layer neural networks) can perform hierarchical learning simply by applying SGD on the training objective, which is the most dominant hierarchical learning framework in practice nowadays.

[16] Recall the target functions in (Allen-Zhu & Li, 2019a) are of the form $F(x) + \alpha \cdot G(F(x))$ for $\alpha \ll 1$, and they were proved learnable by 3-layer ResNet up to generalization error α^2 in (Allen-Zhu & Li, 2019a). Here is a *simple* alternative two-step kernel method to achieve this same result. First, learn some $F'(x)$ that is α-close to $F(x)$ using kernel method. Then, treat $(x, F'(x))$ as the input to learn two more functions F, G using kernel method, to ensure that $F(x) + \alpha G(F'(x))$ is close to the target. This incurs a fixed generalization error of magnitude α^2. Note in particular, both this two-step kernel method as well as the 3-layer ResNet analysis from (Allen-Zhu & Li, 2019a) *never* guarantees to learn any function $F''(x)$ that is α^2 close to $F(x)$, and therefore the "intermediate features" do not get improved. In other words, there is no backward feature correction.

G Details on Empirical Evaluations

Our experiments use the CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 2009). In one of our experiments, we also use what we call CIFAR-2, which is to re-group the 10 classes of CIFAR-10 into two classes (bird,cat,deer,dog,horse vs. the rest) and is a binary classification task. We adopt standard data augmentations: random crops, random flips, and normalization; but for adversarial training, we removed data normalization. For some of the experiments (to be mentioned later), we also adopt random Cutout augmentation (Shankar et al., 2020) to obtain higher accuracy.

We note there is a distinction between the original ResNet (He et al., 2016) and the later more popularized (pre-activation) ResNet (Zagoruyko & Komodakis, 2016). We adopt the later because it is the basic block of WideResNet or WRN (Zagoruyko & Komodakis, 2016). Recall ResNet-34 has 1 convolutional layers plus 15 basic blocks each consisting of 2 convolutional layers. We have also implemented VGG19 and VGG13 in some of our experiments, and they have 16 and 10 convolutional layers respectively.

All the training uses stochastic gradient descent (SGD) with momentum 0.9 and batch size 125, unless otherwise specified.

G.1 Feature Visualization on ResNet-34: Figure 2

We explain how Figure 2 is obtained. Throughout this paper, we adopt the simplest possible feature visualization scheme for ResNet: that is, start from a *random* 32x32 image, then repeatedly take its gradient so as to maximize a given neuron in some layer. We perform gradient updates on the image for 2000 steps, with weight decay factor 0.003.

Note however, if the network is trained normally, then the above feature visualization process outputs images that appear like high-frequency noise (for reasons of this, see (Allen-Zhu & Li, 2021)). Therefore, in order to obtain Figure 2 we run *adversarial training*. The specific adversarial attacker that we used in the training is ℓ_2 PGD perturbation plus Gaussian noise suggested by (Salman et al., 2019). That is, we randomly perturb the input twice each with Gaussian noise $\sigma = 0.12$ per coordinate, and then perform 4 steps of PGD attack with ℓ_2 radius $r = 0.5$. We call this $\ell_2(0.5, 0.12)$ attacker for short.

Recall ResNet-34 has 3 parts, the first part has 11 convolutional layers consisting of 16 channels each; the second part has 10 convolutional layers consisting of 32 channels each (but we plot 24 of them due to space limitation); the third part has 10 convolutional layers consisting of 64 channels each (but we plot 40 of them due to space limitation).

To be consistent with the theoretical results of this paper, to obtain Figure 2, we have modified ResNet-34 to make it more like DenseNet: the network output is now a linear functions (AvgPool+FC) over all the 16 blocks (15 basic blocks plus the first convolutional layer). This modification will not change the final accuracy by much. Without this modification, the feature visualizations will be similar; but with this modification, we can additionally see the "incremental feature change" in each of the 3 parts of ResNet-34.

G.2 Toy Experiment on AlexNet: Figure 3

We explain how Figure 3 is obtained. Recall AlexNet has 5 convolutional layers with ReLU activation, connected sequentially. The output of AlexNet is a linear function over its 5th convolutional layer. To make AlexNet more connected to the language of this paper, we redefine its network output as a linear functions over all the five convolutional layers. We only train the weights of the convolutional layers and keep the weights of the linear layer unchanged.

We use fixed learning rate 0.01, momentum 0.9, batch size 128, and weight decay 0.0005. In the first 80 epochs, we freeze the (randomly initialized) weights of the 2nd through 5th convolutional layers, and only train the weights of the first layer). In the next 120 epochs, we unfreeze those weights and train all the 5 convolutional layers together.

As one can see from Figure 3, in the first 80 epochs, we have sufficiently trained the first layer (alone) so that the features do not move significantly anymore; however, as the 2nd through 5th layers become trained together, the features of the first layer gets significantly improved.

G.3 QUAD VS RELU VS NTK: FIGURE 11

Recall Figure 11 compares the performance of ReLU networks, quadratic networks and kernel methods. We use standard data augmentation plus Cutout augmentation in these experiments. Recall Cutout was also used in (Shankar et al., 2020) for presenting the best accuracy on neural kernel methods, so this comparison is fair.

ReLU network. For the network WRN-L-10, we widen each layer of a depth L ResNet by a factor of 10. We train 140 epochs with weight decay 0.0005. We use initial learning rate 0.1, and decay by a factor of 0.2 at epochs 80, 100 and 120. In the plots we present the best test accuracy out of 10 runs, as well as their ensemble accuracy.

Quadratic network. For the quadratic network WRN-L-10, we make slight modifications to the network to make it closer to our architecture used in the theorem, and make it more easily trainable. Specifically, we use activation function $\sigma(z) = z + 0.1z^2$ instead of $\sigma(z) = z^2$ to make the training more stable. We swap the order of Activation and BatchNorm to make BN come after quadratic activations; this re-scaling also stabilizes training. Finally, consistent with our theory, we add a linear layer connecting the output of each layer to the final soft-max gate; so the final output is a linear combination of all the intermediate layers. We train quadratic WRN-L-10 for also 140 epochs with weight decay 0.0005. We use initial learning rate 0.02, and decay by a factor of 0.3 at epochs 80, 100 and 120. We also present the best test accuracy out of 10 runs and their ensemble accuracy.

Finite-width NTK. We implemented a naive NTK version of the (ReLU) WRN-L-10 architecture on the CIFAR-10 dataset, and use iterative algorithms to train this (linear) NTK model. Per-epoch training is 10 times slower than standard WRN-L-10 because the 10-class outputs each requires a different set of trainable parameters. We find Adam with learning rate 0.001 is best suited for training such tasks, but the convergence speed is rather slow. We use batch size 50 and zero weight decay since the model does not overfit to the training set (thanks to data augmentation). We run the training for 200 epochs, with learning rate decay factor 0.2 at epochs 140 and 170. We run 10 single models using different random initializations (which correspond to 10 slightly different kernels), and report the best single-model accuracy; our ensemble accuracy is by combining the outputs of the 10 models.

In our finite-width NTK experiments, we also try with and without ZCA data preprocessing for comparison: ZCA data preprocessing was known to achieve accuracy gain in neural kernel methods (Shankar et al., 2020), but we observe in practice, it does not help in training *standard* ReLU or quadratic networks.

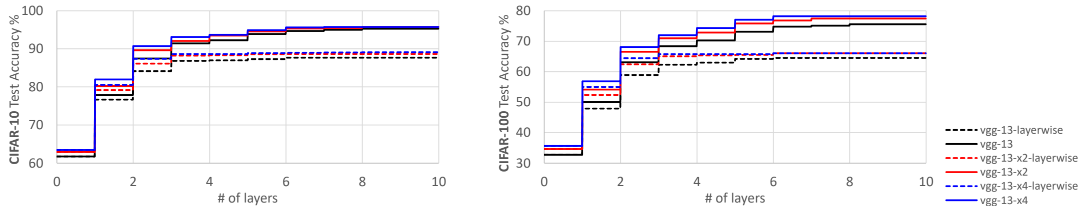
We only run this finite-width NTK for WRN-10-10. Using for instance WRN-16-10 to obtain the same test accuracy, one has to run for much more than 200 epochs; due to resource limitations, we refrain from trying bigger architectures on this finite-width NTK experiment.

G.4 LAYERWISE VS HIERACHICAL LEARNING: FIGURE 7

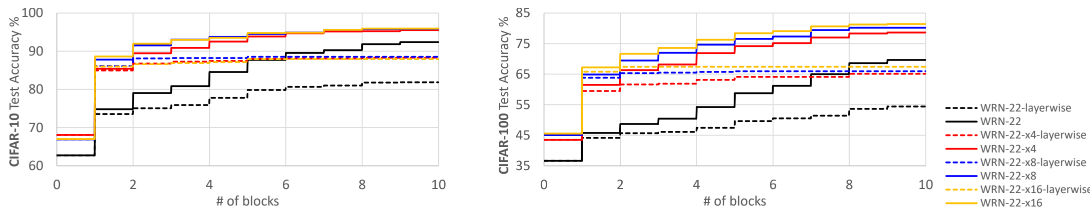
Recall Figure 7 compares the accuracy difference between layerwise training and training all the layers together on VGG19 and ResNet-34 architectures. We also include in Figure 7 additional experiments on VGG13 and ResNet-22.

In those experiments, we use standard data augmentation plus Cutout. When widening an architecture we widen all the layers together by the specific factor.

When performing "layerwise training", we adopt the same setup as Trinh (2019). During the ℓ-th phase, we freeze all the previous $(\ell - 1)$ convolutional layers to their already-trained weights (along with batch norm), add an additional linear layer (AvgPool + FC) connecting the output of the ℓ-th layer to the final soft-max gate, and only train the ℓ-th convolutional layer (with batch-norm) together with this additional linear layer. We train them for 120 epochs with initial learning rate 0.1 and decay it by 0.1 at epochs 80 and 100. We try both weight decay 0.0001 and 0.0005 and report the better accuracy for each phase ℓ (note this is needed for layer-wise training as smaller weight decay is suitable for smaller ℓ). Once we move to the next phase $\ell + 1$, we discard this additional



(a) VGG13+BatchNorm, accuracy at x-axis S indicates only the first S convolutional layers are trained



(b) WideResNet-22, accuracy at x-axis S indicates only the first S convolutional blocks are trained

Figure 12: **Layerwise training vs Training all layers together** (additional experiments to Figure 7).

linear layer.[17]

For "training all layers together", to make our comparison even stronger, we adopt nearly the same training setup as "layerwise training", except in the ℓ-th phase, we do not freeze the previous $\leq \ell - 1$ layers and train all the $\leq \ell$ layers altogether. In this way, we use the first $(\ell - 1)$ layers' pre-trained weights to continue training. The test accuracy obtained from this procedure is nearly identical to training the first ℓ layers altogether *directly from random initialization*.[18]

Finally, for ResNet experiments, we regard each Basic Block (consisting of 2 convolutional layers) as a single "layer" so in each phase (except for the first phase) of layerwise training, we train a single block together with the additional linear layer.

G.5 Measure Backward Feature Correlation: Figures 5, 8, 9 and 10

Recall in Figure 5 and Figure 10 we visualize how layer features change before and after backward feature correction (BFC); in Figure 8 and Figure 9 we present how much accuracy gain is related to BFC, and how much and how deep BFC goes on the CIFAR-100 dataset. In this section, we also provide additional experiments showing how much and how deep BFC goes on (1) the CIFAR-10 dataset in Figure 13(a), (2) on the ℓ_∞ adversarial training in Figure 13(b), and (3) on the ℓ_2 adversarial training in Figure 13(c).

In all of these experiments we use the vanilla WRN-34-5 architecture (Zagoruyko & Komodakis, 2016) (thus without widening the first layer and) without introducing "additional linear layer" like Section G.4. We use initial learning rate 0.1 and weight decay 0.0005. For clean training we train for 120 epochs and decay learning rate by 0.1 at epochs 80 and 100; for adversarial training we train for 100 epochs and decay learning rate by 0.1 at epochs 70 and 85. For the case of $\ell \in \{0, 1, 2, \dots, 10\}$:

- we first train only the first ℓ blocks of WRN-34-5 (and thus $2\ell + 1$ convolutional layers), by zeroing out all the remaining deeper layers. We call this "train only $\leq \ell$";
- we freeze these $2\ell+1$ layers and train only the deeper blocks (starting from random initialization) and call this "fix $\leq \ell$ train the rest";
- we also try to only freeze the $\leq \ell - j$ blocks for $j \in \{1, 2, 3, 4\}$ and train the remaining deeper blocks, and call this "fix $\leq \ell - j$ train the rest";

[17]Our "additional linear layer" is represented by a 2-dimensional average pooling unit followed by a (trainable) fully-connected unit. "Discarding" this additional linear before moving to the next phase is also used in (Trinh, 2019; Belilovsky et al., 2019).

[18]Our adopted process is known as "layer-wise pre-training" in some literature, and is also related to Algorithm 1 that we used in our theoretical analysis. We emphasize that "layer-wise pre-training" should be consider as training all the layers together and they have the same performance.

	CIFAR-10 accuracy	$\ell = 1$	$\ell = 3$	$\ell = 5$	$\ell = 7$	$\ell = 9$	$\ell = 11$	$\ell = 13$	$\ell = 15$	$\ell = 17$	$\ell = 19$	$\ell = 21$	
ensemble	train only $\leq \ell$	41.7%	73.5%	89.1%	92.2%	93.4%	93.9%	94.2%	95.1%	95.7%	95.9%	96.1%	→ no BFC
ensemble	fix $\leq \ell$, train the rest	96.5%	93.7%	91.7%	92.8%	93.7%	94.2%	94.3%	95.3%	95.9%	96.1%	96.3%	→ no BFC
ensemble	fix $\leq \ell - 2$, train the rest	-	96.5%	95.5%	93.4%	93.6%	94.1%	94.4%	95.2%	95.8%	95.9%	96.3%	→ BFC for 2 layers
ensemble	fix $\leq \ell - 4$, train the rest	-	-	96.4%	96.1%	94.9%	94.5%	94.4%	95.5%	95.6%	95.8%	96.1%	→ BFC for 4 layers
ensemble	fix $\leq \ell - 6$, train the rest	-	-	-	96.7%	96.1%	95.4%	95.0%	95.6%	96.0%	95.6%	95.9%	→ BFC for 6 layers
ensemble	fix $\leq \ell - 8$, train the rest	-	-	-	-	96.6%	96.3%	95.9%	96.2%	96.3%	96.1%	95.7%	→ BFC for 8 layers
ensemble	**train all the layers**	96.5%	96.6%	96.5%	96.5%	96.5%	96.4%	96.2%	96.3%	96.4%	96.5%	96.6%	→ full BFC
	average weight correlations (for "train $\leq \ell$" vs "rand init")	*0.276*	*0.083*	*0.068*	*0.060*	*0.053*	*0.043*	*0.043*	*0.040*	*0.036*	*0.037*	*0.031*	→ training neural nets is far from the NTK regime
	average weight correlations (for "train $\leq \ell$" vs "train all")	*0.905*	*0.977*	*0.975*	*0.972*	*0.966*	*0.965*	*0.964*	*0.966*	*0.959*	*0.957*	*0.939*	→ correlation between with vs. without BFC
single model	train only $\leq \ell$	41.4%	71.2%	86.5%	89.9%	91.3%	91.8%	92.3%	93.9%	94.5%	94.7%	95.0%	→ no BFC
single model	fix $\leq \ell$, train the rest	95.6%	90.8%	89.1%	90.4%	91.5%	92.0%	92.5%	93.9%	94.6%	94.8%	95.0%	→ no BFC
single model	fix $\leq \ell - 2$, train the rest	-	95.7%	93.7%	90.9%	91.3%	91.8%	92.3%	93.7%	94.4%	94.6%	94.9%	→ BFC for 2 layers
single model	fix $\leq \ell - 4$, train the rest	-	-	95.7%	94.7%	93.0%	92.2%	92.4%	93.9%	94.1%	94.2%	94.6%	→ BFC for 4 layers
single model	fix $\leq \ell - 6$, train the rest	-	-	-	95.8%	94.9%	93.8%	93.2%	94.2%	94.6%	94.1%	94.2%	→ BFC for 6 layers
single model	fix $\leq \ell - 8$, train the rest	-	-	-	-	95.8%	95.1%	94.5%	94.7%	95.0%	94.8%	94.1%	→ BFC for 8 layers
single model	**train all the layers**	95.8%	95.6%	95.6%	95.8%	95.6%	95.5%	95.4%	95.7%	95.9%	95.8%	95.9%	→ full BFC

(a) clean training on CIFAR-10

	CIFAR-10, L2 adversarial	$\ell = 1$	$\ell = 3$	$\ell = 5$	$\ell = 7$	$\ell = 9$	$\ell = 11$	$\ell = 13$	$\ell = 15$	$\ell = 17$	$\ell = 19$	$\ell = 21$	
single model	train only $\leq \ell$	23.5%	36.2%	47.0%	52.6%	55.9%	57.3%	58.7%	61.9%	63.3%	63.9%	64.4%	→ no BFC
single model	fix $\leq \ell$, train the rest	67.2%	63.4%	64.3%	62.5%	58.3%	57.8%	59.3%	61.8%	62.6%	63.7%	63.7%	→ no BFC
single model	fix $\leq \ell - 2$, train the rest	-	67.3%	65.7%	63.7%	62.3%	60.1%	59.9%	61.6%	62.7%	63.7%	64.1%	→ BFC for 2 layers
single model	fix $\leq \ell - 4$, train the rest	-	-	67.0%	65.9%	64.9%	63.1%	62.1%	63.5%	63.3%	63.5%	64.0%	→ BFC for 4 layers
single model	fix $\leq \ell - 6$, train the rest	-	-	-	67.0%	65.8%	64.7%	64.5%	65.3%	65.6%	64.7%	63.9%	→ BFC for 6 layers
single model	fix $\leq \ell - 8$, train the rest	-	-	-	-	67.0%	66.5%	65.8%	66.0%	66.3%	66.4%	65.1%	→ BFC for 8 layers
single model	**train all the layers**	66.7%	67.0%	66.9%	66.8%	66.7%	66.7%	66.7%	66.5%	67.0%	66.0%	65.5%	→ full BFC
	average weight correlations (for "train $\leq \ell$" vs "rand init")	*0.354*	*0.134*	*0.108*	*0.089*	*0.074*	*0.065*	*0.059*	*0.050*	*0.045*	*0.045*	*0.042*	→ training neural nets is far from the NTK regime
	average weight correlations (for "train $\leq \ell$" vs "train all")	*0.798*	*0.868*	*0.877*	*0.864*	*0.911*	*0.890*	*0.862*	*0.860*	*0.861*	*0.834*	*0.802*	→ correlation between with vs. without BFC

(b) adversarial training on CIFAR-10 with ℓ_∞ radius 6/255

	CIFAR-10, Linf adversarial	$\ell = 1$	$\ell = 3$	$\ell = 5$	$\ell = 7$	$\ell = 9$	$\ell = 11$	$\ell = 13$	$\ell = 15$	$\ell = 17$	$\ell = 19$	$\ell = 21$	
single model	train only $\leq \ell$	21.6%	28.4%	37.6%	42.8%	46.7%	48.1%	50.6%	54.1%	55.6%	56.5%	57.4%	→ no BFC
single model	fix $\leq \ell$, train the rest	60.6%	55.7%	56.7%	54.5%	52.6%	51.0%	52.0%	54.0%	55.5%	56.6%	57.5%	→ no BFC
single model	fix $\leq \ell - 2$, train the rest	-	61.3%	58.2%	55.9%	54.2%	53.3%	53.1%	55.6%	56.4%	56.5%	57.2%	→ BFC for 2 layers
single model	fix $\leq \ell - 4$, train the rest	-	-	61.0%	58.6%	57.0%	56.3%	55.7%	57.5%	57.8%	57.3%	57.6%	→ BFC for 4 layers
single model	fix $\leq \ell - 6$, train the rest	-	-	-	61.4%	59.9%	57.6%	58.0%	59.1%	59.5%	58.7%	58.3%	→ BFC for 6 layers
single model	fix $\leq \ell - 8$, train the rest	-	-	-	-	61.1%	59.3%	58.6%	59.4%	60.0%	60.4%	60.0%	→ BFC for 8 layers
single model	**train all the layers**	60.6%	60.5%	61.1%	61.3%	60.7%	60.1%	60.5%	60.5%	60.2%	60.4%	60.8%	→ full BFC
	average weight correlations (for "train $\leq \ell$" vs "rand init")	*0.164*	*0.137*	*0.086*	*0.070*	*0.061*	*0.054*	*0.054*	*0.041*	*0.047*	*0.040*	*0.037*	→ training neural nets is far from the NTK regime
	average weight correlations (for "train $\leq \ell$" vs "train all")	*0.622*	*0.783*	*0.845*	*0.835*	*0.898*	*0.883*	*0.847*	*0.867*	*0.849*	*0.942*	*0.904*	→ correlation between with vs. without BFC

(c) adversarial training on CIFAR-10 with $\ell_2(0.5, 0.12)$ attacker

Figure 13: This table gives more experiments comparing to Figure 8.

- we start from random initialization and train all the layers, but regularize the weights of the first $\leq \ell$ blocks so that they stay close to those obtained from "train only $\leq \ell$", and we call this "train all the layers".[19]

This explains how we obtained Figure 8, Figure 9 and Figure 13. We emphasize that by comparing the accuracy difference between "train all the layers" and "fix $\leq \ell - j$ and train the rest", one can immediately conclusion on how deep is it necessary for backward feature correction to go.

As for feature visualizations in Figure 5 and Figure 10, we compare the last layer visualizations of "train only $\leq \ell$" (or equivalently "fix $\leq \ell$ train the rest") which has *no* backward feature correction from deeper layers, as well as that of "train all the layers" which is after backward feature correction from all the deeper layers.

For the adversarial attacker used in Figure 13(b), we used ℓ_∞ PGD attacker for 7 steps during training, and for 20 steps during testing; for the adversarial attacker used in Figure 13(c), we used $\ell_2(0.5, 0.12)$ (see Section G.1) for training and replaces its PGD number of steps to 20 during testing.

[19]In principle, one can tune this regularizer weight so as to maximize neuron correlations to a magnitude without hurting the final accuracy. We did not do that, and simply trained using weights 0.0005 and 0.0007 and simply reported the better one without hurting the final accuracy.

G.6 Gap Assumption Verification: Figure 4

Recall in Figure 4 we have compared the accuracy performance of WRN-34-10 with various depths. In this experiment we have widened all the layers of the original ResNet-34 by a factor of 10, and we remove the deepest j basic blocks of the architecture for $j \in \{0, 1, 2, \dots, 15\}$ in order to represent WRN-34-10 with various depths.

We train each architecture for 120 epochs with weight decay 0.0005, and initial learning rate 0.1 with decay factor 0.1 at epochs 80 and 100. In the single model experiments, we run the training 10 times, and report the average accuracy of those 8 runs excluding the top and bottom ones; in the ensemble experiment, we use the average output of the 10 runs to perform classification.

APPENDIX II: COMPLETE PROOFS

We provide clear roadmap of what is included in this appendix. Note that a full statement of our theorem and its high-level proof plan begin on the next page.

- SECTION H : In this section, we first state the general version of the main theorem, including agnostic case in Section H.5.
- SECTION I : In this section, we introduce notations including defining the symmetric tensor product $*$ and the twice symmetrization operator $\mathbf{Sym}(\mathbf{M})$.
- SECTION J : In this section, we show useful properties of our loss function. To mention a few:
    1. In Section J.1 we show the truncated version $\widetilde{S}_\ell$ is close to S_ℓ in the population loss.
    2. In Section J.3 we show S_ℓ is Lipschitz continuous in the population loss. We need this to show that when doing a gradient update step, the quantity $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell\|^2]$ does not move too much in population loss. This is important for the self-regularization property we discussed in Section E to hold.
    3. In Section J.4 we show the empirical truncated loss is Lipschitz w.r.t. $\mathbf{K}$.
    4. In Section J.5 we show the empirical truncated loss satisfies higher-order Lipschitz smoothness w.r.t. $\mathbf{K}$ and $\mathbf{W}$. We need this to derive the time complexity of SGD.
    5. In Section J.6 we show empirical truncated loss is close to the population truncated loss. We need this together with Section J.1 to deriv the final generalization bound.
- SECTION K : In this section, we prove the critical result about the "coefficient preserving" property of $\widehat{S}^\star_\ell(x)$, as we discussed in Section E. This is used to show that if the output of F is close to $G^\star$ in population, then the high degree coefficient must match, thus $\mathbf{W}$ must be close to $\mathbf{W}^\star$ in some measure.
- SECTION L : In this section, we present our main technical lemma for hierarchical learning. It says as long as the (population) objective is as small as ε^2, then the following properties hold: loosely speaking, for every layer ℓ,
    1. (hierarchical learning): $S_\ell(x)$ close to $S^\star_\ell(x)$ by error $\sim \varepsilon/\alpha_\ell$, up to unitary transformation.
    2. (boundedness): each $\mathbb{E}[\|S_\ell(x)\|_2^2]$ is bounded. (This is needed in self-regularization.)

    We emphasize that these properties are maintained *gradually*. In the sense that we need to start with a case where these properties are already *approximately* satisfied, and then we show that the network will *self-regularize* to improve these properties. It does not mean, for example in the "hierarchical learning" property above, any network with loss smaller than ε^2 satisfies this property; we need to conclude from the fact that this network is obtained via a (small step) gradient update from an earlier network that has this property with loss $\leq 2\varepsilon$.
- SECTION M : In this section, we use the main technical lemma to show that there is a descent direction of the training objective, as long as the objective value is not too small. Specifically, we show that there is a gradient update direction of $\mathbf{K}$ and a second order Hessian update direction of $\mathbf{W}$, which guarantees to decrease the objective. This means, in the non-convex optimization language, there is no second-order critical points, so one can apply SGD to sufficiently decrease the objective.
- SECTION N : We show how to extend our theorems to classification.
- SECTION O : This section contains our lower bounds.

H MAIN THEOREM AND PROOF PLAN

Let us recall that d is the input dimension and $x \in \mathbb{R}^d$ is the input. We use L to denote the total number of layers in the network, and use k_ℓ to denote the width (number of neurons) of the hidden layer ℓ. Throughout the appendix, we make the following conventions:

- $k = \max_\ell\{k_\ell\}$ and $\overline{k}_\ell = \max\{k_j \,:\, j \in \mathcal{J}_\ell \wedge j \geq 2\}$.
- $B = \max_\ell\{B_\ell\}$ and $\overline{B}_\ell = \max\{B_j \,:\, j \in \mathcal{J}_\ell \wedge j \geq 2\}$.

Our main theorem in its full generalization can be stated as follows.

Theorem 1' (general case of Theorem 1)**.** *There is absolute constant $c_0 \geq 2$ so that for any desired accuracy $\varepsilon \in (0,1)$, suppose the following gap assumption is satisfied*

$$\frac{\alpha_\ell}{\alpha_{\ell+1}} \geq (c_4(2^\ell)\log(dL/\varepsilon))^{c_4(2^\ell)} \cdot (\kappa \cdot c_1(2^\ell) \cdot c_3(2^\ell))^{2^{c_0 \cdot L}} \prod_{j=\ell}^{L} \left(\overline{k}_\ell \overline{B}_\ell\right)^{L 2^{c_0(j-\ell)}}$$

Then, there exist choices of parameters (i.e., regularizer weight, learning rate, over parameterization) so that using

$$N \geq d^2 \cdot \log^{\Omega(1)} \frac{d}{\delta} + \frac{d \log d}{\varepsilon^6} \cdot \mathsf{poly}(B, k, \kappa) \cdot \left(c_4(2^L) \log \frac{BkL\kappa d}{\delta \varepsilon}\right)^{\Omega(c_4(2^L))}$$

samples. With probability at least 0.99 over the randomness of $\{\mathbf{R}_\ell\}_\ell$, with probability at least $1-\delta$ over the randomness of $\mathcal{Z}$, in at most time complexity

$$T \leq \mathsf{poly}\left(\kappa^L, \prod_\ell \overline{k}_\ell \overline{B}_\ell, (c_4(2^L))^{c_4(2^L)}, \log^{c_4(2^L)} \frac{1}{\delta}, \frac{d}{\varepsilon}\right)$$

SGD converges to a point with

$$\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2 \quad \widetilde{\mathbf{Obj}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2 \quad \mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2$$

Corollary H.1. *In the typical setting when $c_3(q) \leq q^{O(q)}$, $c_1(q) \leq O(q^q)$, and $c_4(q) \leq O(q)$, Theorem 1' simplifies to*

$$\frac{\alpha_\ell}{\alpha_{\ell+1}} \geq \left(\log \frac{d}{\varepsilon}\right)^{c_0 \cdot 2^\ell} (\kappa)^{2^{c_0 \cdot L}} \prod_{j=\ell}^{L} \left(\overline{k}_\ell \overline{B}_\ell\right)^{L 2^{c_0(j-\ell)}}$$

$$N \geq d^2 \cdot \log^{\Omega(1)} \frac{d}{\delta} + \frac{d \log d}{\varepsilon^6} \cdot \mathsf{poly}(B, k, \kappa) \cdot \left(2^L \log \frac{Bk\kappa d}{\delta \varepsilon}\right)^{\Omega(2^L)}$$

$$T \leq \mathsf{poly}\left(\kappa^L, \prod_\ell \overline{k}_\ell \overline{B}_\ell, 2^{L 2^L}, \log^{2^L} \frac{1}{\delta}, \frac{d}{\varepsilon}\right)$$

Corollary H.2. *In the special case Theorem 1, we have additional assumed $\delta = 0.01$, $L = o(\log \log d)$, $\kappa \leq 2^{C_1^L}, B_\ell \leq 2^{C_1^\ell} k_\ell$, and $k_\ell \leq d^{\frac{1}{C^{\ell+C_1}}}$. This together with the typical setting $c_3(q) \leq q^{O(q)}$, $c_1(q) \leq O(q^q)$, and $c_4(q) \leq O(q)$, simplifies Theorem 1' to*

$$\frac{\alpha_{\ell+1}}{\alpha_\ell} \leq d^{-\frac{1}{C^\ell}}, \quad N \geq \mathsf{poly}(d/\varepsilon), \quad \textit{and} \quad T \leq \mathsf{poly}(d/\varepsilon)$$

H.1 Truncated quadratic activation (for training)

To make our analysis simpler, it would be easier to work with an activation function that has bounded derivatives in the entire space. For each layer ℓ, we consider a "truncated, smooth" version of the square activation $\widetilde{\sigma}_\ell(z)$ defined as follows. For some sufficiently large B'_ℓ (to be chosen later), let

$$\widetilde{\sigma}_\ell(z) = \begin{cases} \sigma(z), & \text{if } |z| \leq B'_\ell \\ B''_\ell & \text{if } |z| \geq 2B'_\ell \end{cases} \quad \text{for some } B''_\ell = \Theta((B'_\ell)^2)$$

and in the range $[B'_\ell, 2B'_\ell]$, function $\widetilde{\sigma}(z)$ can be chosen as any monotone increasing function such that $|\widetilde{\sigma}_\ell(z)'|, |\widetilde{\sigma}_\ell(z)''|, |\widetilde{\sigma}_\ell(z)'''| = O(B'_\ell)$ are bounded for every z.

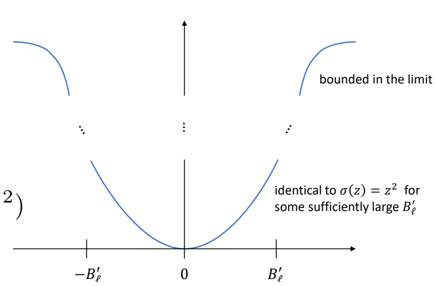


Figure 14: truncated quadratic activation

Accordingly, we define the learner network with respect to the truncated activation as follows.

$$\widetilde{S}_0(x) = G_0^\star(x) \quad , \quad \widetilde{S}_1(x) = G_1^\star(x), \quad \widetilde{S}_\ell(x) = \textstyle\sum_{j\in\mathcal{J}_\ell, j\geq 2} \mathbf{K}_{\ell,j}\widetilde{\sigma}_j\left(\mathbf{R}_j\widetilde{S}_j(x)\right) + \sum_{j\in\{0,1\}\cap\mathcal{J}_\ell} \mathbf{K}_{\ell,j}\widetilde{S}_j(x)$$

$$\widetilde{F}(x) = \textstyle\sum_{\ell=2}^L \alpha_\ell \mathbf{Sum}(\widetilde{F}_\ell(x)) \quad , \quad \widetilde{F}_\ell(x) = \sigma\left(\sum_{j\in\mathcal{J}_\ell, j\geq 2} \mathbf{W}_{\ell,j}\widetilde{\sigma}_j\left(\mathbf{R}_j\widetilde{S}_j(x)\right) + \sum_{j\in\{0,1\}\cap\mathcal{J}_\ell} \mathbf{W}_{\ell,j}\widetilde{S}_j(x)\right)$$

We also use $\widetilde{\sigma}$ instead of $\widetilde{\sigma}_j$ when its clear from content.

Remark H.3. The truncated $\widetilde{F}$ is for *training purpose* to ensure the network is Lipschitz smooth, so we can obtain simpler proofs. Our choice B'_ℓ makes sure when taking expectation over data, the difference between $\widetilde{\sigma}_\ell(z)$ and $\sigma(z)$ is negligible, see Appendix J.1. Thus, our *final learned network* $F(x)$ *is truly quadratic*. In practice, people use regularizers such as batch/layer normalization to make sure activations stay bounded, but truncation is much simpler to analyze in theory.

H.2 Parameter Choices

Definition H.4. *In our analysis, let us introduce a few more notations.*

- *With the following notation we can write* $\mathsf{poly}(\widetilde{\kappa}_\ell)$ *instead of* $\mathsf{poly}(\overline{k}_\ell, L, \kappa)$ *whenever needed.*
$$\widetilde{\kappa}_\ell = (\overline{k}_\ell \cdot L \cdot \kappa)^4 \text{ and } \tau_\ell = (\overline{B}_\ell \cdot \overline{k}_\ell \cdot L \cdot \kappa)^4.$$
- *The next one is our final choice of the truncation parameter for* $\widetilde{\sigma}_\ell(x)$ *at each layer* ℓ*.*
$$B'_\ell := \mathsf{poly}(\tau_\ell) \cdot \Omega(c_4(2^\ell)\log(dL/\varepsilon))^{c_4(2^\ell)} \text{ and } \overline{B}'_\ell = \max\{B'_j \,:\, j \in \mathcal{J}_\ell \wedge j \geq 2\}$$
- *The following can simplify our notations.*
$$k = \max_\ell\{k_\ell\},\ B = \max_\ell\{B_\ell\},\ \widetilde{\kappa} = \max_\ell\{\widetilde{\kappa}_\ell\},\ \tau = \max_\ell\{\tau_\ell\},\ B' = \max_\ell\{B'_\ell\}$$
- *The following is our main "big polynomial factors" to carry around, and it satisfies*
$$D_\ell := \left(\tau_\ell \cdot \kappa^{2^\ell} \cdot (2^\ell)^{2^\ell} \cdot c_1(2^\ell) \cdot c_3(2^\ell)\right)^{c_0 \ell} \text{ and } \Upsilon_\ell = \prod_{j=\ell}^{L} (D_j)^{20\cdot 2^{6(j-\ell)}}$$
Note it satisfies $\Upsilon_\ell \geq (D_\ell)^{20}(\Upsilon_{\ell+1}\Upsilon_{\ell+2}\cdots\Upsilon_L)^6$.
- *The following is our gap assumption.*
$$\frac{\alpha_{\ell+1}}{\alpha_\ell} \leq \frac{1}{(\Upsilon_{\ell+1})^6 \overline{B}'_{\ell+1}}$$
- *Our thresholds*
$$\mathsf{Thres}_{\ell,\triangle} = \left(\frac{\alpha_{\ell-1}}{(D_{\ell-1})^9\Upsilon_{\ell-1}}\right)^2 \quad , \quad \mathsf{Thres}_{\ell,\triangledown} = \frac{1}{4}\left(\frac{\alpha_\ell}{(D_\ell)^3\sqrt{\Upsilon_\ell}}\right)^2$$
- *The following is our choice of the regularizer weights*[20]
$$\lambda_{6,\ell} = \frac{\varepsilon^2}{(\widetilde{\kappa}_\ell)^2}, \quad \lambda_{3,\ell} = \frac{\alpha_\ell^2}{D_\ell \cdot \Upsilon_\ell}, \quad \lambda_{4,\ell} = \frac{\alpha_\ell^2}{(D_\ell)^7\Upsilon_\ell^2}, \quad \lambda_{5,\ell} = \frac{\alpha_\ell^2}{(D_\ell)^{13}\Upsilon_\ell^3}$$
- *The following is our amount of the over-parametrization*
$$m \geq \mathsf{poly}(\widetilde{\kappa}, B')/\varepsilon^2$$

[20]Let us make a comment on $\lambda_{6,\ell} = \frac{\varepsilon^2}{(\widetilde{\kappa}_\ell)^2}$. In Algorithm 1, we have in fact chosen $\lambda_{6,\ell} = \frac{(\varepsilon_0)^4}{(\widetilde{\kappa}_\ell)^2}$, where ε_0 is the current "target error", that is guaranteed to be within a factor of 2 comparing to the true ε (that comes from $\varepsilon^2 = \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$). To make the notations simpler, we have ignored this constant factor 2.

- *The following is our final choice of the sample complexity*

$$N \geq d^2 \cdot \log^{\Omega(1)} \frac{d}{\delta} + \frac{md \log d}{\varepsilon^4} \cdot \mathsf{poly}(\tau) \left(2^L c_4(2^L) \log \frac{\tau d}{\delta\varepsilon} \right)^{c_4(2^L)+\Omega(1)}$$

H.3 Algorithm Description For Analysis Purpose

For analysis purpose, it would be nice to divide our Algorithm 1 into stages for $\ell = 2, 3, \dots, L$.

- Stage $\ell^\triangle$ begins with $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \mathsf{Thres}_{\ell,\triangle} := \left(\frac{\alpha_{\ell-1}}{(D_{\ell-1})^9 \Upsilon_{\ell-1}}\right)^2$.

  Our algorithm satisfies $\eta_j = 0$ for $j > \ell$ and $\lambda_{3,j} = \lambda_{4,j} = \lambda_{5,j} = 0$ for $j \geq \ell$. In other words, only the matrices $\mathbf{W}_2, \dots, \mathbf{W}_\ell, \mathbf{K}_2, \dots, \mathbf{K}_{\ell-1}$ are training parameters and the rest of the matrices stay at zeros. Our analysis will ensure that applying (noisy) SGD one can decrease this objective to $\frac{1}{4}\left(\frac{\alpha_\ell}{(D_\ell)^3 \sqrt{\Upsilon_\ell}}\right)^2$, and when this point is reached we move to stage $\ell^\diamond$.

- $\ell^\diamond$ begins with $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \mathsf{Thres}_{\ell,\triangledown} := \frac{1}{4}\left(\frac{\alpha_\ell}{(D_\ell)^3 \sqrt{\Upsilon_\ell}}\right)^2$.

  In this stage, our analysis will guarantee that $\mathbf{W}_{\ell,\ell-1}^\top \mathbf{W}_{\ell\triangleleft}$ is extremely close to a rank k_ℓ matrix, so we can apply k-SVD decomposition to get some warm-up choice of $\mathbf{K}_\ell$ satisfying

  $$\|\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft} - \mathbf{W}_{\ell,\ell-1}^\top \mathbf{W}_{\ell\triangleleft}\|_F$$

  being sufficiently small. Then, we set $\lambda_{3,\ell}, \lambda_{4,\ell}, \lambda_{5,\ell}$ from Definition H.4, and our analysis will ensure that the objective increases to at most $\left(\frac{\alpha_\ell}{(D_\ell)^3 \sqrt{\Upsilon_\ell}}\right)^2$. We move to stage $\ell^\triangledown$.

- $\ell^\triangledown$ begins with $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq 4\mathsf{Thres}_{\ell,\triangledown} = \left(\frac{\alpha_\ell}{(D_\ell)^3 \sqrt{\Upsilon_\ell}}\right)^2$.

  Our algorithm satisfies $\eta_j = 0$ for $j > \ell$ and $\lambda_{3,j} = \lambda_{4,j} = \lambda_{5,j} = 0$ for $j > \ell$. In other words, only the matrices $\mathbf{W}_2, \dots, \mathbf{W}_\ell, \mathbf{K}_2, \dots, \mathbf{K}_\ell$ are training parameters and the rest of the matrices stay at zeros. Our analysis will ensure that applying (noisy) SGD one can decrease this objective to $\left(\frac{\alpha_\ell}{(D_\ell)^9 \Upsilon_\ell}\right)^2$, so we can move to stage $(\ell+1)^\triangle$.

H.4 Proof of Theorem 1'

We begin by noting that our truncated empirical objective $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$ is in fact lip-bounded, lip-Lipschitz continuous, lip-Lipschitz smooth, and lip-second-order smooth for some parameter $\mathsf{lip} = (\widetilde{\kappa}, B')^{O(L)} \cdot \mathsf{poly}\left(B, (c_4(2^L))^{c_4(2^L)}, \log^{c_4(2^L)} \frac{1}{\delta}, d\right)$ that is sufficiently small (see Claim J.5). This parameter lip will eventually go into our running time, but not anywhere else.

Throughout this proof, we assume as if $\lambda_{6,\ell}$ is always set to be $\frac{\varepsilon^2}{(\widetilde{\kappa}_\ell)^2}$, where $\varepsilon^2 = \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$ is the current objective value. (We can assume so because Algorithm 1 will iteratively shrink the target error ε_0 by a factor of 2.)

Stage $\ell^\triangle$. Suppose we begin this stage with the promise that (guaranteed by the previous stage)

$$\varepsilon^2 = \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \left(\frac{\alpha_{\ell-1}}{(D_{\ell-1})^9 \Upsilon_{\ell-1}}\right)^2 \quad \text{and} \quad \left\{ \mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2] \leq \tau_j \right\}_{j<\ell} \tag{H.1}$$

and Algorithm 1 will ensure that $\mathbf{W}_\ell = 0$ is now added to the trainable parameters.

Our main difficulty is to prove (see Theorem M.10) that whenever (H.1) holds, for every small $\eta_1 > 0$, there must exist some update direction $(\mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})})$ satisfying

- $\|\mathbf{K}^{(\mathsf{new})} - \mathbf{K}\|_F \leq \eta_1 \cdot \mathsf{poly}(\widetilde{\kappa})$,
- $\mathbb{E}_{\mathbf{D}} \|\mathbf{W}^{(\mathsf{new})} - \mathbf{W}\|_F^2 \leq \eta_1 \cdot \mathsf{poly}(\widetilde{\kappa})$,
- $\mathbb{E}_{\mathbf{D}}\left[\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})})\right] \leq \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) - \eta_1(0.7\varepsilon^2 - 2\alpha_{\ell+1}^2)$.

Therefore, as long as $\varepsilon^2 > 4\alpha_{\ell+1}^2$, by classical theory from optimization (see Fact P.11 for completeness), we know that

$$\text{either } \|\nabla\widetilde{\mathbf{Obj}}(\mathcal{Z};\mathbf{W},\mathbf{K})\|_F > \frac{\varepsilon^2}{\mathsf{poly}(\widetilde{\kappa})} \text{ or } \lambda_{\min}\left(\nabla^2\widetilde{\mathbf{Obj}}(\mathcal{Z};\mathbf{W},\mathbf{K})\right) \leq -\frac{\varepsilon^2}{\mathsf{poly}(\widetilde{\kappa})} \quad . \tag{H.2}$$

This means, the current point cannot be an (even approximate) second-order critical point. Invoking known results on stochastic non-conex optimization (Ge et al., 2015), we know starting from this point, (noisy) SGD can decrease the objective. Note the objective will continue to decrease at least until $\varepsilon^2 \leq 8\alpha_{\ell+1}^2$, but we do not need to wait until the objective is this small, and whenever ε hits $\frac{1}{2}\frac{\alpha_\ell}{(D_\ell)^3\sqrt{\Upsilon_\ell}}$, we can go into stage $\ell^\diamond$.

Remark H.5. In order to apply SGD to decrease the objective, we need to maintain that the boundedness $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2] \leq \tau_j$ in (H.1) always holds. This is ensured because of *self-regularization*: we proved that (1) whenever (H.1) holds it must satisfy a tighter bound $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2] \leq 2B_j \ll \tau_j$, and (2) the quantity $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2]$ satisfies a Lipschitz continuity statement (see Claim J.3). Specifically, if we move by η in step length, then $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2]$ is affected by at most $\eta\cdot\left(\prod_{j=\ell}^{i}\mathsf{poly}(\tau_j, c_3(2^j))\right)$. If we choose the step length of SGD to be smaller than this amount, then the quantity $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2]$ self-regularizes. (This Lipschitz continuity factor also goes into the running time.)

Stage $\ell^\diamond$. Using $\varepsilon^2 \leq \frac{1}{4}\left(\frac{\alpha_\ell}{(D_\ell)^3\sqrt{\Upsilon_\ell}}\right)^2$, we shall have a theorem to derive that[21]

$$\left\|\mathbf{W}_{\ell,\ell-1}^\top\mathbf{W}_{\ell\lhd} - \mathbf{M}\right\|_F^2 \leq \frac{\mathsf{poly}(\widetilde{\kappa}_\ell)}{(D_\ell)^4\Upsilon_\ell}$$

for some matrix $\mathbf{M}$ with rank k_ℓ and singular values between $[\frac{1}{\kappa^2}, \kappa^2L^2]$. Note that when connecting this back to Line 21 of Algorithm 1, we immediately know that the computed k_ℓ is correct. Therefore, applying k_ℓ-SVD decomposition on $\mathbf{W}_{\ell,\ell-1}^\top\mathbf{W}_{\ell\lhd}$ on Line 23, one can derive a warm-up solution of $\mathbf{K}_\ell$ satisfying

$$\|\mathbf{K}_{\ell,\ell-1}^\top\mathbf{K}_{\ell\lhd} - \mathbf{W}_{\ell,\ell-1}^\top\mathbf{W}_{\ell\lhd}\|_F^2 \leq \frac{\mathsf{poly}(\widetilde{\kappa}_\ell)}{(D_\ell)^4\Upsilon_\ell} \quad .$$

Note that, without loss of generality, we can assume $\|\mathbf{K}_\ell\|_F \leq \mathsf{poly}(\kappa, L) \leq \widetilde{\kappa}_\ell/100$ and

$$\|\mathbf{K}_{\ell,\ell-1}^\top\mathbf{K}_{\ell,\ell-1} - \mathbf{W}_{\ell,\ell-1}^\top\mathbf{W}_{\ell,\ell-1}\|_F^2 \leq \mathsf{poly}(\widetilde{\kappa}_\ell) \quad\text{and}\quad \|\mathbf{K}_\ell^\top\mathbf{K}_\ell - \mathbf{W}_\ell^\top\mathbf{W}_\ell\|_F^2 \leq \mathsf{poly}(\widetilde{\kappa}_\ell)$$

(This can be done by left/right multiplying the SVD solution as the solution is not unique.

Since we have chosen regularizer weights (see Definition H.4)

$$\lambda_{6,\ell} = \frac{\varepsilon^2}{(\widetilde{\kappa}_\ell)^2}, \quad \lambda_{3,\ell} = \frac{\alpha_\ell^2}{D_\ell\cdot\Upsilon_\ell}, \quad \lambda_{4,\ell} = \frac{\alpha_\ell^2}{(D_\ell)^7\Upsilon_\ell^2}, \quad \lambda_{5,\ell} = \frac{\alpha_\ell^2}{(D_\ell)^{13}\Upsilon_\ell^3}$$

with the introduction of new trainable variables $\mathbf{K}_\ell$, our objective has increased by at most

$$\begin{aligned}&\lambda_{6,\ell}\frac{(\widetilde{\kappa}_\ell)^2}{100} + \lambda_{3,\ell}\cdot\frac{\mathsf{poly}(\widetilde{\kappa}_\ell)}{(D_\ell)^4\Upsilon_\ell} + \lambda_{4,\ell}\cdot\mathsf{poly}(\widetilde{\kappa}_\ell) + \lambda_{5,\ell}\cdot\mathsf{poly}(\widetilde{\kappa}_\ell)\\ \leq\ &\frac{\varepsilon^2}{100} + \frac{\alpha_\ell^2}{\Upsilon_\ell^2(D_\ell)^4} + +\frac{\alpha_\ell^2}{\Upsilon_\ell^2(D_\ell)^6} + +\frac{\alpha_\ell^2}{\Upsilon_\ell^3(D_\ell)^{12}} \leq \frac{1}{4}\left(\frac{\alpha_\ell}{(D_\ell)^3\sqrt{\Upsilon_\ell}}\right)^2\end{aligned}$$

This means we can move to stage $\ell^\triangledown$.

[21] In the language of later sections, Corollary L.4a implies

$$\left\|\mathbf{Q}_{\ell-1}^\top\overline{\mathbf{W}}_{\ell,\ell-1}^\top\overline{\mathbf{W}}_{\ell\lhd}\mathbf{Q}_{\ell\lhd} - \overline{\mathbf{W}^\star}_{\ell,\ell-1}^\top\overline{\mathbf{W}^\star}_{\ell\lhd}\right\|_F^2 \leq \frac{1}{(D_\ell)^4\Upsilon_\ell} \quad .$$

Since $\overline{\mathbf{W}^\star}_{\ell,\ell-1}^\top\overline{\mathbf{W}^\star}_{\ell\lhd}$ is of rank k_ℓ, this means $\mathbf{Q}_{\ell-1}^\top\overline{\mathbf{W}}_{\ell,\ell-1}^\top\overline{\mathbf{W}}_{\ell\lhd}\mathbf{Q}_{\ell\lhd}$ is close to rank k_ℓ. Since our notation $\overline{\mathbf{W}}_{\ell,j}\mathbf{Q}_j$ is only an abbreviation of $\mathbf{W}_{\ell,j}(\mathbf{R}_j\mathbf{U}_j * \mathbf{R}_j\mathbf{U}_j)$ for some well conditioned matrix $(\mathbf{R}_j\mathbf{U}_j * \mathbf{R}_j\mathbf{U}_j)$, this also implies $\mathbf{W}_{\ell,\ell-1}^\top\mathbf{W}_{\ell\lhd}$ is close to being rank k_ℓ. At the same time, we know that the singular values of $\overline{\mathbf{W}^\star}_{\ell,\ell-1}^\top\overline{\mathbf{W}^\star}_{\ell\lhd}$ are between $[\frac{1}{\kappa^2}, \kappa^2L^2]$ (see Fact I.7).

Stage $\boldsymbol{\ell}^{\triangledown}$. We begin this stage with the promise

$$\varepsilon^2 = \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \left(\frac{\alpha_\ell}{(D_\ell)^3 \sqrt{\Upsilon_\ell}}\right)^2 \quad \text{and} \quad \left\{ \mathbb{E}_{x \sim \mathcal{D}}[\|S_j(x)\|_2^2] \leq \tau_j \right\}_{j<\ell} \tag{H.3}$$

and our trainable parameters are $\mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{K}_1, \ldots, \mathbf{K}_\ell$. This time, we have another Theorem M.11 to guarantee that as long as (H.3) is satisfied, then (H.2) still holds (namely, it is not an approximate second-order critical point). Therefore, one can still apply standard (noisy) SGD to sufficiently decrease the objective at least until $\varepsilon^2 \leq 8\alpha_{\ell+1}^2$ (or until arbitrarily small $\varepsilon^2 > 0$ if $\ell = L$). This is much smaller than the requirement of stage $(\ell+1)^{\vartriangle}$.

For similar reason as Remark H.5, we have self-regularization so $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2] \leq \tau_j$ (for $j < \ell$) holds throughout the optimization process. In addition, this time Theorem M.11 also implies that whenever we exit this stage, namely when $\varepsilon \leq \frac{\alpha_\ell}{(D_\ell)^9 \Upsilon_\ell}$ is satisfied, then $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|_2^2] \leq 2B_\ell$.

End of Algorithm. Note in the last $L^{\triangledown}$ stage, we can decrease the objective until arbitrarily small $\varepsilon^2 > 0$ and thus we have $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2$. Applying Proposition J.7 (relating empirical and population losses) and Claim J.1 (relating truncated and quadratic losses), we have

$$\widetilde{\mathbf{Obj}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 2\varepsilon^2 \quad \text{and} \quad \mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 3\varepsilon^2 \ .$$

Time Complexity. As for the time complexity, since our objective satisfies lip-Lipschitz property until second-order smoothness, the time complexity of SGD depends only on $\mathsf{poly}(\mathsf{lip}, \frac{1}{\varepsilon}, d)$ (see (Ge et al., 2015)).

Quadratic Activation. We used the truncated quadratic activation $\widetilde{\sigma}_j(x)$ only for the purpose to make sure the training objective is sufficiently smooth. Our analysis will ensure that, in fact, when substituting $\widetilde{\sigma}_j(x)$ back with the vanilla quadratic activation, the objective is also small (see (M.8) and (M.9)).

H.5 Our Theorem on Agnostic Learning

For notational simplicity, throughout this paper we have assumed that the exact true label $G^\star(x)$ is given for every training input $x \sim \mathcal{Z}$. This is called *realizable learning*.

In fact, our proof trivially generalizes to the *agnostic learning* case at the expense of introducing extra notations. Suppose that $Y(x) \in \mathbb{R}$ is a label function (not necessarily a polynomial) and is OPT close to some target network, or in symbols,

$$\mathbb{E}_{x\sim\mathcal{D}}\left[(G^\star(x) - Y(x))^2\right] \leq \mathsf{OPT} \ .$$

Suppose the algorithm is given training set $\{(x, Y(x)) \colon x \in \mathcal{Z}\}$, so the loss function now becomes

$$\mathbf{Loss}(x; \mathbf{W}, \mathbf{K}) = (F(x; \mathbf{W}, \mathbf{K}) - Y(x))^2$$

Suppose in addition that $|Y(x)| \leq B$ almost surely. Then,[22]

Theorem 3' (agonistic version of Theorem 1')**.** *For every constant $\gamma > 1$, for any desired accuracy $\varepsilon \in (\sqrt{\mathsf{OPT}}, 1)$, in the same setting as Theorem 1', Algorithm 1 can find a point with*

$$\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq (1+\frac{1}{\gamma})\mathsf{OPT}+\varepsilon^2 \quad \widetilde{\mathbf{Obj}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq (1+\frac{1}{\gamma})\mathsf{OPT}+\varepsilon^2 \quad \mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq (1+\frac{1}{\gamma})\mathsf{OPT}+\varepsilon^2$$

I Notations and Preliminaries

We denote by $\|w\|_2$ and $\|w\|_\infty$ the Euclidean and infinity norms of vectors w, and $\|w\|_0$ the number of non-zeros of w. We also abbreviate $\|w\| = \|w\|_2$ when it is clear from the context. We use $\|\mathbf{W}\|_F, \|\mathbf{W}\|_2$ to denote the Frobenius and spectral norm of matrix $\mathbf{W}$. We use $\mathbf{A} \succeq \mathbf{B}$ to denote that the difference between two symmetric matrices $\mathbf{A} - \mathbf{B}$ is positive semi-definite. We use

[22] The proof is nearly identical. The main difference is to replace the use of $\mathsf{OPT}_{\leq \ell} \leq 2\alpha_{\ell+1}^2$ with $\mathsf{OPT}_{\leq \ell} \leq O(\alpha_{\ell+1}^2) + (1+\frac{1}{\gamma})\mathsf{OPT}$ (when invoking Lemma M.8) in the final proofs of Theorem M.10 and Theorem M.11.

$\sigma_{\min}(\mathbf{A}), \sigma_{\max}(\mathbf{A})$ to denote the minimum and maximum singular values of a rectangular matrix, and $\lambda_{\min}(\mathbf{A}), \lambda_{\max}(\mathbf{A})$ for the minimum and maximum eigenvalues.

We use $\mathcal{N}(\mu, \sigma)$ to denote Gaussian distribution with mean μ and variance σ; or $\mathcal{N}(\mu, \Sigma)$ to denote Gaussian vector with mean μ and covariance Σ. We use $\mathbb{1}_{event}$ or $\mathbb{1}[event]$ to denote the indicator function of whether $event$ is true.

We denote $\mathbf{Sum}(x) = \sum_i x_i$ as the sum of the coordinate of this vector. We use $\sigma(x) = x^2$ as the quadratic activation function. Also recall

Definition I.1. *Given any degree-q homogenous polynomial $f(x) = \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I \prod_{j \in [n]} x_j^{I_j}$, define*

$$\mathcal{C}_x(f) := \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I^2$$

When it is clear from the context, we also denote $\mathcal{C}(f) = \mathcal{C}_x(f)$.

I.1 Symmetric Tensor

When it is clear from the context, in this paper sets can be multisets. This allows us to write $\{i, i\}$. We also support notation $\forall \{i, j\} \in \binom{n+1}{2}$ to denote all possible (unordered) sub multi-sets of $[n]$ with cardinality 2.

Definition I.2 (symmetric tensor)**.** *The symmetric tensor $*$ for two vectors $x, y \in \mathbb{R}^n$ is given as:*

$$[x * y]_{\{i,j\}} = a_{i,j} x_i x_j, \quad \forall 1 \le i \le j \le p$$

for $a_{i,i} = 1$ and $a_{i,j} = \sqrt{2}$ for $j \neq i$. Note $x * y \in \mathbb{R}^{\binom{n+1}{2}}$. The symmetric tensor $*$ for two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}_{m \times n}$ is given as:

$$[\mathbf{X} * \mathbf{Y}]_{p,\{i,j\}} = a_{i,j} \mathbf{X}_{p,i} \mathbf{X}_{p,j}, \quad \forall p \in [m], 1 \le i \le j \le p$$

and it satisfies $\mathbf{X} * \mathbf{Y} \in \mathbb{R}^{m \times \binom{n+1}{2}}$.

It is a simple exercise to verify that $\langle x, y \rangle^2 = \langle x * x, y * y \rangle$.

Definition I.3 (**Sym**)**.** *For any $\mathbf{M} \in \mathbb{R}^{\binom{n+1}{2} \times \binom{n+1}{2}}$, define $\mathbf{Sym}(\mathbf{M}) \in \mathbb{R}^{\binom{n+1}{2} \times \binom{n+1}{2}}$ to be the "twice-symmetric" version of $\mathbf{M}$. For every $1 \le i \le j \le n$ and $1 \le k \le l \le n$, define*[23]

$$\mathbf{Sym}(\mathbf{M})_{\{i,j\},\{k,l\}} := \frac{\sum_{\{p,q\},\{r,s\} \in \binom{n+1}{2} \wedge \{p,q,r,s\} = \{i,j,k,l\}} a_{p,q} a_{r,s} \mathbf{M}_{\{p,q\},\{r,s\}}}{a_{i,j} a_{k,l} \cdot \left| \left\{ \{p,q\}, \{r,s\} \in \binom{n+1}{2} \colon \{p,q,r,s\} = \{i,j,k,l\} \right\} \right|}$$

Fact I.4. $\mathbf{Sym}(\mathbf{M})$ *satisfies the following three properties.*

- $(z * z)^\top \mathbf{Sym}(\mathbf{M}) (z * z) = (z * z)^\top \mathbf{M} (z * z)$ *for every* $z \in \mathbb{R}^n$*;*
- *If $\mathbf{M}$ is symmetric and satisfies $\mathbf{M}_{\{i,j\},\{k,l\}} = 0$ whenever $i \neq j$ or $k \neq l$, then $\mathbf{Sym}(\mathbf{M}) = \mathbf{M}$.*
- $O(1) \|\mathbf{M}\|_F^2 \ge \mathcal{C}_z\big((z * z)^\top \mathbf{M} (z * z)\big) \ge \|\mathbf{Sym}(\mathbf{M})\|_F^2$

It is not hard to derive the following important property (proof see Appendix P.3)

Lemma I.5. *If $\mathbf{U} \in \mathbb{R}^{p \times p}$ is unitary and $\mathbf{R} \in \mathbb{R}^{s \times p}$ for $s \ge \binom{p+1}{2}$, then there exists some* unitary *matrix $\mathbf{Q} \in \mathbb{R}^{\binom{p+1}{2} \times \binom{p+1}{2}}$ so that $\mathbf{RU} * \mathbf{RU} = (\mathbf{R} * \mathbf{R}) \mathbf{Q}$.*

I.2 Network Initialization and Network Tensor Notions

We show the following lemma on random initialization (proved in Appendix P.2).

[23]For instance, when $i, j, k, l \in [n]$ are distinct, this means

$$\mathbf{Sym}(\mathbf{M})_{\{i,j\},\{k,l\}} = \frac{\mathbf{M}_{\{i,j\},\{k,l\}} + \mathbf{M}_{\{i,k\},\{j,l\}} + \mathbf{M}_{\{i,l\},\{j,k\}} + \mathbf{M}_{\{j,k\},\{i,l\}} + \mathbf{M}_{\{j,l\},\{i,k\}} + \mathbf{M}_{\{k,l\},\{i,j\}}}{6}.$$

Lemma I.6. *Let $\mathbf{R}_\ell \in \mathbb{R}^{\binom{k_\ell+1}{2} \times k_\ell}$ be a random matrix such that each entry is i.i.d. from $\mathcal{N}\left(0, \frac{1}{k_\ell^2}\right)$, then with probability at least $1-p$, $\mathbf{R}_\ell * \mathbf{R}_\ell$ has singular values between $[\frac{1}{O(k_\ell^4 p^2)}, O(1+\frac{1}{k_\ell^2}\log\frac{k_\ell}{p})]$, and $\|\mathbf{R}_\ell\|_2 \le O(1+\frac{\sqrt{\log(1/p)}}{k_\ell})$.*

As a result, with probability at least 0.99, it satisfies for all $\ell = 2,3,\dots,L$, the square matrices $\mathbf{R}_\ell * \mathbf{R}_\ell$ have singular values between $[\frac{1}{O(k_\ell^4 L^2)}, O(1+\frac{\log(Lk_\ell)}{k_\ell})]$ and $\|\mathbf{R}_\ell\|_2 \le O(1+\frac{\sqrt{\log L}}{k_\ell})$.

Through out the analysis, it is more convenient to work on the matrix symmetric tensors. For every $\ell = 2,3,4,\dots,L$ and every $j \in \mathcal{J}_\ell \setminus \{0,1\}$, we define

$$
\begin{aligned}
\overline{\mathbf{W}^\star}_{\ell,j} &:= \mathbf{W}^\star_{\ell,j}(\mathbf{I} * \mathbf{I}) = \mathbf{W}^\star_{\ell,j} * \mathbf{W}^\star_{\ell,j} && \in \mathbb{R}^{k_\ell \times \binom{k_j+1}{2}} \\
\overline{\mathbf{W}}_{\ell,j} &:= \mathbf{W}_{\ell,j}(\mathbf{R}_j * \mathbf{R}_j) = \mathbf{W}_{\ell,j}\mathbf{R}_j * \mathbf{W}_{\ell,j}\mathbf{R}_j && \in \mathbb{R}^{m \times \binom{k_j+1}{2}} \\
\overline{\mathbf{K}}_{\ell,j} &:= \mathbf{K}_{\ell,j}(\mathbf{R}_j * \mathbf{R}_j) = \mathbf{K}_{\ell,j}\mathbf{R}_j * \mathbf{K}_{\ell,j}\mathbf{R}_j && \in \mathbb{R}^{k_\ell \times \binom{k_j+1}{2}}
\end{aligned}
$$

so that

$$
\begin{aligned}
\forall z \in \mathbb{R}^{k_j}: \quad \overline{\mathbf{W}^\star}_{\ell,j}(z * z) &= \mathbf{W}^\star_{\ell,j}\sigma(z) \\
\overline{\mathbf{W}}_{\ell,j}(z * z) &= \mathbf{W}_{\ell,j}\sigma(\mathbf{R}_j z) \\
\overline{\mathbf{K}}_{\ell,j}(z * z) &= \mathbf{K}_{\ell,j}\sigma(\mathbf{R}_j z)
\end{aligned}
$$

For convenience, whenever $j \in \mathcal{J}_\ell \cap \{0,1\}$, we also write

$$
\overline{\mathbf{W}^\star}_{\ell,j} = \mathbf{W}^\star_{\ell,j} \quad \overline{\mathbf{W}}_{\ell,j} = \mathbf{W}_{\ell,j} \quad \overline{\mathbf{K}}_{\ell,j} = \mathbf{K}_{\ell,j}
$$

We define

$$
\begin{aligned}
&\overline{\mathbf{W}^\star}_\ell = \left(\overline{\mathbf{W}^\star}_{\ell,j}\right)_{j\in\mathcal{J}_\ell} \in \mathbb{R}^{k_\ell \times *}, \quad \overline{\mathbf{W}}_\ell = \left(\overline{\mathbf{W}}_{\ell,j}\right)_{j\in\mathcal{J}_\ell} \in \mathbb{R}^{m \times *}, \quad \overline{\mathbf{K}}_\ell = \left(\overline{\mathbf{K}}_{\ell,j}\right)_{j\in\mathcal{J}_\ell} \in \mathbb{R}^{k_\ell \times *} \\
&\overline{\mathbf{W}^\star}_{\ell\triangleleft} = \left(\overline{\mathbf{W}^\star}_{\ell,j}\right)_{j\in\mathcal{J}_\ell, j\neq \ell-1}, \quad \overline{\mathbf{W}}_{\ell\triangleleft} = \left(\overline{\mathbf{W}}_{\ell,j}\right)_{j\in\mathcal{J}_\ell, j\neq \ell-1}, \quad \overline{\mathbf{K}}_{\ell\triangleleft} = \left(\overline{\mathbf{K}}_{\ell,j}\right)_{j\in\mathcal{J}_\ell, j\neq \ell-1}
\end{aligned}
$$

Fact I.7. *Singular values of $\mathbf{W}^\star_{\ell,j}$ are in $[1/\kappa, \kappa]$. Singular values of $\overline{\mathbf{W}^\star}_\ell$ and $\overline{\mathbf{W}^\star}_{\ell\triangleleft}$ are in $[1/\kappa, \ell\kappa]$.*

J USEFUL PROPERTIES OF OUR OBJECTIVE FUNCTION

J.1 CLOSENESS: POPULATION QUADRATIC VS. POPULATION TRUNCATED LOSS

Claim J.1. *Suppose for every $\ell \in [L]$, $\|\mathbf{K}_\ell\|_2, \|\mathbf{W}_\ell\|_2 \le \widetilde{\kappa}_\ell$ for some $\widetilde{\kappa}_\ell \ge k_\ell + L + \kappa$ and $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2] \le \tau_\ell$ for some $\tau_\ell \ge \widetilde{\kappa}_\ell$. Then, for every $\varepsilon \in (0,1]$, when choosing*

$$
\textit{truncation parameter:} \qquad B'_\ell \ge \tau_\ell^2 \cdot \mathsf{poly}(\widetilde{\kappa}_\ell) \cdot \Omega(2^\ell c_4(2^\ell)\log(dL/\varepsilon))^{c_4(2^\ell)} \ ,
$$

we have for every integer constant $p \le 10$,

$$
\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\widetilde{F}(x) - F(x)\right)^p\right] \le \varepsilon \quad \textit{and} \quad \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\|\widetilde{S}_\ell(x) - S_\ell(x)\|_2\right)^p\right] \le \varepsilon
$$

Proof of Claim J.1. We first focus on $\widetilde{S}_\ell(x) - S_\ell(x)$. We first note that for every $S_\ell(x), \widetilde{S}_\ell(x)$, there is a crude (but absolute) upper bound:

$$
\|S_\ell(x)\|_2, \|\widetilde{S}_\ell(x)\|_2 \le (\widetilde{\kappa}_\ell k_\ell \ell)^{O(2^\ell)} \|x\|_2^{2^\ell} =: C_1 \|x\|_2^{2^\ell} \ .
$$

By the isotropic property of x (see (D.1)) and the hyper-contractivity (see (D.2)), we know that for R_1 is as large as $R_1 = (d\log(C_1/\varepsilon))^{\Omega(2^\ell)}$, it holds that

$$
\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\mathbb{1}_{\|x\|_2^{2^\ell} \ge R_1} \|x\|_2^{p\cdot 2^\ell}\right] \le \frac{\varepsilon}{2C_1^p}
$$

This implies

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\|\widetilde{S}_\ell(x)-S_\ell(x)\|_2\right)^p \mathbb{1}_{\|x\|_2^{2^\ell}\geq R_1}\right]\leq\frac{\varepsilon}{2} \tag{J.1}$$

Next, we consider the remaining part, since $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2]\leq\tau_\ell$, we know that when $B'_\ell\geq\tau_\ell\cdot\Omega(c_4(2^\ell))^{c_4(2^\ell)}\log^{c_4(2^\ell)}(C_1R_1L/\varepsilon)$, by the hyper-contractivity Property D.2, we have for every fixed ℓ,

$$\mathbf{Pr}[\|\mathbf{R}_\ell S_\ell(x)\|_2\geq B'_\ell]\leq\frac{\varepsilon}{2(2C_1R_1)^pL}$$

Therefore, with probability at least $1-\frac{\varepsilon}{2(2C_1R_1)^p}$, at every layer ℓ, the value plugged into $\widetilde{\sigma}$ and σ are the same. As a result,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\|\widetilde{S}_\ell(x)-S_\ell(x)\|_2\right)^p \mathbb{1}_{\|x\|_2^{2^\ell}\leq R_1}\right]\leq(2C_1R_1)^p\,\mathbf{Pr}\left[\exists\ell'\leq\ell,\|\mathbf{R}_{\ell'}S_{\ell'}(x)\|_2\geq B'_\ell\right]\leq\varepsilon/2 \tag{J.2}$$

Putting together (J.1) and (J.2) we complete the proof that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\|\widetilde{S}_\ell(x)-S_\ell(x)\|_2\right)^p\right]\leq\varepsilon$$

An identical proof also shows that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\|\mathbf{Sum}(\widetilde{F}_\ell(x))-\mathbf{Sum}(F_\ell)(x)\|_2\right)^p\right]\leq\varepsilon$$

Thus, scaling down by a factor of Lp we can derive the bound on $\mathbb{E}_{x\sim\mathcal{D}}\left[\left(\widetilde{F}(x)-F(x)\right)^p\right]$. $\square$

J.2 Covariance: Empirical vs. Population

Recall that our isotropic Property D.1 says for every $w\in\mathbb{R}^d$,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\langle w,x\rangle^2]\leq O(1)\cdot\|w\|^2\quad\text{and}\quad\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\langle w,S_1(x)\rangle^2]\leq O(1)\cdot\|w\|^2\ .$$

Below we show that this also holds for the empirical dataset as long as enough samples are given.

Proposition J.2. *As long as $N=d^2\cdot\log^{\Omega(1)}\frac{d}{\delta}$, with probability at least $1-\delta$ over the random choice of $\mathcal{Z}$, for every vector $w\in\mathbb{R}^d$,*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}[\langle w,x\rangle^4]\leq O(1)\cdot\|w\|^2\quad\textit{and}\quad\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}[\langle w,S_1(x)\rangle^4]\leq O(1)\cdot\|w\|^2$$
$$\forall x\in\mathcal{Z}\colon\ \max\{\|x\|^2,\|S_1(x)\|^2\}\leq d\log^{O(1)}\frac{d}{\delta}$$

Proof of Proposition J.2. Our isotropic Property D.1 together with the hyper-contractivity Property D.2 implies if $N\geq d\log^{\Omega(1)}\frac{d}{\delta}$, then with probability at least $1-\delta/4$,

$$\forall x\in\mathcal{Z}:\quad\|x\|^2\leq R_3\quad\text{and}\quad\|S_1(x)\|^2\leq R_3$$

Where $R_3=d\cdot\log^{O(1)}\frac{d}{\delta}$. Next, conditioning on this event, we can apply Bernstein's inequality to derive that as long as $N\geq\Omega(R_3\cdot\log\frac{1}{\delta_0})$ with probability at least $1-\delta_0$, for every fixed $w\in\mathbb{R}^d$,

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[\langle w,x\rangle^4\geq\Omega(1)\right]\geq1-\delta_0$$

Taking an epsilon-net over all possible w finishes the proof. $\square$

J.3 Lipschitz Continuity: Population Quadratic

Claim J.3. *Suppose $\mathbf{K}$ satisfies $\|\mathbf{K}_j\|_2\leq\tau_j$ for every $j\in\{2,3,\cdots,L\}$ where $\tau_j\geq\overline{k}_j+\kappa+L$, and suppose for some $\ell\in\{2,3,\cdots,L\}$, $\mathbf{K}_\ell$ replaced with $\mathbf{K}'_\ell=\mathbf{K}_\ell+\mathbf{\Delta}_\ell$ with any $\|\mathbf{\Delta}_\ell\|_F\leq$*

$\left(\prod_{j=\ell}^{L} \mathsf{poly}(\tau_j, c_3(2^j))\right)^{-1}$*, then for every* $i \geq \ell$

$$\mathbb{E}_{x\sim\mathcal{D}}\left[\left|\|S_i'(x)\|^2 - \|S_i(x)\|^2\right|\right] \leq \eta \cdot \left(\prod_{j=\ell}^{i} \mathsf{poly}(\tau_j, c_3(2^j))\right)$$

and for every $i < \ell$ *obviously* $S_i(x) = S_i'(x)$*.*

Proof of Claim J.3. We first check the stability with respect to $\mathbf{K}$, and suppose without loss of generality that only one $\mathbf{W}_\ell$ is changed for some ℓ. For notation simplicity, suppose we do an update $\mathbf{K}_\ell' = \mathbf{K}_\ell + \eta\mathbf{\Delta}_\ell$ for $\|\mathbf{\Delta}_\ell\|_F = 1$. We use S' to denote the sequence of S after the update, and we have $S_j'(x) = S_j(x)$ for every $j < \ell$. As for $S_\ell'(x)$, we have

$$\begin{aligned}
\|S_\ell'(x) - S_\ell(x)\| &\leq \eta\left(\sum_{j\geq 2}^{\ell-1} \|\mathbf{\Delta}_{\ell,j}\|_2 \|\sigma(\mathbf{R}_j S_j(x))\| + \|\mathbf{\Delta}_{\ell,1} S_1(x)\| + \|\mathbf{\Delta}_{\ell,0} x\|\right) \\
&\leq \eta \mathsf{poly}(\overline{k}_\ell, \kappa, L)\left(\sum_{j<\ell} \|S_j(x)\|^2 + \|\mathbf{\Delta}_{\ell,1} S_1(x)\| + \|\mathbf{\Delta}_{\ell,0} x\|\right)
\end{aligned}$$

so using $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|^2] \leq \tau_j$, the isotropic Property D.1 and the hyper-contractivity Property D.3, we can write

$$\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell'(x) - S_\ell(x)\|^2] \leq \eta^2 \mathsf{poly}(\tau_\ell, c_3(2^\ell)) =: \theta_\ell$$

As for later layers $i > \ell$, we have

$$\|S_i'(x) - S_i(x)\| \leq 4\sum_{j\geq 2}^{i-1} \|\mathbf{K}_{i,j}\|_2 \|\mathbf{R}_j\|_2^2 (\|S_j(x)\| \|S_j'(x) - S_j(x)\| + \|S_j'(x) - S_j(x)\|^2)$$

so taking square and expectation, and using hyper-contractivity Property D.3 again, (and using our assumption on η)[24]

$$\mathbb{E}_{x\sim\mathcal{D}} \|S_i'(x) - S_i(x)\|^2 \leq \mathsf{poly}(\tau_i, c_3(2^i)) \cdot \theta_{i-1} =: \theta_i$$

by recursing $\theta_i = \mathsf{poly}(\tau_i, c_3(2^i)) \cdot \theta_{i-1}$ we have

$$\mathbb{E}_{x\sim\mathcal{D}} \|S_i'(x) - S_i(x)\|^2 \leq \left(\prod_{j=\ell}^{i} \mathsf{poly}(\tau_j, c_3(2^j))\right)$$

□

J.4 Lipschitz Continuity: Empirical Truncated Loss in $\mathbf{K}$

Claim J.4. *Suppose the sampled set $\mathcal{Z}$ satisfies the event of Proposition J.2. For every $\mathbf{W}, \mathbf{K}$ satisfying*

$$\forall j = 2, 3, \ldots, L: \quad \|\mathbf{W}_j\|_2 \leq \widetilde{\kappa}_j,\ \|\mathbf{K}_j\|_2 \leq \widetilde{\kappa}_j$$

for some $\widetilde{\kappa}_j \geq k_j + \kappa + L$. Then, for any $\ell \in \{2, 3, \cdots, L-1\}$ and consider $\mathbf{K}_\ell$ replaced with $\mathbf{K}_\ell' = \mathbf{K}_\ell + \mathbf{\Delta}_\ell$ for any $\|\mathbf{\Delta}_\ell\|_F \leq \frac{1}{\mathsf{poly}(\widetilde{\kappa}_\ell, \overline{B}_\ell', d)}$. Then,

$$|\widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) - \widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}')| \leq \alpha_{\ell+1}\sqrt{\widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})} \cdot \mathsf{poly}(\widetilde{\kappa}_j, \overline{B}_j') \cdot \|\mathbf{\Delta}_\ell\|_F$$

Proof of Claim J.4. Let us denote $\varepsilon^2 = \widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$. For notation simplicity, suppose we do an update $\mathbf{K}_\ell' = \mathbf{K}_\ell + \eta\mathbf{\Delta}_\ell$ for $\eta > 0$ and $\|\mathbf{\Delta}_\ell\|_F = 1$. We use $\widetilde{S}'$ to denote the sequence of $\widetilde{S}$ after the

[24] This requires one to repeatedly apply the trivial inequality $ab \leq \eta a^2 + b^2/\eta$.

update, and we have $\widetilde{S}'_j(x) = \widetilde{S}_j(x)$ for every $j < \ell$. As for $\widetilde{S}'_\ell(x)$, we have (using the boundedness of $\widetilde{\sigma}$)

$$\|\widetilde{S}'_\ell(x) - \widetilde{S}_\ell(x)\| \leq \eta \left(\sum_{j\geq 2}^{\ell-1} \|\mathbf{\Delta}_{\ell,j}\|_2 \|\widetilde{\sigma}(\widetilde{S}_j(x))\| + \|\mathbf{\Delta}_{\ell,1} S_1(x)\| + \|\mathbf{\Delta}_{\ell,0} x\| \right)$$
$$\leq \eta L \overline{B}'_\ell + \eta \left(\|\mathbf{\Delta}_{\ell,1} S_1(x)\| + \|\mathbf{\Delta}_{\ell,0} x\| \right)$$

As for later layers $i > \ell$, we have (using the Lipschitz continuity of $\widetilde{\sigma}$)

$$\|\widetilde{S}'_i(x) - \widetilde{S}_i(x)\| \leq \sum_{j\geq 2}^{i-1} \|\mathbf{K}_{i,j}\|_2 B'_j \|\mathbf{R}_j\|_2 \|\widetilde{S}'_j(x) - \widetilde{S}_j(x)\|$$
$$\leq \cdots \leq \prod_{j=\ell+1}^{i} (\widetilde{\kappa}_j \overline{B}'_j L^2) \left(\eta L \overline{B}'_\ell + \eta \left(\|\mathbf{\Delta}_{\ell,1} S_1(x)\| + \|\mathbf{\Delta}_{\ell,0} x\| \right) \right) =: p_i$$

As for $\widetilde{F}(x)$, recall

$$\widetilde{F}(x) = \sum_i \alpha_i \left\| \mathbf{W}_{i,0} x + \mathbf{W}_{i,1} S_1(x) + \sum_{j\in\{2,3,\cdots,i-1\}} \mathbf{W}_{i,j} \sigma\left(\mathbf{R}_j \widetilde{S}_j(x) \right) \right\|^2 =: \sum_i \alpha_i \|A_i\|^2 \ .$$

Using the bound $\|A_i\| \leq \|\mathbf{W}_{i,0} x\| + \|\mathbf{W}_{i,1} S_1(x)\| + \mathsf{poly}(\widetilde{\kappa}_i, \overline{B}'_i)$, one can carefully verify[25]

$$|\widetilde{F}'(x) - \widetilde{F}(x)| \leq \sum_{i\geq \ell+1} \alpha_i \left(\|A_i\| \cdot p_{i-1} + p_{i-1}^2 \right) \cdot \mathsf{poly}(\widetilde{\kappa}_i, \overline{B}'_i)$$
$$\leq \alpha_{\ell+1} \eta \mathsf{poly}(\widetilde{\kappa}_\ell, \overline{B}'_\ell) \cdot \left(1 + (\|\mathbf{W}_{\ell,0} x\| + \|\mathbf{W}_{\ell,1} S_1(x)\|)(\|\mathbf{\Delta}_{\ell,1} S_1(x)\| + \|\mathbf{\Delta}_{\ell,0} x\|)\right)$$

Therefore, we know that

$$\left| \left(G^\star(x) - \widetilde{F}(x) \right)^2 - \left(G^\star(x) - \widetilde{F}'(x) \right)^2 \right|$$
$$\leq 2 \left| G^\star(x) - \widetilde{F}(x) \right| \cdot |\widetilde{F}'(x) - \widetilde{F}(x)| + |\widetilde{F}'(x) - \widetilde{F}(x)|^2$$
$$\leq \frac{\alpha_{\ell+1}\eta}{\varepsilon} \cdot \left| G^\star(x) - \widetilde{F}(x) \right|^2 + \varepsilon \frac{|\widetilde{F}'(x) - \widetilde{F}(x)|^2}{\alpha_{\ell+1}\eta} + |\widetilde{F}'(x) - \widetilde{F}(x)|^2$$
$$\leq \frac{\alpha_{\ell+1}\eta}{\varepsilon} \cdot \left| G^\star(x) - \widetilde{F}(x) \right|^2$$
$$+ \varepsilon \alpha_{\ell+1} \eta \mathsf{poly}(\widetilde{\kappa}_\ell, \overline{B}'_\ell) \left(1 + (\|\mathbf{W}_{\ell,0} x\|^2 + \|\mathbf{W}_{\ell,1} S_1(x)\|^2)(\|\mathbf{\Delta}_{\ell,1} S_1(x)\|^2 + \|\mathbf{\Delta}_{\ell,0} x\|)^2 \right)$$

Note that $2a^2b^2 \leq a^4 + b^4$ and:

- From Proposition J.2 we have $\mathbb{E}_{x\sim\mathcal{Z}} \|\mathbf{W}_{\ell,0} x\|^4, \mathbb{E}_{x\sim\mathcal{Z}} \|\mathbf{W}_{\ell,1} S_1(x)\|^4 \leq \widetilde{\kappa}_\ell$.
- From Proposition J.2 we have $\mathbb{E}_{x\sim\mathcal{Z}} \|\mathbf{\Delta}_{\ell,1} S_1(x)\|^4 + \|\mathbf{\Delta}_{\ell,0} x\|^4 \leq \mathsf{poly}(\widetilde{\kappa}_\ell)$.
- From definition of ε we have $\mathbb{E}_{x\sim\mathcal{Z}} \left| G^\star(x) - \widetilde{F}(x) \right|^2 = \varepsilon^2$.

Therefore, taking expectation we have

$$\mathbb{E}_{x\sim\mathcal{Z}} \left| \left(G^\star(x) - \widetilde{F}(x) \right)^2 - \left(G^\star(x) - \widetilde{F}(x)' \right)^2 \right| \leq \varepsilon \alpha_{\ell+1} \eta \mathsf{poly}(\widetilde{\kappa}_\ell, \overline{B}'_\ell) \ . \qquad \square$$

J.5 Lipschitz Smoothness: Empirical Truncated Loss (Crude Bound)

Recall a function $f(x)$ over domain $\mathcal{X}$ is

[25] This requires us to use the gap assumption between α_{i+1} and α_i, and the sufficient small choice of $\eta > 0$. For instance, the $\eta^2 \|\mathbf{\Delta}_{\ell,0} x\|^2$ term diminishes because η is sufficiently small and $\|x\|$ is bounded for every $x \sim \mathcal{Z}$ (see Proposition J.2).

- lip-Lipschitz continuous if $f(y) \le f(x) + \mathsf{lip} \cdot \|y-x\|_F$ for all $x, y \in \mathcal{X}$;
- lip-Lipschitz smooth if $f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{\mathsf{lip}}{2} \cdot \|y-x\|_F^2$ for all $x, y \in \mathcal{X}$;
- lip-Lipschitz second-order smooth if $f(y) \le f(x) + \langle \nabla f(x), y-x\rangle + \frac{1}{2}(y-x)^\top \nabla f(x)(y-x) + \frac{\mathsf{lip}}{6} \cdot \|y-x\|_F^3$ for all $x, y \in \mathcal{X}$.

We have the following crude bound:

Claim J.5. *Consider the domain consisting of all* $\mathbf{W}, \mathbf{K}$ *with*

$$\forall j = 2, 3, \dots, L\colon \quad \|\mathbf{W}_j\|_2 \le \widetilde{\kappa}_j,\ \|\mathbf{K}_j\|_2 \le \widetilde{\kappa}_j$$

for some $\widetilde{\kappa}_j \ge \overline{k}_j + L + \kappa$*, we have for every* $x \sim \mathcal{D}$*,*

- $|\widetilde{F}(x; \mathbf{W}, \mathbf{K})| \le \mathsf{poly}(\widetilde{\kappa}, B') \cdot \sum_\ell (\|\mathbf{W}_{\ell,0} x\|^2 + \|\mathbf{W}_{\ell,1} S_1(x)\|^2)$.
- $\widetilde{F}(x; \mathbf{W}, \mathbf{K})$ *is* lip*-Lipschitz continuous,* lip*-Lipschitz smooth, and* lip*-Lipschitz second-order smooth in* $\mathbf{W}, \mathbf{K}$ *for* $\mathsf{lip} = \prod_\ell (\widetilde{\kappa}_\ell, \overline{B}'_\ell)^{O(1)} \cdot \mathsf{poly}(G^\star(x), \|x\|)$

Suppose the sampled set $\mathcal{Z}$ *satisfies the event of Proposition J.2, then*

- $\widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$ *is* lip*-Lipschitz continuous,* lip*-Lipschitz smooth, and* lip*-Lipschitz second-order smooth in* $\mathbf{W}, \mathbf{K}$ *for* $\mathsf{lip} = \prod_\ell (\widetilde{\kappa}_\ell, \overline{B}'_\ell)^{O(1)} \cdot \mathsf{poly}\Big(B, (c_4(2^L))^{c_4(2^L)}, \log^{c_4(2^L)} \frac{1}{\delta}, d\Big)$.

We first state the following bound on chain of derivatives

Claim J.6 (chain derivatives)**.** *For every integer* $K > 0$*, every functions* $f, g_1, g_2, \dots, g_K : \mathbb{R} \to \mathbb{R}$*, and every integer* $p_0 > 0$*, suppose there exists a value* $R_0, R_1 > 1$ *and an integer* $s \ge 0$ *such that*

$$\forall p \in \{0, 1, \cdots, p_0\}, i \in [K]\colon \left|\frac{d^p f(x)}{dx^p}\right| \le R_0^p, \quad \left|\frac{d^p g_i(x)}{dx^p}\right| \le R_1^p \ .$$

Then, the function $h(x, w) = f(\sum_{i \in [K]} w_i g_i(x))$ *satisfies:*

$$\forall p \in \{0, 1, \cdots, p_0\}\colon \left|\frac{\partial^p h(x,w)}{\partial x^p}\right| \le (p R_0 \|w_i\|_1 R_1)^p$$

$$\forall p \in \{0, 1, \cdots, p_0\}, i \in [K]\colon \left|\frac{\partial^p h(x,w)}{\partial w_i^p}\right| \le |R_0 g_i(x)|^p$$

Proof of Claim J.6. We first consider $\left|\frac{\partial^p h(x,w)}{\partial x^p}\right|$. Using Faà di Bruno's formula, we have that

$$\frac{\partial^p h(x,w)}{\partial x^p} = \sum_{1 \cdot p_1 + 2 \cdot p_2 + \cdots + p \cdot p_p = p} \frac{p!}{p_1! p_2! \cdots p_p!} f^{(p_1 + \cdots p_p)} \left(\sum_{i \in [K]} w_i g_i(x)\right) \prod_{j=1}^p \left(\frac{\sum_{i \in [K]} w_i g_i^{(j)}(x)}{j!}\right)^{p_j}$$

Note that from our assumption

- $\prod_{j=1}^p \left|\left(\frac{\sum_{i \in [K]} w_i g_i^{(j)}(x)}{j!}\right)^{p_j}\right| \le \prod_{j=1}^p (\|w\|_1 R_1)^{j p_j} = (\|w\|_1 R_1)^p$.
- $|f^{(p_1 + \cdots p_p)} \Big(\sum_{i \in [K]} w_i g_i(x)\Big)| \le R_0^p$

Combining them, we have

$$\left|\frac{\partial^p h(x,w)}{\partial x^p}\right| \le (p R_0 \|w_i\|_1 R_1)^p$$

On the other hand, consider each w_i, we also have:

$$\left|\frac{\partial^p h(x,w)}{\partial w_i^p}\right| = \left|f^{(p)} \left(\sum_{i \in [K]} w_i g_i(x)\right) (g_i(x))^p\right| \le |R_0 g_i(x)|^p \qquad \square$$

Proof of Claim J.5. The first 4 inequalities is a direct corollary of Claim J.6.

Initially, we have a multivariate function but it suffices to check its directional first, second and third-order gradient. (For any function $g(y)\colon \mathbb{R}^m \to \mathbb{R}^n$, we can take $g(y+\alpha\delta)$ and consider $\frac{d^p g_j(y+\alpha\delta)}{d\alpha^p}$ for every coordinate j and every unit vector w.)

- In the base case, we have multivariate functions $f(\mathbf{K}_{\ell,0}) = \mathbf{K}_{\ell,0}x$ or $f(\mathbf{K}_{\ell,1}) = \mathbf{K}_{\ell,1}S_1(x)$. For each direction $\|\mathbf{\Delta}\|_F = 1$ we have $\left|\frac{d}{d\alpha^p} f(\mathbf{K}_{\ell,0} + \alpha\mathbf{\Delta}_{\ell,0})\right| \leq \|x\|^p$ so we can take $R_1 = \|x\|$ (and for $f(\mathbf{K}_{\ell,1})$ we can take $R_1 = \|x\|^2$.)
- Whenever we compose with $\widetilde{\sigma}$ at layer ℓ, for instance calculating $h(w,y) = \widetilde{\sigma}(\sum_i w_i f_i(y))$ (when viewing all matrices as vectors), we only need to calculate $\frac{\partial^p}{\partial\alpha^p} h_j(w, y+\alpha\delta) = \frac{\partial^p}{\partial\alpha^p}\widetilde{\sigma}(\sum_i w_{j,i} f_i(y+\alpha\delta))$, so we can apply Claim J.6 and R_1 becomes $O(\overline{B}'_\ell \widetilde{\kappa}_\ell \overline{k}_\ell L)\cdot R_1$. We can do the same for the w variables, so overall for any unit (δ_x,δ_w) it satisfies $|\frac{\partial^p}{\partial\alpha^p} h_j(w + \alpha\delta_w, y+\alpha\delta_y)| \leq \big(O(\overline{B}'_\ell \widetilde{\kappa}_\ell (\overline{k}_\ell L)^2)\cdot R_1\big)^p$.
- We also need to compose with the vanilla σ function three times:
  - once of the form $\sigma(f(\mathbf{K}_2,\dots,\mathbf{K}_{\ell-1}))$ for calculating $\widetilde{F}_\ell(x)$,
  - once of the form $\sigma(\mathbf{W}_\ell f(\mathbf{K}_2,\dots,\mathbf{K}_{\ell-1}))$ for calculating $\widetilde{F}_\ell(x)$, and
  - once of the form $(f(\mathbf{W},\mathbf{K}) - G^\star(x))^2$ for the final squared loss.

  In those calculations, although $g(x) = x^2$ does not have a bounded gradient (indeed, $\frac{d}{dx}g(x) = x$ can go to infinity when x is infinite), we know that the input x is always *bounded* by $\mathsf{poly}(\widetilde{\kappa}, \|x\|, B', G^\star(x))$. Therefore, we can also invoke Claim J.6.

Finally, we obtain the desired bounds on the first, second, and third order Lipschitzness property of $\widetilde{\mathbf{Loss}}(x;\mathbf{W},\mathbf{K})$.

For the bounds on $\widetilde{\mathbf{Loss}}(\mathcal{Z};\mathbf{W},\mathbf{K})$, we can use the absolute bounds on $\mathbf{Sum}(G^\star(x))$ and $\|x\|$ for all $x\in\mathcal{Z}$ (see Proposition J.2). $\square$

J.6 CLOSENESS: EMPIRICAL TRUNCATED VS. POPULATION TRUNCATED LOSS

Proposition J.7 (population $\leq$ empirical + ε_{s})**.** *Let P be the total number of parameters in $\{\mathbf{W}_\ell,\mathbf{K}_\ell\}_{\ell\in[L]}$. Then for every $\varepsilon_{\mathsf{s}},\delta\geq 0$ and $\widetilde{\kappa} \geq k+L+\kappa$, as long as*

$$N = \Omega\left(\frac{P\log(d/\delta)}{\varepsilon_{\mathsf{s}}^2}\cdot \mathsf{poly}(\widetilde{\kappa}, B')\left(c_4(2^L)\log\frac{\widetilde{\kappa}B'}{\varepsilon_{\mathsf{s}}}\right)^{c_4(2^L)+O(1)}\right) ,$$

with probability at least $1-\delta$ over the choice of $\mathcal{Z}$, we have that for every $\{\mathbf{W}_\ell,\mathbf{K}_\ell\}_{\ell\in[L]}$ satisfying $\|\mathbf{W}_\ell\|_F, \|\mathbf{K}_\ell\|_F \leq \widetilde{\kappa}$, it holds:

$$\widetilde{\mathbf{Loss}}(\mathcal{D};\mathbf{W},\mathbf{K}) \leq \widetilde{\mathbf{Loss}}(\mathcal{Z};\mathbf{W},\mathbf{K}) + \varepsilon_{\mathsf{s}}$$

Proof of Proposition J.7. Observe that for every fixed $R_0 > 0$ and $R_1 > B' > 0$ (to be chosen later),

$$\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}\left[\left(G^\star(x) - \widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|\leq R_0, \|x\|\leq R_1}\right] \leq \mathop{\mathbb{E}}_{x\sim\mathcal{Z}}\left[\left(G^\star(x) - \widetilde{F}(x)\right)^2\right]$$

Moreover, each function $R(x) = \left(G^\star(x) - \widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|\leq R_0, \|x\|\leq R_1}$ satisfies that

- boundedness: $|R(x)| \leq R_0^2$, and
- Lipschitz continuity: $R(x)$ is a $\mathsf{lip} \leq \mathsf{poly}(\widetilde{\kappa}, B', R_0, R_1, d)$-Lipschitz continuous in $(\mathbf{W},\mathbf{K})$ (by applying Claim J.5 and the fact $G^\star(x) \leq R_0 + \widetilde{F}(x) \leq \mathsf{poly}(\widetilde{\kappa}, B', R_0, R_1, d)$)

Therefore, we can take an epsilon-net on $(\mathbf{W},\mathbf{K})$ to conclude that as long as $N = \Omega\left(\frac{R_0^4 P\log(\widetilde{\kappa}B' R_1 d/(\delta\varepsilon_{\mathsf{s}}))}{\varepsilon_{\mathsf{s}}^2}\right)$, we have that w.p. at least $1-\delta$, for every $(\mathbf{W},\mathbf{K})$ within our bound (e.g. every $\|\mathbf{W}_\ell\|_2, \|\mathbf{K}_\ell\|_2 \leq \widetilde{\kappa}$), it holds:

$$\begin{aligned}&\mathbb{E}_{x\sim\mathcal{D}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|\leq R_0, \|x\|\leq R_1}\right]\\ &\leq \mathbb{E}_{x\sim\mathcal{Z}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|\leq R_0, \|x\|\leq R_1}\right] + \varepsilon_{\mathsf{s}}/2 \leq \mathbb{E}_{x\sim\mathcal{Z}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2\right] + \varepsilon_{\mathsf{s}}/2\end{aligned}$$

As for the remaining terms, let us write

$$\begin{aligned}&\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|>R_0 \text{ or } \|x\|>R_1}\\ &\leq \left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|>R_0} + R_0^2 \cdot \mathbb{1}_{\|x\|>R_1}\\ &\leq 4\left(G^\star(x)\right)^2 \mathbb{1}_{|G^\star(x)|>R_0/2} + 4(\widetilde{F}(x))^2 \mathbb{1}_{|\widetilde{F}(x)|>R_0/2} + R_0^2 \cdot \mathbb{1}_{\|x\|>R_1}\end{aligned}$$

- For the first term, recalling $\mathbb{E}_{x\sim\mathcal{D}}[G^\star(x) \leq B]$ so we can apply the hyper-contractivity Property D.2 to show that, as long as $R_0 \geq \mathsf{poly}(\widetilde{\kappa}) \cdot \left(c_4(2^L)\log\frac{\widetilde{\kappa}}{\varepsilon_{\mathsf{s}}}\right)^{c_4(2^L)}$ then it satisfies $\mathbb{E}_{x\sim\mathcal{D}}[4\left(G^\star(x)\right)^2 \mathbb{1}_{|G^\star(x)|>R_0/2}] \leq \varepsilon_{\mathsf{s}}/10$.

- For the second term, recall from Claim J.5 that $|\widetilde{F}(x)| \leq \mathsf{poly}(\widetilde{\kappa}, B') \cdot \sum_\ell(\|\mathbf{W}_{\ell,0}x\|^2 + \|\mathbf{W}_{\ell,1}S_1(x)\|^2)$; therefore, we can write

  $$\begin{aligned}&4(\widetilde{F}(x))^2\mathbb{1}_{|\widetilde{F}(x)|>R_0/2}\\ &\leq \mathsf{poly}(\widetilde{\kappa}, B')\sum_\ell\left(\|\mathbf{W}_{\ell,0}x\|^2\mathbb{1}_{\|\mathbf{W}_{\ell,0}x\|^2>\frac{R_0}{\mathsf{poly}(\widetilde{\kappa},B')}} + \|\mathbf{W}_{\ell,1}S_1(x)\|^2\mathbb{1}_{\|\mathbf{W}_{\ell,1}S_1(x)\|^2>\frac{R_0}{\mathsf{poly}(\widetilde{\kappa},B')}}\right) .\end{aligned}$$

  Applying the isotropic Property D.1 and the hyper-contractivity (D.2) on $\|\mathbf{W}_{\ell,0}x\|^2$ and $\|\mathbf{W}_{\ell,1}S_1(x)\|^2$, we have as long as $R_0 \geq \mathsf{poly}(\widetilde{\kappa}, B') \cdot \left(\log\frac{\widetilde{\kappa}B'}{\varepsilon_{\mathsf{s}}}\right)^{\Omega(1)}$, then it satisfies

  $$\mathbb{E}_{x\sim\mathcal{D}}[4(\widetilde{F}(x))^2\mathbb{1}_{|\widetilde{F}(x)|>R_0/2}] \leq \varepsilon_{\mathsf{s}}/10 \qquad \text{(for every } \mathbf{W}, \mathbf{K} \text{ in the range)}$$

- For the third term, as long as $R_1 = d\log^{\Omega(1)}(R_0/\varepsilon_{\mathsf{s}})$ then we have $\mathbb{E}_{x\sim\mathcal{D}}[R_0^2 \cdot \mathbb{1}_{\|x\|>R_1}] \leq \varepsilon_{\mathsf{s}}/10$.

Putting them together, we can choose $R_0 = \mathsf{poly}(\widetilde{\kappa}, B')\left(c_4(2^L)\log\frac{\widetilde{\kappa}B'}{\varepsilon_{\mathsf{s}}}\right)^{O(1)+c_4(2^L)}$ and we have

$$\mathbb{E}_{x\sim\mathcal{D}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|>R_0 \text{ or } \|x\|>R_1}\right] \leq \varepsilon_{\mathsf{s}}/2 .$$

This completes the proof that

$$\mathbb{E}_{x\sim\mathcal{D}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2\right] \leq \mathbb{E}_{x\sim\mathcal{Z}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2\right] + \varepsilon_{\mathsf{s}} . \qquad \square$$

Proposition J.8 (empirical $\leq$ population + ε_{s})**.** *Let P be the total number of parameters in $\{\mathbf{W}_\ell, \mathbf{K}_\ell\}_{\ell\in[L]}$. Then for every $\varepsilon_{\mathsf{s}}, \delta \geq 0$ and $\widetilde{\kappa} \geq k + L + \kappa$, as long as*

$$N = \Omega\left(\frac{P\log d}{\varepsilon_{\mathsf{s}}^2} \cdot \mathsf{poly}(\widetilde{\kappa}, B')\left(c_4(2^L)\log\frac{\widetilde{\kappa}B'}{\delta\varepsilon_{\mathsf{s}}}\right)^{c_4(2^L)+O(1)}\right) ,$$

for any **fixed** $\{\mathbf{W}_{\ell,0}, \mathbf{W}_{\ell,1}\}_{\ell\in[L]}$*, with probability at least $1-\delta$ over the choice of $\mathcal{Z}$, we have that for* every $\{\mathbf{W}_\ell, \mathbf{K}_\ell\}_{\ell\in[L]}$ *satisfying (1) $\|\mathbf{W}_\ell\|_F, \|\mathbf{K}_\ell\|_F \leq \widetilde{\kappa}$ and (2) consistent with $\{\mathbf{W}_{\ell,0}, \mathbf{W}_{\ell,1}\}_{\ell\in[L]}$, it holds:*

$$\mathbb{E}_{x\sim\mathcal{Z}}[\widetilde{\mathbf{Loss}}(x;\mathbf{W},\mathbf{K})] = \mathbb{E}_{x\sim\mathcal{Z}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2\right] \leq \mathbb{E}_{x\sim\mathcal{D}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2\right] + \varepsilon_{\mathsf{s}} = \mathbb{E}_{x\sim\mathcal{D}}[\widetilde{\mathbf{Loss}}(x;\mathbf{W},\mathbf{K})] + \varepsilon_{\mathsf{s}}$$

Proof. We first reverse the argument of Proposition J.7 and have that as long as $N = \Omega\left(\frac{R_0^4 P\log(\widetilde{\kappa}B'R_1 d/(\delta\varepsilon_{\mathsf{s}}))}{\varepsilon_{\mathsf{s}}^2}\right)$, we have that w.p. at least $1-\delta/2$, for every $(\mathbf{W}, \mathbf{K})$ within our bound (e.g. every $\|\mathbf{W}_\ell\|_2, \|\mathbf{K}_\ell\|_2 \leq \widetilde{\kappa}$), it holds:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|\leq R_0, \|x\|\leq R_1}\right] \\
&\leq \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|\leq R_0, \|x\|\leq R_1}\right] + \varepsilon_{\mathsf{s}}/2 \leq \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2\right] + \varepsilon_{\mathsf{s}}/2
\end{aligned}
$$

As for the remaining terms, we again write

$$
\begin{aligned}
&\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|> R_0 \text{ or } \|x\|> R_1} \\
&\leq 4\left(G^\star(x)\right)^2 \mathbb{1}_{|G^\star(x)|>R_0/2} + R_0^2\cdot \mathbb{1}_{\|x\|>R_1} \\
&\quad + \mathsf{poly}(\widetilde{\kappa}, B')\sum_\ell \left(\|\mathbf{W}_{\ell,0}x\|^2 \mathbb{1}_{\|\mathbf{W}_{\ell,0}x\|^2 > \frac{R_0}{\mathsf{poly}(\widetilde{\kappa},B')}} + \|\mathbf{W}_{\ell,1}S_1(x)\|^2 \mathbb{1}_{\|\mathbf{W}_{\ell,1}S_1(x)\|^2 > \frac{R_0}{\mathsf{poly}(\widetilde{\kappa},B')}}\right) := RHS
\end{aligned}
$$

For this right hand side RHS, we notice that it does not depend on $\mathbf{K}$. The identical proof of Proposition J.7 in fact proves that if $R_0 = \mathsf{poly}(\widetilde{\kappa}, B')\left(c_4(2^L)\log\frac{\widetilde{\kappa}B'}{\delta\varepsilon_{\mathsf{s}}}\right)^{O(1)+c_4(2^L)}$ then for every $\mathbf{W}$ with $\|\mathbf{K}_\ell\|_2 \leq \widetilde{\kappa}$,

$$
\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[RHS] \leq \delta\varepsilon_{\mathsf{s}}/4 \ .
$$

This means, by Markov bound, for the given *fixed* $\mathbf{W}$, with probability at least $1-\delta/2$ over the randomness of $\mathcal{Z}$, it satisfies

$$
\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}[RHS] \leq \varepsilon_{\mathsf{s}}/2 \ .
$$

This implies for every $\mathbf{K}$ in the given range,

$$
\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}\left[\left(G^\star(x)-\widetilde{F}(x)\right)^2 \mathbb{1}_{|G^\star(x)-\widetilde{F}(x)|> R_0 \text{ or } \|x\|> R_1}\right] \leq \varepsilon_{\mathsf{s}}/2 \ . \qquad \square
$$

K AN IMPLICIT IMPLICATION OF OUR DISTRIBUTION ASSUMPTION

Let us define

$$
\begin{aligned}
\widehat{S}_0^\star(x) &= x \\
\widehat{S}_1^\star(x) &= \sigma(x) \\
\widehat{S}_2^\star(x) &= \mathbf{W}_{2,1}^\star \widehat{S}_1^\star(x) = \mathbf{W}_{2,1}^\star\sigma(x) \\
\widehat{S}_\ell^\star(x) &= \mathbf{W}_{\ell,\ell-1}^\star\sigma\left(\widehat{S}_{\ell-1}^\star(x)\right) \text{ for } \ell = 2,\dots,L
\end{aligned}
$$

so that $\widehat{S}_\ell^\star(x)$ is the top-degree (i.e. degree $2^{\ell-1}$) part of $S_\ell^\star(x)$.[26] We have the following implication:

Lemma K.1 (Implication of singular-value preserving)**.** *Let us define*

$$z^0 = z^0(x) = \widehat{S}_0^\star(x) = x \tag{K.1}$$

$$z^1 = z^1(x) = \widehat{S}_1^\star(x) = \sigma(x) \tag{K.2}$$

$$z^\ell = z^\ell(x) = \widehat{S}_\ell^\star(x) * \widehat{S}_\ell^\star(x) \tag{K.3}$$

Then, for every $\ell \geq \ell_1, \ell_2 \geq 0$ with $|\ell_1-\ell_2| \neq 1$, for every matrix $\mathbf{M}$: and the associated homogeneous polynomial $g_{\mathbf{M}}(x) = (z^{\ell_1})^\top \mathbf{M} z^{\ell_2}$,

- *If $\ell_1 = \ell_2 = \ell = 0$ or 1, then $\mathcal{C}_x(g_{\mathbf{M}}) = \|\mathbf{M}\|_F^2$,*
- *If $\ell_1 = \ell_2 = \ell \geq 2$, then $\mathcal{C}_x(g_{\mathbf{M}}) \geq \frac{1}{(\kappa 2^\ell)^{O(2^\ell)}}\|\mathbf{Sym}(\mathbf{M})\|_F^2$, and*
- *If $\ell_1 - 2 \geq \ell_2 \geq 0$, then $\mathcal{C}_x(g_{\mathbf{M}}) \geq \frac{1}{(\kappa 2^\ell)^{O(2^\ell)}}\|\mathbf{M}\|_F^2$ for $\ell = \ell_1$.*

[26]Meaning that $\widehat{S}_\ell^\star(x)$ is a (vector) of homogenous polynomials of x with degree $2^{\ell-1}$, and its coefficients coincide with $S_\ell^\star(x)$ on those monomials.

K.1 PROOF OF LEMMA K.1

Proof of Lemma K.1. We divide the proof into several cases.

Case A: When $\ell_1 = \ell_2 = \ell$. The situation for $\ell = 0$ or $\ell = 1$ is obvious, so below we consider $\ell \geq 2$. Let $h_\ell(z) = (z * z)\mathbf{M}(z * z) = \sum_{i \leq j, k \leq l} \mathbf{M}_{\{i,j\},\{k,l\}} a_{i,j} a_{k,l} z_i z_j z_k z_l$ be the degree-4 polynomial defined by $\mathbf{M}$. We have

$$\mathcal{C}_z(h_\ell) \geq \|\mathbf{Sym}(\mathbf{M})\|_F^2$$

For every for every $j = \ell - 1, \dots, 1$, we define $h_j(z) = h_{j+1}(\mathbf{W}^\star_{j+1,j}\sigma(z))$, it holds that

Let $\widetilde{h}(z) = h_{j+1}(\mathbf{W}^\star_{j+1,j} z)$ so that $h_j(z) = \widetilde{h}(\sigma(z))$. This means

$$\mathcal{C}(h_j) = \mathcal{C}(\widetilde{h}) \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \mathcal{C}(h_{j+1})$$

and finally we have $(z^\ell)^\top \mathbf{M} z^\ell = h_1(x)$ and therefore

$$\mathcal{C}_x\left((z^\ell)^\top \mathbf{M} z^\ell\right) \geq \frac{1}{(\kappa 2^\ell)^{O(2^\ell)}} \|\mathbf{Sym}(\mathbf{M})\|_F^2$$

Case B: When $\ell_1 - 1 > \ell_2 \geq 2$. We define $h_{\ell_1}(z, y) = (z * z)^\top \mathbf{M}(y * y)$ which is a degree-4 homogenous polynomial in (z, y), and obviously $\mathcal{C}_{y,z}(h_{\ell_1}) \geq \|\mathbf{M}\|_F^2$. Let us define

$$\forall j = \ell_1 - 1, \dots, \ell_2 + 2: \qquad h_j(z, y) = h_{j+1}\left((\mathbf{W}^\star_{j+1,j}\sigma(z), y)\right)$$

By the same argument as before, we have

$$\mathcal{C}_{z,y}(h_j) \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \mathcal{C}_{z,y}(h_{j+1})$$

Next, for $j = \ell_2$, we define

$$h_j(y) = h_{j+2}\left(\mathbf{W}^\star_{j+2,j+1}\sigma(\mathbf{W}^\star_{j+1,j}\sigma(y)), y\right)$$

To analyze this, we first define

$$h'(z, y) = h_{j+2}\left(\mathbf{W}^\star_{j+2,j+1} z, y\right) \quad \text{so that} \quad h_j(y) = h'\left(\sigma(\mathbf{W}^\star_{j+1,j}\sigma(y)), y\right)$$

Since $h'(z, y)$ is of degree 2 in the variables from y, we can write it as

$$h'(z, y) = \underbrace{\sum_p (y_p)^2 h''_{\{p,p\}}(z)}_{h''_\perp(z, \sigma(y)))} + \sum_{p<q} y_p y_q h''_{\{p,q\}}(z) \tag{K.4}$$

where the first summation contains only those quadratic terms in $(y_p)^2$ and the second contain cross terms $y_p y_q$. Note in particular if we write the first summation as $h''_\perp(z, \sigma(y)))$ for polynomial $h''_\perp(z, \gamma)$ and $\gamma = \sigma(y)$, then $h''_\perp$ is *linear* in γ. Clearly,

$$\mathcal{C}_{z,y}(h') = \mathcal{C}_{z,\gamma}(h''_\perp) + \sum_{p<q} \mathcal{C}_z(h''_{\{p,q\}}) \tag{K.5}$$

As a consequence, we can write

$$h_j(y) = \underbrace{h''_\perp(\sigma(\mathbf{W}^\star_{j+1,j}\sigma(y)), \sigma(y))}_{\widetilde{h}_\perp(y)} + y_p y_q \cdot \underbrace{h''_{\{p,q\}}(\sigma(\mathbf{W}^\star_{j+1,j}\sigma(y)))}_{\widetilde{h}_{\{p,q\}}(y)}$$

Clearly, since any polynomial in $\sigma(y)$ only contain even degrees of variables in y, so $\widetilde{h}_\perp(y)$ and each $\widetilde{h}_{\{p,q\}}$ share no common monomial, we have

$$\mathcal{C}_y(h_j) = \mathcal{C}_y(\widetilde{h}_\perp) + \sum_{p<q} \mathcal{C}_y(\widetilde{h}_{\{p,q\}}) \tag{K.6}$$

- On one hand, we have $\widetilde{h}_{\{p,q\}}(y) = h''_{\{p,q\}}(\sigma(\mathbf{W}^\star_{j+1,j}\sigma(y)))$ and therefore by previous argument

$$\mathcal{C}_y(\widetilde{h}_{\{p,q\}}) \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \mathcal{C}_z(h''_{\{p,q\}}) \tag{K.7}$$

- On the other hand, to analyze $\widetilde{h}_\perp(y)$, let us construct a square matrix $\mathbf{W} \in \mathbb{R}^{k_j \times k_j}$ with singular values between $[1/\kappa, \kappa]$ so that

$$\mathbf{W}^\star_{j+1,j}\mathbf{W} = (\mathbf{I}_{k_{j+1}\times k_{j+1}}, 0) \tag{K.8}$$

Define $h'''_\perp(z,\beta) = h''_\perp(z, \mathbf{W}\beta)$ which is linear in β, it holds:[27]

$$\begin{aligned}
\mathcal{C}_y\left(\widetilde{h}_\perp(y)\right) &= \mathcal{C}_y\left(h''_\perp(\sigma(\mathbf{W}^\star_{j+1,j}\sigma(y), \sigma(y)))\right) \\
&= \mathcal{C}_y\left(h''_\perp(\sigma(\mathbf{W}^\star_{j+1,j}y), y)\right) \\
&\geq \mathcal{C}_\beta\left(h''_\perp\left(\sigma(\mathbf{W}^\star_{j+1,j}\mathbf{W}\beta), \mathbf{W}\beta\right)\right) \cdot \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \\
&= \mathcal{C}_\beta\left(h''_\perp\left(\sigma((\mathbf{I},0)\beta), \mathbf{W}\beta\right)\right) \cdot \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \\
&= \mathcal{C}_\beta\left(h'''_\perp\left(\sigma((\mathbf{I},0)\beta), \beta\right)\right) \cdot \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \\
&\overset{①}{=} \mathcal{C}_{z,\beta}\left(h'''_\perp(z,\beta)\right) \cdot \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \\
&\geq \mathcal{C}_{z,\gamma}\left(h'''_\perp(z, \mathbf{W}^{-1}\gamma)\right) \cdot \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \\
&= \mathcal{C}_{z,\gamma}\left(h''_\perp(z,\gamma)\right) \cdot \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}}
\end{aligned} \tag{K.9}$$

Finally, plugging the lower bounds (K.7) and (K.9) into expansions (K.5) and (K.6), we conclude that

$$C_y(h_j) \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \cdot C_{z,y}(h') \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \cdot C_{z,y}(h_{j+2})$$

Continuing from here, we can define $h_j(y) = h_{j+1}(\mathbf{W}^\star_{j+1,j}\sigma(y))$ for every $j = \ell_2 - 1, \ell_2 - 2, \dots, 1$ and using the same analysis as Case A, we have

$$\mathcal{C}(h_j) = \mathcal{C}(\widetilde{h}) \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \mathcal{C}(h_{j+1})$$

and finally we have $(z^\ell)^\top \mathbf{M} z^\ell = h_1(x)$ and therefore

$$\mathcal{C}_x\left((z^\ell)^\top \mathbf{M} z^\ell\right) \geq \frac{1}{(\kappa 2^\ell)^{O(2^\ell)}} \|\mathbf{M}\|_F^2$$

$\square$

[27] Above, equality ① holds because $h'''_\perp(z,\beta)$ is a multi-variate polynomial which is linear in β, so it can be written as

$$h'''_\perp(z,\beta) = \sum_i \beta_i \cdot h'''_{\perp,i}(z)$$

for each $h'''_{\perp,i}(z)$ being a polynomial in z; next, since we plug in $z = \sigma((\mathbf{I},0)\beta)$ which only contains even-degree variables in β, we have

$$\mathcal{C}_\beta\left(h'''_\perp\left(\sigma((\mathbf{I},0)\beta), \beta\right)\right) = \sum_i \mathcal{C}_\beta\left(h'''_{\perp,i}\left(\sigma((\mathbf{I},0)\beta)\right)\right) = \sum_i \mathcal{C}_z\left(h'''_{\perp,i}(z)\right) = \mathcal{C}_{z,\gamma}\left(h'''_\perp(z,\gamma)\right)$$

Case C: When $\ell_1 - 1 > \ell_2 = 1$. Similar to Case B, we can $h_{\ell_1}(z, y) = (z * z)^\top \mathbf{M}\sigma(y)$ which is a degree-4 homogenous polynomial in (z, y), and obviously $\mathcal{C}_{y,z}(h_{\ell_1}) \geq \|\mathbf{M}\|_F^2$. Let us define

$$\forall j = \ell_1 - 1, \dots, 3\colon \qquad h_j(z, y) = h_{j+1}\left((\mathbf{W}^\star_{j+1,j}\sigma(z), y)\right)$$
$$h_1(y) = h_3\left(\mathbf{W}^\star_{3,2}\sigma(\mathbf{W}^\star_{2,1}\sigma(y)), y\right)$$

The rest of the proof now becomes identical to Case B. (In fact, we no longer have cross terms in (K.4) so the proof only becomes simpler.)

Case D: When $\ell_1 - 1 > \ell_2 = 0$. We define $h_{\ell_1}(z, y) = (z * z)^\top \mathbf{M} y$ which is a degree-3 homogenous polynomial in (z, y), and obviously $\mathcal{C}_{y,z}(h_{\ell_1}) \geq \|\mathbf{M}\|_F^2$. Let us define

$$\forall j = \ell_1 - 1, \dots, 2\colon \qquad h_j(z, y) = h_{j+1}\left((\mathbf{W}^\star_{j+1,j}\sigma(z), y)\right)$$
$$h_1(y) = h_2\left(\mathbf{W}^\star_{2,1}\sigma(y), y\right)$$

By defining $h'(z, y) = h_2(\mathbf{W}^\star_{2,1} z, y)$ we have $h_1(y) = h'(\sigma(y), y)$. This time, we have $\mathcal{C}_y(h_1) = \mathcal{C}_{z,y}(h')$, but the same proof of Case B tells us $\mathcal{C}_{z,y}(h') \geq \frac{1}{(\kappa 2^\ell)^{O(2^{\ell-j})}} \cdot \|\mathbf{M}\|_F^2$.

L Critical Lemma For Implicit Hierarchical Learning

The implicit hierarchical learning only requires one Lemma, which can be stated as the following:

Lemma L.1. *There exists absolute constant $c_0 \geq 2$ so that the following holds. Let $\tau_\ell \geq \overline{k}_\ell + L + \kappa$ and $\Upsilon_\ell \geq 1$ be arbitrary parameters for each layer $\ell \leq L$. Define parameters*

$$D_\ell := \left(\tau_\ell \cdot \kappa^{2^\ell} \cdot (2^\ell)^{2^\ell} \cdot c_1(2^\ell) \cdot c_3(2^\ell)\right)^{c_0 \ell}$$
$$C_\ell := C_{\ell-1} \cdot 2\Upsilon_\ell^3 (D_\ell)^{17} \qquad \textit{with } C_2 = 1$$

Suppose $\mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2$ for some $0 \leq \varepsilon \leq \frac{\alpha_L}{(D_L)^9 \Upsilon_L}$ and suppose the parameters satisfy

- $\frac{\alpha_{\ell+1}}{\alpha_\ell} \leq \frac{1}{C_{\ell+1}}$ *for every $\ell = 2, 3, \dots, L-1$*
- $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2] \leq \tau_\ell$ *for every $\ell = 2, 3, \dots, L-1$*
- $\lambda_{6,\ell} \geq \frac{\varepsilon^2}{\tau_\ell^2}$, $\lambda_{3,\ell} \geq \frac{\alpha_\ell^2}{D_\ell \cdot \Upsilon_\ell}$, $\lambda_{4,\ell} \geq \frac{\alpha_\ell^2}{(D_\ell)^7 \Upsilon_\ell^2}$, $\lambda_{5,\ell} \geq \frac{\alpha_\ell^2}{(D_\ell)^{13} \Upsilon_\ell^3}$ *for every $\ell = 2, 3, \dots, L$*

Then, there exist unitary matrices $\mathbf{U}_\ell$ such that for every $\ell = 2, 3, \dots, L$

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}} \|\mathbf{U}_\ell S^\star_\ell(x) - S_\ell(x)\|_2^2 \leq \left(\frac{\varepsilon}{\sqrt{\alpha_{\ell+1}\alpha_\ell}}\right)^2 C_L$$

Since we shall prove Corollary L.1 by induction, we have stated only one of the main conclusions in order for the induction to go through. Once the Theorem L.1 is proved, in fact we can strengthen it as follows.

Definition L.2. *For each $\ell \geq 2$, let $\mathbf{Q}_\ell$ be the unitary matrix defined from Lemma I.5 satisfying*

$$\mathbf{R}_\ell \mathbf{U}_\ell * \mathbf{R}_\ell \mathbf{U}_\ell = (\mathbf{R}_\ell * \mathbf{R}_\ell)\mathbf{Q}_\ell$$

We also let $\mathbf{Q}_0 = \mathbf{Q}_1 = \mathbf{I}_{d\times d}$, and let

$$\mathbf{Q}_{\ell\triangleleft} := \mathbf{diag}(\mathbf{Q}_j)_{\mathcal{J}_\ell} \quad \textit{and} \quad \vec{\mathbf{Q}}_\ell := \mathbf{diag}(\mathbf{Q}_j)_{j\in\mathcal{J}_\ell}$$

Corollary L.3. *Under the same setting as Theorem L.1, we actually have for all $\ell = 2, 3, \dots, L$,*

(a) $\left\|\mathbf{Q}_{\ell-1}^\top \overline{\mathbf{W}}_{\ell,\ell-1}^\top \overline{\mathbf{W}}_{\ell\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{\ell,\ell-1}^\top \overline{\mathbf{W}^\star}_{\ell\triangleleft}\right\|_F^2 \leq (D_\ell)^2 \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot \frac{C_L}{C_\ell}$

(b) $\left\|\mathbf{Q}_{\ell-1}^\top \overline{\mathbf{K}}_{\ell,\ell-1}^\top \overline{\mathbf{K}}_{\ell\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{\ell,\ell-1}^\top \overline{\mathbf{W}^\star}_{\ell\triangleleft}\right\|_F^2 \leq \Upsilon_\ell (D_\ell)^4 \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot \frac{C_L}{C_\ell}$

(c) $\left\|\vec{\mathbf{Q}}_\ell^\top \overline{\mathbf{K}}_\ell^\top \overline{\mathbf{K}}_\ell \vec{\mathbf{Q}}_\ell - \overline{\mathbf{W}^\star}_\ell^\top \overline{\mathbf{W}^\star}_\ell\right\|_F^2 \leq \Upsilon_\ell^2 (D_\ell)^{14} \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot \frac{C_L}{C_\ell}$

(d) $\mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_\ell S^\star_\ell(x) - S_\ell(x)\|_2^2 \leq 2\Upsilon_\ell^2 (D_\ell)^{17} \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot \frac{C_L}{C_\ell}$

(e) $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2] \leq 2B_\ell$.

Corollary L.4. *Suppose we only have $\varepsilon \leq \frac{\alpha_L}{(D_L)^3\sqrt{\Upsilon_L}}$, which is a weaker requirement comparing to Theorem L.1. Then, Theorem L.1 and Corollary L.3 still hold for the first $L-1$ layers but for ε replaced with $\alpha_L \cdot \sqrt{D_L}$. In addition, for $\ell = L$, we have*

(a) $\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F^2 \leq 2(D_L)^2 \left(\frac{\varepsilon}{\alpha_L}\right)^2$

(b) $\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F^2 \leq 2\Upsilon_L (D_L)^4 \left(\frac{\varepsilon}{\alpha_L}\right)^2$

(c) $\left\|\vec{\mathbf{Q}}_L^\top \overline{\mathbf{K}}_L^\top \overline{\mathbf{K}}_L \vec{\mathbf{Q}}_L - \overline{\mathbf{W}^\star}_L^\top \overline{\mathbf{W}^\star}_L\right\|_F^2 \leq 2\Upsilon_L^2 (D_L)^{14} \left(\frac{\varepsilon}{\alpha_L}\right)^2$

L.1 Base Case

The base case is $L = 2$. In this case, the loss function

$$\varepsilon^2 \geq \mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \geq \alpha_2^2 \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \left(\|\mathbf{W}_{2,1} S_1(x)\|^2 - \|\mathbf{W}^\star_{2,1} S_1(x)\|^2\right)^2$$

Applying the degree-preservation Property D.4, we have

$$\mathcal{C}_x \left(\|\mathbf{W}_{2,1} \widehat{S}_1(x)\|^2 - \|\mathbf{W}^\star_{2,1} \widehat{S}_1(x)\|^2\right) \leq O(1) \left(\frac{\varepsilon}{\alpha_2}\right)^2$$

where recall from Section K that $\widehat{S}_1(x) = \sigma(x)$ is the top-degree homogeneous part of $S_1(x)$, and $\mathcal{C}_x(f(x))$ is the sum of squares of f's monomial coefficients. Applying Lemma K.1, we know

$$\|\mathbf{W}_{2,1}^\top \mathbf{W}_{2,1} - (\mathbf{W}^\star_{2,1})^\top \mathbf{W}^\star_{2,1}\|_F^2 \leq O(1) \left(\frac{\varepsilon}{\alpha_2}\right)^2$$

On the other hand, our regularizer $\lambda_{4,L}$ ensures that

$$\left\|\mathbf{W}_{2,1}^\top \mathbf{W}_{2,1} - \mathbf{K}_{2,1}^\top \mathbf{K}_{2,1}\right\|_F^2 \leq \frac{\varepsilon^2}{\lambda_{4,2}} \leq (D_L)^7 \Upsilon_L^2 \left(\frac{\varepsilon}{\alpha_2}\right)^2$$

Putting them together we have

$$\left\|(\mathbf{W^\star}_{2,1})^\top \mathbf{W^\star}_{2,1} - \mathbf{K}_{2,1}^\top \mathbf{K}_{2,1}\right\|_F^2 \leq \frac{\varepsilon^2}{\lambda_{4,2}} \leq (D_L)^7 \Upsilon_L^2 \left(\frac{\varepsilon}{\alpha_2}\right)^2$$

By putting it into SVD decomposition, it is easy to derive the existence of some unitary matrix $\mathbf{U}_2$ satisfying (for a proof see Claim P.10)

$$\|\mathbf{U}_2 \mathbf{K}_{2,1} - \mathbf{W}^\star_{2,1}\|_F^2 \leq (D_L)^8 \Upsilon_L^2 \left(\frac{\varepsilon}{\alpha_2}\right)^2$$

Right multiplying it to $S_1(x)$, we have (using the isotropic Property D.1)

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\mathcal{D}} \|\mathbf{U}_2 S_2(x) - S^\star_2(x)\|_F^2 &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \|\mathbf{U}_2 \mathbf{K}_{2,1} S_1(x) - \mathbf{W}^\star_{2,1} S_1(x)\|_F^2 \\ &\leq O(1) \cdot (D_L)^8 \Upsilon_L^2 \left(\frac{\varepsilon}{\alpha_2}\right)^2 \ll \left(\frac{\varepsilon}{\sqrt{\alpha_3\alpha_2}}\right)^2\end{aligned}$$

L.2 Preparing to Prove Theorem L.1

Let us do the proof by induction with the number of layers L. Suppose this Lemma is true for every $L \leq L_0$, then let us consider $L = L_0 + 1$ Define

$$G^\star_{\leq L-1}(x) = \textstyle\sum_{\ell=2}^{L-1} \alpha_\ell \mathbf{Sum}(G^\star_\ell(x))$$

$$F_{\le L-1}(x) = \sum_{\ell=2}^{L-1} \alpha_\ell \mathbf{Sum}(F_\ell(x))$$

We know that the objective of the first $L-1$ layers

$$\begin{aligned}
\mathbf{Loss}_{L-1}(\mathcal{D}) + \mathbf{Reg}_{L-1} &= \mathbb{E}_{x\sim\mathcal{D}} \left(G^\star_{\le L-1}(x) - F_{\le L-1}(x)\right)^2 + \mathbf{Reg}_{L-1} \\
&\le 2\mathbb{E}_{x\sim\mathcal{D}} \left(G^\star(x) - F(x)\right)^2 + 2\alpha_L^2 \mathbb{E}_{x\sim\mathcal{D}} \left(\mathbf{Sum}(F_L(x)) - \mathbf{Sum}(G^\star_L(x))\right)^2 + \mathbf{Reg}_L \\
&\le 2\alpha_L^2 \mathbb{E}_{x\sim\mathcal{D}} \left(\mathbf{Sum}(F_L(x)) - \mathbf{Sum}(G^\star_L(x))\right)^2 + 2\mathbf{Loss}(\mathcal{D}) + \mathbf{Reg}\ .
\end{aligned} \tag{L.1}$$

By our assumption on the network $G^\star$, we know that for every $\ell \in [L]$,

$$\mathbb{E}_{x\sim\mathcal{D}}[\mathbf{Sum}(G^\star_\ell(x))] \le B_\ell \iff \mathbb{E}_{x\sim\mathcal{D}}[\|S^\star_\ell(x)\|^2] \le B_\ell$$

By hyper-contractivity assumption (D.3), we have that

$$\mathbb{E}_{x\sim\mathcal{D}}[(\mathbf{Sum}(G^\star_\ell(x))^2] \le c_3(2^\ell)\cdot B_\ell^2 \iff \mathbb{E}_{x\sim\mathcal{D}}[\|S^\star_\ell(x)\|^4] \le c_3(2^\ell)\cdot B_\ell^2 \tag{L.2}$$

Using our assumption $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2] \le \tau_\ell$ and the hyper-contractivity Property D.3 we also have

$$\mathbb{E}_{x\sim\mathcal{D}}[\mathbf{Sum}(F_\ell(x))] \le c_3(2^\ell)(k_\ell L \tau_\ell)^4 \quad \text{and} \quad \mathbb{E}_{x\sim\mathcal{D}}[\mathbf{Sum}(F_\ell(x))^2] \le c_3(2^\ell)(k_\ell L \tau_\ell)^8$$

Putting these into (L.1) we have

$$\mathbf{Obj}_{L-1} \le \alpha_L^2 \cdot (k_L L \overline{B}_L \tau_L)^8 c_3(2^L) + 2\varepsilon^2 \tag{L.3}$$

By induction hypothesis[28] for every L replaced with $L-1$, there exist unitary matrices $\mathbf{U}_\ell$ such that

$$\forall \ell = 2,3,\dots,L-1\colon \quad \mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_\ell S^\star_\ell(x) - S_\ell(x)\|_2^2 \le \delta_\ell^2 := \left(\frac{\alpha_L}{\sqrt{\alpha_\ell \alpha_{\ell+1}}}\right)^2 C_{L-1}\cdot (k_L L \overline{B}_L \tau_L)^8 c_3(2^L) \ll 1 \tag{L.4}$$

Let $\widehat{S}_\ell(x), \widehat{S}^\star_\ell(x)$ be the degree $2^{\ell-1}$ homogeneous part of $S_\ell(x), S^\star_\ell(x)$ respectively, notice that $\|\mathbf{U}_\ell S^\star_\ell(x) - S_\ell(x)\|_2^2$ is a polynomial of maximum degree $2^{\ell-1}$, therefore, using the degree-preservation Property D.4, we know that

$$\forall \ell = 2,3,\dots,L-1\colon \qquad \sum_{i\in[k_\ell]} \mathcal{C}_x\left(\left[\mathbf{U}_\ell \widehat{S}^\star_\ell(x) - \widehat{S}_\ell(x)\right]_i\right) \le c_1(2^\ell)\cdot \delta_\ell^2 \tag{L.5}$$

$$\forall \ell = 2,3,\dots,L\colon \qquad \sum_{i\in[k_\ell]} \mathcal{C}_x\left(\left[\widehat{S}^\star_\ell(x)\right]_i\right) \le c_1(2^\ell)\cdot B_\ell$$

We begin by proof by grouping the 2^L-degree polynomials $G^\star(x)$ and $F(x)$, into monomials of different degrees. Since

$$G^\star(x) = \sum_{\ell=2}^{L} \alpha_\ell \mathbf{Sum}(G^\star_\ell(x)) \text{ and } F(x) = \sum_{\ell=2}^{L} \alpha_\ell \mathbf{Sum}(F(x)),$$

it is clear that all the monomials with degree between $2^{L-1}+1$ and 2^L are only present in the terms $\mathbf{Sum}(G^\star_L(x))$ and $\mathbf{Sum}(F_L(x))$ respectively. Recall also (we assume L is even for the rest of the proof, and the odd case is analogous).

$$\mathbf{Sum}(G^\star_L(x)) = \left\|\sum_{\ell\in\mathcal{J}_L\setminus\{0,1\}} \mathbf{W}^\star_{L,\ell}\sigma(S^\star_\ell(x)) + \sum_{\ell\in\mathcal{J}_L\cap\{0,1\}} \mathbf{W}^\star_{L,\ell} S^\star_\ell(x)\right\|^2 \tag{L.6}$$

$$\mathbf{Sum}(F_L(x)) = \left\|\sum_{\ell\in\mathcal{J}_L\setminus\{0,1\}} \mathbf{W}_{L,\ell}\sigma(\mathbf{R}_\ell S_\ell(x)) + \sum_{\ell\in\mathcal{J}_L\cap\{0,1\}} \mathbf{W}_{L,\ell} S_\ell(x)\right\|^2$$

[28]To be precise, using our assumption on $\frac{\alpha_L}{\alpha_{L-1}}$ one can verify that $O\big(\alpha_L^2\cdot(k_L L\overline{B}_L\tau_L)^8 c_3(2^\ell)\big) \le \frac{\alpha_{L-1}^2}{2(D_{L-1})^8\sqrt{\Upsilon_{L-1}^3}}$ so the assumption from the inductive case holds.

L.3 DEGREE 2^L

We first consider all the monomials from $G^\star(x)$ and $F(x)$ in degree $2^{L-1}+2^{L-1}=2^L$ (i.e., top degree). As argued above, they must come from the top degree of (L.6).

Let $\widehat{G^\star}_L, \widehat{F}_L\colon \mathbb{R}^d \to \mathbb{R}^{k_L}$ be the degree 2^L part of $G^\star_L(x), F_L(x)$ respectively. Using

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}} |F(x)-G^\star(x)|^2 \le \mathbf{Obj} \le \varepsilon^2$$

and the degree-preservation Property D.4 again, we have

$$\mathcal{C}_x\left(\mathbf{Sum}(\widehat{F}_L(x)) - \mathbf{Sum}(\widehat{G^\star}_L(x))\right) \le c_1(2^L)\left(\frac{\varepsilon}{\alpha_L}\right)^2 \tag{L.7}$$

From (L.6), we know that

$$\mathbf{Sum}(\widehat{G^\star}_L(x)) = \left\|\mathbf{W}^\star_{L,L-1}\sigma\left(\widehat{S}^\star_{L-1}(x)\right)\right\|^2 = \left\|\overline{\mathbf{W}^\star}_{L,L-1}\left(\widehat{S}^\star_{L-1}(x)*\widehat{S}^\star_{L-1}(x)\right)\right\|^2$$

We also have

$$\mathbf{Sum}(\widehat{F}_L(x)) = \left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\widehat{S}_{L-1}(x)\right)\right\|^2 = \left\|\overline{\mathbf{W}}_{L,L-1}\left(\widehat{S}_{L-1}(x)*\widehat{S}_{L-1}(x)\right)\right\|^2$$

For analysis, we also define $\overline{\overline{\mathbf{W}}}_{L,L-1} = \mathbf{W}_{L,L-1}(\mathbf{R}_{L-1}\mathbf{U}_{L-1} * \mathbf{R}_{L-1}\mathbf{U}_{L-1}) \in \mathbb{R}^{k_L\times\binom{k_{L-1}+1}{2}}$ so that

$$\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\mathbf{U}_{L-1}\widehat{S}^\star_{L-1}(x)\right) = \overline{\overline{\mathbf{W}}}_{L,L-1}\left(\widehat{S}^\star_{L-1}(x)*\widehat{S}^\star_{L-1}(x)\right)$$

where $\overline{\overline{\mathbf{W}}}_{L,L-1} = \overline{\mathbf{W}}_{L,L-1}\mathbf{Q}_{L-1}$ for a unitary matrix $\mathbf{Q}_{L-1}$ by Lemma I.5.

Using $\sum_{i\in[k_\ell]}\mathcal{C}_x\left(\left[\mathbf{U}_\ell\widehat{S}^\star_\ell(x)-\widehat{S}_\ell(x)\right]_i\right) \le c_1(2^\ell)\cdot\delta_\ell^2$ from (L.5) and $\sum_{i\in[k_\ell]}\mathcal{C}_x\left(\left[\widehat{S}^\star_\ell(x)\right]_i\right) \le c_1(2^\ell)B_\ell$, it is not hard to derive that[29]

$$\mathcal{C}_x\left(\left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\widehat{S}_{L-1}(x)\right)\right\|^2 - \left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\mathbf{U}_{L-1}\widehat{S}^\star_{L-1}(x)\right)\right\|^2\right) \le \xi_1$$
$$\text{for some } \xi_1 \le \tau_L^6\cdot\mathsf{poly}(\overline{B}_L, 2^{2^L}, c_1(2^L))\delta_{L-1}^2. \tag{L.8}$$

Combining (L.7) and (L.8) with the fact that $\mathcal{C}_x(f_1+f_2)\le 2\mathcal{C}_x(f_1)+2\mathcal{C}_x(f_2)$, we have

$$\mathcal{C}_x\left(\left\|\overline{\mathbf{W}^\star}_{L,L-1}\left(\widehat{S}^\star_{L-1}(x)*\widehat{S}^\star_{L-1}(x)\right)\right\|^2 - \left\|\overline{\overline{\mathbf{W}}}_{L,L-1}\left(\widehat{S}^\star_{L-1}(x)*\widehat{S}^\star_{L-1}(x)\right)\right\|^2\right) = \xi_2$$
$$\text{for some } \xi_2 \le \tau_L^6\cdot\mathsf{poly}(\overline{B}_L, 2^{2^L}, c_1(2^L))\delta_{L-1}^2 + 2c_1(2^L)\left(\frac{\varepsilon}{\alpha_L}\right)^2$$

Applying the singular value property Lemma K.1 to the above formula, we have

$$\left\|\mathbf{Sym}\left(\overline{\overline{\mathbf{W}}}_{L,L-1}^\top\overline{\overline{\mathbf{W}}}_{L,L-1}\right) - \mathbf{Sym}\left(\overline{\mathbf{W}^\star}_{L,L-1}^\top\overline{\mathbf{W}^\star}_{L,L-1}\right)\right\|_F \le \mathsf{poly}_1\left(\frac{\varepsilon}{\alpha_L}+\tau_L^3\delta_{L-1}\right) \tag{L.9}$$

for some sufficiently large polynomial

$$\mathsf{poly}_1 = \mathsf{poly}(\overline{B}_L, \kappa^{2^L}, (2^L)^{2^L}, c_1(2^L), c_3(2^L))$$

[29] Indeed, if we define $g(z) = \|\mathbf{W}_{L,L-1}\sigma(\mathbf{R}z)\|^2 = \|\overline{\mathbf{W}}_{L,L-1}(z*z)\|^2$ then we have $\mathcal{C}_z(g) \le O(1)\cdot\|\overline{\mathbf{W}}_{L,L-1}\|_F^2$ using Fact I.4, and therefore $\mathcal{C}_z(g)\le O(\tau_L^2L^2)$ using $\|\mathbf{W}_{L,L-1}\|_F \le \tau_L$ and $\|\mathbf{R}_{L-1}*\mathbf{R}_{L-1}\|_2 \le O(L)$ from Lemma I.6. Next, we apply Lemma P.7 with $f^{(1)}(x) = \mathbf{U}_{L-1}\widehat{S}^\star_{L-1}(x)$ and $f^{(2)}(x)=\widehat{S}_{L-1}(x)$ to derive the bound

$$\mathcal{C}_x(g(f_1(x)) - g(f_2(x))) \le k_L^4\cdot 2^{O(2^L)}\cdot(c_1(2^L))^8\cdot(\delta_{L-1}^8+\delta_{L-1}^2\overline{B}_L^3)\cdot\mathcal{C}_z(g)\ .$$

This implies

$$
\begin{aligned}
\left\|\mathbf{W}^\star_{L,L-1}\sigma\left(S^\star_{L-1}(x)\right)\right\|^2 &= \left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right)^\top \overline{\mathbf{W}^\star}^\top_{L,L-1}\overline{\mathbf{W}^\star}_{L,L-1}\left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right) \\
&\overset{①}{=} \left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right)^\top \mathbf{Sym}\left(\overline{\mathbf{W}^\star}^\top_{L,L-1}\overline{\mathbf{W}^\star}_{L,L-1}\right)\left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right) \\
&\overset{②}{=} \left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right)^\top \mathbf{Sym}\left(\overline{\overline{\mathbf{W}}}^\top_{L,L-1}\overline{\overline{\mathbf{W}}}_{L,L-1}\right)\left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right) + \xi_3 \\
&\overset{③}{=} \left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right)^\top \overline{\overline{\mathbf{W}}}^\top_{L,L-1}\overline{\overline{\mathbf{W}}}_{L,L-1}\left(S^\star_{L-1}(x) * S^\star_{L-1}(x)\right) + \xi_3 \\
&= \left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\mathbf{U}_{L-1}S^\star_{L-1}(x)\right)\right\|^2 + \xi_3 \\
&= \left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}S_{L-1}(x)\right)\right\|^2 + \xi_4 \qquad \text{(L.10)}
\end{aligned}
$$

Above, ① and ③ hold because of Fact I.4. ② holds for some error term ξ_3 with

$$\mathbb{E}[(\xi_3)^2] \le (\mathsf{poly}_1)^2 \cdot \left(\frac{\varepsilon}{\alpha_L} + \tau_L^3\delta_{L-1}\right)^2$$

because of (L.9) and $\mathbb{E}_{x\sim\mathcal{D}}[\|S^\star_\ell(x)\|^2] \le B_\ell$ together with the hyper-contractivity Property D.3. ④ holds for

$$\mathbb{E}[(\xi_4)^2] \le (\mathsf{poly}_1)^3 \cdot \left(\frac{\varepsilon}{\alpha_L} + \tau_L^3\delta_{L-1}\right)^2$$

because of $\mathbb{E}_{x\sim\mathcal{D}}\left\|\mathbf{U}_{L-1}S^\star_{L-1}(x) - S_{L-1}(x)\right\|^2 \le c_1(2^{L-1})\cdot\delta^2_{L-1}$ which implies[30]

$$
\begin{aligned}
&\left|\left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}S_{L-1}(x)\right)\right\|^2 - \left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\mathbf{U}_{L-1}S^\star_{L-1}(x)\right)\right\|^2\right| = \xi_4' \\
&\qquad\text{for some } \xi_4' \in \mathbb{R} \text{ with } \mathbb{E}_{x\sim\mathcal{D}}[(\xi_1)^2] \le \tau_L^{12}\cdot\mathsf{poly}(\overline{B}_L, c_3(2^L))\delta^2_{L-1}. \qquad \text{(L.11)}
\end{aligned}
$$

L.4 Degree $2^{L-1}+2^{L-3}$ Or Lower

Let us without loss of generality assuming that $L-3 \in \mathcal{J}_L$, otherwise we move to lower degrees. We now describe the strategy for this weight matrix $\mathbf{W}_{L,L-3}$.

Let us consider all the monomials from $G^\star(x)$ and $F(x)$ in degree $2^{L-1}+2^{L-3}$. As argued above, they must come from equation (L.6).

As for the degree $2^{L-1}+2^{L-3}$ degree monomials in $G^\star(x)$ and $F(x)$, either they come from

$$\left\|\mathbf{W}^\star_{L,L-1}\sigma(S^\star_{L-1}(x))\right\|^2 \text{ and } \left\|\mathbf{W}_{L,L-1}\sigma(\mathbf{R}_{L-1}S_{L-1}(x))\right\|^2,$$

which as we have argued in (L.10), they are sufficiently close; or they come from

$$\sigma\left(\widehat{S}^\star_{L-3}(x)\right)^\top\left(\mathbf{W}^\star_{L,L-3}\right)^\top\mathbf{W}^\star_{L,L-1}\sigma\left(\widehat{S}^\star_{L-1}(x)\right) \qquad \text{from } \mathbf{Sum}(G^\star_{L-1}(x))$$

[30]Specifically, one can combine
- $\|\sigma(a)-\sigma(b)\| \le \|a-b\|\cdot(\|a\|+2\|a-b\|)$,
- $(\|\mathbf{W}_{L,L-1}a\|^2 - \|\mathbf{W}_{L,L-1}b\|^2)^2 \le \|\mathbf{W}_{L,L-1}(a-b)\|^2\cdot(2\|\mathbf{W}_{L,L-1}a\| + \|\mathbf{W}_{L,L-1}(a-b)\|)^2$,
- the spectral norm bound $\|\mathbf{W}_{L,L-1}\|_2 \le \tau_L$, $\|\mathbf{R}_{L-1}\|_2 \le O(\tau_L)$,

to derive that

$$
\begin{aligned}
&\left(\left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}S_{L-1}(x)\right)\right\|^2 - \left\|\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\mathbf{U}_{L-1}S^\star_{L-1}(x)\right)\right\|^2\right)^2 \\
&\le O(\tau_L^{12})\cdot\left(\left\|S^\star_\ell(x)\right\|^6\left\|\mathbf{U}_\ell S^\star_\ell(x) - S_\ell(x)\right\|^2 + \left\|\mathbf{U}_\ell S^\star_\ell(x) - S_\ell(x)\right\|^8\right)
\end{aligned}
$$

Using $\|a\|^6\|b\|^2 \le O(\delta^2_{L-1}\|a\|^{12} + \frac{\|b\|^4}{\delta^2_{L-1}})$, as well as the aforementioned bounds
- $\mathbb{E}_{x\sim\mathcal{D}}\left\|S^\star_{L-1}(x)\right\|^2 \le \overline{B}_L$ and $\mathbb{E}_{x\sim\mathcal{D}}\left\|\mathbf{U}_{L-1}S^\star_{L-1}(x) - S_{L-1}(x)\right\|^2 \le \delta^2_{L-1}$

and the hyper-contractivity assumption (D.3), we can prove (L.11).

$$\sigma\left(\mathbf{R}_{L-3}\widehat{S}_{L-3}(x)\right)^\top(\mathbf{W}_{L,L-3})^\top\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\widehat{S}_{L-1}(x)\right) \qquad \text{from } \mathbf{Sum}(F_{L-1}(x))$$

For this reason, suppose we compare the following two polynomials

$$G^\star(x)-\alpha_L\left\|\mathbf{W}^\star_{L,L-1}\sigma(S^\star_{L-1}(x))\right\|^2 \quad \text{vs} \quad F(x)-\alpha_L\left\|\mathbf{W}_{L,L-1}\sigma(\mathbf{R}_{L-1}S_{L-1}(x))\right\|^2 \ ,$$

they are both of degree at most $2^{L-1}+2^{L-3}$, and they differ by an error term

$$\xi_5=\left(G^\star(x)-\alpha_L\left\|\mathbf{W}^\star_{L,L-1}\sigma(S^\star_{L-1}(x))\right\|^2\right)-\left(F(x)-\alpha_L\left\|\mathbf{W}_{L,L-1}\sigma(\mathbf{R}_{L-1}S_{L-1}(x))\right\|^2\right)$$

which satisfies (using $\mathbf{Obj}\le\varepsilon^2$ together with (L.10))

$$\mathbb{E}_{x\sim\mathcal{D}}[(\xi_5)^2]\le(\mathsf{poly}_1)^4\cdot\left(\varepsilon+\tau_L^3\alpha_L\delta_{L-1}\right)^2$$

Using and the degree-preservation Property D.4 again (for the top degree $2^{L-1}+2^{L-3}$), we have

$$\begin{aligned}&\mathcal{C}_x\bigg(\sigma\left(\widehat{S}^\star_{L-3}(x)\right)^\top\left(\mathbf{W}^\star_{L,L-3}\right)^\top\mathbf{W}^\star_{L,L-1}\sigma\left(\widehat{S}^\star_{L-1}(x)\right)\\&-\sigma\left(\mathbf{R}_{L-3}\widehat{S}_{L-3}(x)\right)^\top\left(\mathbf{W}_{L,L-3}\right)^\top\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\widehat{S}_{L-1}(x)\right)\bigg)\le\xi_6^2\end{aligned}$$

for some error term ξ_6 with $[(\xi_6)^2]\le(\mathsf{poly}_1)^5\cdot\left(\frac{\varepsilon}{\alpha_L}+\tau_L^3\delta_{L-1}\right)^2$. Using a similar argument as (L.8), we also have

$$\begin{aligned}&\mathcal{C}_x\bigg(\left(\mathbf{R}_{L-3}\widehat{S}_{L-3}(x)\right)^\top\left(\mathbf{W}_{L,L-3}\right)^\top\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\widehat{S}_{L-1}(x)\right)\\&-\sigma\left(\mathbf{R}_{L-3}\mathbf{U}_{L-3}\widehat{S}^\star_{L-3}(x)\right)^\top\left(\mathbf{W}_{L,L-3}\right)^\top\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}\mathbf{U}_{L-3}\widehat{S}^\star_{L-1}(x)\right)\bigg)\le\xi_7\end{aligned}$$

for $\xi_7\le\tau_L^6\cdot\mathsf{poly}(\overline{B}_L,2^{2^L},c_1(2^L))\delta_{L-1}^2$. If we define $\overline{\overline{\mathbf{W}}}_{L,L-3}=\overline{\mathbf{W}}_{L,L-3}\mathbf{Q}_{L-1}$ for the same unitary matrix $\mathbf{Q}_{L-1}$ as before, we have

$$\mathbf{W}_{L,L-3}\sigma\left(\mathbf{R}_{L-3}\mathbf{U}_{L-3}\widehat{S}^\star_{L-2}(x)\right)=\overline{\overline{\mathbf{W}}}_{L,L-3}\left(\widehat{S}^\star_{L-3}(x)*\widehat{S}^\star_{L-3}(x)\right) \ .$$

Using this notation, the error bounds on ξ_6 and ξ_7 together imply

$$\begin{aligned}&\mathcal{C}_x\bigg(\left(\widehat{S}^\star_{L-3}(x)*\widehat{S}^\star_{L-3}(x)\right)^\top\overline{\mathbf{W}^\star}^\top_{L,L-3}\overline{\mathbf{W}^\star}_{L,L-1}\left(\widehat{S}^\star_{L-1}(x)*\widehat{S}^\star_{L-1}(x)\right)\\&-\left(\widehat{S}^\star_{L-3}(x)*\widehat{S}^\star_{L-3}(x)\right)^\top\overline{\overline{\mathbf{W}}}^\top_{L,L-3}\overline{\overline{\mathbf{W}}}_{L,L-1}\left(\widehat{S}^\star_{L-1}(x)*\widehat{S}^\star_{L-1}(x)\right)\bigg)^2\le\xi_8\end{aligned}$$

for $\xi_8\le(\mathsf{poly}_1)^6\cdot\left(\frac{\varepsilon}{\alpha_L}+\tau_L^3\delta_{L-1}\right)^2$. Applying the singular value property Lemma K.1 to the above formula, we have

$$\left\|\overline{\overline{\mathbf{W}}}^\top_{L,L-3}\overline{\overline{\mathbf{W}}}_{L,L-1}-\overline{\mathbf{W}^\star}^\top_{L,L-3}\overline{\mathbf{W}^\star}_{L,L-1}\right\|_F^2\le(\mathsf{poly}_1)^7\left(\frac{\varepsilon}{\alpha_L}+\tau_L^3\delta_{L-1}\right)^2 \ . \tag{L.12}$$

Following a similar argument to (L.10), we can derive that This implies

$$\begin{aligned}&(\mathbf{W}^\star_{L,L-3}\sigma\left(S^\star_{L-3}(x)\right))^\top\mathbf{W}^\star_{L,L-1}\sigma\left(S^\star_{L-1}(x)\right)\\&=(\mathbf{W}_{L,L-3}\sigma\left(\mathbf{R}_{L-3}S_{L-3}(x)\right))^\top\mathbf{W}_{L,L-1}\sigma\left(\mathbf{R}_{L-1}S_{L-1}(x)\right)+\xi_9\end{aligned}$$

for some $\mathbb{E}[(\xi_9)^2]\le(\mathsf{poly}_1)^8\left(\frac{\varepsilon}{\alpha_L}+\tau_L^3\delta_{L-1}\right)^2$

L.5 Until Degree $2^{L-1}+1$

If we repeat the process in Section L.4 to analyze monomials of degrees $2^{L-1}+2^j$ until $2^{L-1}+1$ (for all $j \in \mathcal{J}_L$), eventually we can conclude that[31]

$$\left\| \overline{\overline{\mathbf{W}}}_{L,L-1}^\top \overline{\overline{\mathbf{W}}}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft} \right\|_F \leq (\mathsf{poly}_1)^{2L+3}\left(\frac{\varepsilon}{\alpha_L} + \tau_L^3 \delta_{L-1}\right)$$

which implies that for unitary matrix $\mathbf{Q}_{L\triangleleft} := \operatorname{diag}(\mathbf{Q}_\ell)_{\ell \in \mathcal{J}_L \setminus \{L-1\}}$, we have that

$$\left\| \mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft} \right\|_F \leq (\mathsf{poly}_1)^{2L+3}\left(\frac{\varepsilon}{\alpha_L} + \tau_L^3 \delta_{L-1}\right)$$

Let us define

$$\mathsf{poly}_2 = (\mathsf{poly}_1)^{2L+3}\tau_L^3 \qquad \text{(we eventually choose } D_L = \mathsf{poly}_2\text{)}$$

so that

$$\left\| \mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft} \right\|_F \leq \mathsf{poly}_2\left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \tag{L.13}$$

By the regularizer that

$$\left\| \mathbf{W}_{L,L-1}^\top \mathbf{W}_{L\triangleleft} - \mathbf{K}_{L,L-1}^\top \mathbf{K}_{L\triangleleft} \right\|_F^2 \leq \frac{\varepsilon^2}{\lambda_{3,L}}$$

Using $\overline{\mathbf{W}}_{L,j} = \mathbf{W}_{L,j}(\mathbf{R}_j * \mathbf{R}_j)$ and $\overline{\mathbf{K}}_{L,j} = \mathbf{K}_{L,j}(\mathbf{R}_j * \mathbf{R}_j)$, using the properties that $\mathbf{R}_j * \mathbf{R}_j$ is well-conditioned (see Lemma I.6), and using $\mathbf{Q}_{L-1}$ and $\mathbf{Q}_{L\triangleleft}$ are unitary (see Lemma I.5), we have

$$\left\| \mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} \right\|_F^2 \leq \frac{\varepsilon^2}{\lambda_{3,L}} \cdot \mathsf{poly}(\overline{k}_L, L) \tag{L.14}$$

By our choice of $\lambda_{3,L} \geq \frac{1}{\mathsf{poly}_2 \cdot \Upsilon_L}\alpha_L^2$ and (L.13), we have

$$\left\| \mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft} \right\|_F \leq \sqrt{\Upsilon_L}(\mathsf{poly}_2)^2\left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \tag{L.15}$$

L.6 Deriving $\overline{\mathbf{K}}_L$ Close To $\overline{\mathbf{W}^\star}_L$

Since $\|\mathbf{K}_{L,\triangleleft}\|_F, \|\mathbf{K}_{L,L-1}\|_F \leq \tau_L$, we have $\|\overline{\mathbf{K}}_{L,\triangleleft}\|_F, \|\overline{\mathbf{K}}_{L,L-1}\|_F \leq O(\tau_L L)$ from Lemma I.6. Also, the singular values of $\overline{\mathbf{W}^\star}_{L\triangleleft}, \overline{\mathbf{W}^\star}_{L,L-1}$ are between $1/\kappa$ and $L\kappa$ (see Fact I.7). Therefore, applying Claim P.9 to (L.15), we know that there exists square matrix $\mathbf{P} \in \mathbb{R}^{k_L \times k_L}$ satisfying[32]

$$\left\| \overline{\mathbf{K}}_{L,L-1}\mathbf{Q}_{L-1} - \mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1} \right\|_F \leq \sqrt{\Upsilon_L}(\mathsf{poly}_2)^3\left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)$$

$$\left\| \overline{\mathbf{K}}_{L\triangleleft}\mathbf{Q}_{L\triangleleft} - (\mathbf{P}^\top)^{-1}\overline{\mathbf{W}^\star}_{L\triangleleft} \right\|_F \leq \sqrt{\Upsilon_L}(\mathsf{poly}_2)^3\left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)$$

and all the singular values of $\mathbf{P}$ are between $\frac{1}{\mathsf{poly}(\tau_L)}$ and $\mathsf{poly}(\tau_L)$. This implies that

$$\left\| \mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L,L-1} \mathbf{Q}_{L-1} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top \mathbf{P} \overline{\mathbf{W}^\star}_{L,L-1} \right\|_F \leq \sqrt{\Upsilon_L}(\mathsf{poly}_2)^4\left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \tag{L.16}$$

---

[31] Technically speaking, for $j \in \mathcal{J}_L \cap \{0,1\}$, one needs to modify Section L.4 a bit, because the 4-tensor becomes 3-tensor: $\left(\widehat{S}_j^\star(x)\right)^\top \overline{\mathbf{W}^\star}_{L,j}^\top \overline{\mathbf{W}^\star}_{L,L-1}\left(\widehat{S}_{L-1}^\star(x) * \widehat{S}_{L-1}^\star(x)\right)$.

[32] We note here, to apply Claim P.9, one also needs to ensure $\varepsilon \leq \frac{\alpha_L}{(\mathsf{poly}_2)^3\sqrt{\Upsilon_L}}$ and $\delta_{L-1} \leq \frac{1}{(\mathsf{poly}_2)^3\sqrt{\Upsilon_L}}$; however, both of them are satisfied under the assumptions $\varepsilon \leq \frac{\alpha_L}{(D_L)^9\Upsilon_L}$ and $\frac{\alpha_L}{\alpha_{L-1}} \leq \frac{1}{4\Upsilon_L^3(D_L)^{16}C_{L-1}}$, and the definition of δ_{L-1} from (L.4).

$$\left\|\mathbf{Q}_{L\triangleleft}^\top \overline{\mathbf{K}}_{L\triangleleft}^\top \overline{\mathbf{K}}_{L\triangleleft}\mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L\triangleleft}^\top (\mathbf{P}^\top\mathbf{P})^{-1}\overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F \le \sqrt{\Upsilon_L}(\mathsf{poly}_2)^4 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \tag{L.17}$$

Our regularizer $\lambda_{4,L}$ ensures that

$$\left\|\mathbf{W}_{L,L-1}^\top \mathbf{W}_{L,L-1} - \mathbf{K}_{L,L-1}^\top \mathbf{K}_{L,L-1}\right\|_F^2 \le \frac{\varepsilon^2}{\lambda_{4,L}}$$

Using $\overline{\mathbf{W}}_{L,j} = \mathbf{W}_{L,j}(\mathbf{R}_j * \mathbf{R}_j)$ and $\overline{\mathbf{K}}_{L,j} = \mathbf{K}_{L,j}(\mathbf{R}_j * \mathbf{R}_j)$, using the properties that $\mathbf{R}_j * \mathbf{R}_j$ is well-conditioned (see Lemma I.6), and using $\mathbf{Q}_{L-1}$ and $\mathbf{Q}_{L\triangleleft}$ are unitary (see Lemma I.5), we have

$$\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L,L-1}\mathbf{Q}_{L-1} - \mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L,L-1}\mathbf{Q}_{L-1}\right\|_F^2 \le \frac{\varepsilon^2}{\lambda_{4,L}} \cdot \mathsf{poly}(\overline{k}_L, L)$$

By our choice $\lambda_{4,L} \ge \frac{1}{(\mathsf{poly}_2)^7\sqrt{\Upsilon_L^2}}\alpha_L^2$, this together with (L.16) implies

$$\begin{aligned}
&\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L,L-1}\mathbf{Q}_{L-1} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top\mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1}\right\|_F \le 2\sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^4 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \\
\iff &\left\|\overline{\overline{\mathbf{W}}}_{L,L-1}^\top \overline{\overline{\mathbf{W}}}_{L,L-1} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top\mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1}\right\|_F \le 2\sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^4 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)
\end{aligned} \tag{L.18}$$

Recall we have already concluded in (L.9) that

$$\left\|\mathbf{Sym}\left(\overline{\overline{\mathbf{W}}}_{L,L-1}^\top \overline{\overline{\mathbf{W}}}_{L,L-1}\right) - \mathbf{Sym}\left(\overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L,L-1}\right)\right\|_F \le \mathsf{poly}_2 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)$$

so putting it into (L.18) we have

$$\left\|\mathbf{Sym}\left(\overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top\mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1}\right) - \mathbf{Sym}\left(\overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L,L-1}\right)\right\|_F \le 3\sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^4 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)$$

Since $\overline{\mathbf{W}^\star}_{L,L-1} = \mathbf{W}^\star_{L,L-1}$, by Fact I.4, we know that for any matrix $\mathbf{P}$,

$$\mathbf{Sym}\left(\overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top\mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1}\right) = \overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top\mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1}$$

This implies

$$\left\|\overline{\mathbf{W}^\star}_{L,L-1}^\top \mathbf{P}^\top\mathbf{P}\overline{\mathbf{W}^\star}_{L,L-1} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L,L-1}\right\|_F \le 4\sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^4 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) .$$

By expanding $\overline{\mathbf{W}^\star}_{L,L-1}$ into its SVD decomposition, one can derive from the above inequality that

$$\left\|\mathbf{P}^\top\mathbf{P} - \mathbf{I}\right\|_F \le \sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^5 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \tag{L.19}$$

Putting this back to (L.16) and (L.17), we have

$$\begin{aligned}
\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L,L-1}\mathbf{Q}_{L-1} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L,L-1}\right\|_F &\le \sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^6 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \\
\left\|\mathbf{Q}_{L\triangleleft}^\top \overline{\mathbf{K}}_{L\triangleleft}^\top \overline{\mathbf{K}}_{L\triangleleft}\mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L\triangleleft}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F &\le \sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^6 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)
\end{aligned}$$

Combining this with (L.15), we derive that (denoting by $\vec{\mathbf{Q}}_L := \operatorname{diag}(\mathbf{Q}_\ell)_{\ell\in\mathcal{J}_L}$)

$$\left\|\vec{\mathbf{Q}}_L^\top \overline{\mathbf{K}}_L^\top \overline{\mathbf{K}}_L \vec{\mathbf{Q}}_L - \overline{\mathbf{W}^\star}_L^\top \overline{\mathbf{W}^\star}_L\right\|_F \le \sqrt{\Upsilon_L^2}(\mathsf{poly}_2)^7 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right) \tag{L.20}$$

L.7 DERIVING $S_L(x)$ CLOSE TO $S^\star_L(x)$, CONSTRUCT $\mathbf{U}_L$

From (L.20) we can also apply Claim P.10 and derive the existence of some unitary $\mathbf{U}_L \in \mathbb{R}^{k_L \times k_L}$ so that[33]

$$\left\| \overline{\mathbf{K}}_L \vec{\mathbf{Q}}_L - \mathbf{U}_L \overline{\mathbf{W}^\star}_L \right\|_F \leq \sqrt{\Upsilon_L^2} (\mathsf{poly}_2)^8 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1} \right) \quad . \tag{L.21}$$

Simultaneously right applying the two matrices in (L.21) by the vector (where the operator $\frown$ is for concatenating two vectors)

$$\left(S^\star_j(x) * S^\star_j(x) \right)_{j \in \mathcal{J}_L \setminus \{0,1\}} \frown \left(S^\star_j(x) \right)_{j \in \mathcal{J}_L \setminus \{0,1\}} \ ,$$

we have

$$\begin{aligned} &\sum_{j \in \mathcal{J}_L \setminus \{0,1\}} \mathbf{K}_{L,j} \sigma \left(\mathbf{R}_j \mathbf{U}_j S^\star_j(x) \right) + \sum_{j \in \mathcal{J}_L \cap \{0,1\}} \mathbf{K}_{L,j} S^\star_j(x) \\ &= \mathbf{U}_L \left(\sum_{j \in \mathcal{J}_L \setminus \{0,1\}} \mathbf{W}^\star_{L,j} \sigma \left(S^\star_j(x) \right) + \sum_{j \in \mathcal{J}_L \cap \{0,1\}} \mathbf{W}^\star_{L,j} S^\star_j(x) \right) + \xi_{10} \end{aligned}$$

for some error vector ξ_{10} with

$$\mathbb{E}_{x \sim \mathcal{D}} [\|\xi_{10}\|^2] \leq \Upsilon_L^2 \cdot L \overline{B}_L^2 (\mathsf{poly}_2)^{16} \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1} \right)^2 \quad .$$

Combining it with $\mathbb{E}_{x \sim \mathcal{D}} \left\| \mathbf{U}_{L-1} S^\star_{L-1}(x) - S_{L-1}(x) \right\|_2^2 \leq \delta_{L-1}^2$ (see (L.4)) we know

$$\begin{aligned} S_L(x) &= \sum_{j \in \mathcal{J}_L \setminus \{0,1\}} \mathbf{K}_{L,j} \sigma \left(\mathbf{R}_j S_j(x) \right) + \sum_{j \in \mathcal{J}_L \cap \{0,1\}} \mathbf{K}_{L,j} S_j(x) \\ &= \mathbf{U}_L \left(\sum_{j \in \mathcal{J}_L \setminus \{0,1\}} \mathbf{W}^\star_{L,j} \sigma \left(S^\star_j(x) \right) + \sum_{j \in \mathcal{J}_L \cap \{0,1\}} \mathbf{W}^\star_{L,j} S^\star_j(x) \right) + \xi_{11} = \mathbf{U}_L S^\star_L(x) + \xi_{11} \end{aligned}$$

for some error vector ξ_{11} with

$$\mathbb{E}_{x \sim \mathcal{D}} [\|\xi_{11}\|^2] = \mathbb{E}_{x \sim \mathcal{D}} \|\mathbf{U}_L S^\star_L(x) - S_L(x)\|_2^2 \leq \Upsilon_L^2 (\mathsf{poly}_2)^{17} \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1} \right)^2 \quad . \tag{L.22}$$

L.8 DERIVING $F_L(x)$ CLOSE TO $G^\star(x)$

By the regularizer $\lambda_{5,L}$, we have that

$$\left\| \mathbf{W}_L^\top \mathbf{W}_L - \mathbf{K}_L^\top \mathbf{K}_L \right\|_F \leq \frac{\varepsilon^2}{\lambda_{5,L}} \tag{L.23}$$

Using $\overline{\mathbf{W}}_{L,j} = \mathbf{W}_{L,j}(\mathbf{R}_j * \mathbf{R}_j)$ and $\overline{\mathbf{K}}_{L,j} = \mathbf{K}_{L,j}(\mathbf{R}_j * \mathbf{R}_j)$, using the properties that $\mathbf{R}_j * \mathbf{R}_j$ is well-conditioned (see Lemma I.6), and using $\mathbf{Q}_{L-1}$ and $\mathbf{Q}_{L\triangleleft}$ are unitary (see Lemma I.5), we have

$$\left\| \vec{\mathbf{Q}}_L^\top \overline{\mathbf{W}}_L^\top \overline{\mathbf{W}}_L \vec{\mathbf{Q}}_L - \vec{\mathbf{Q}}_L^\top \overline{\mathbf{K}}_L^\top \overline{\mathbf{K}}_L \vec{\mathbf{Q}}_L \right\|_F^2 \leq \frac{\varepsilon^2}{\lambda_{5,L}} \cdot \mathsf{poly}(\overline{k}_L, L)$$

By our choice of $\lambda_{5,L} \geq \frac{1}{(\mathsf{poly}_2)^{13} \Upsilon_L^3} \alpha_L^2$, together with (L.20), we have that

$$\left\| \vec{\mathbf{Q}}_L^\top \overline{\mathbf{W}}_L^\top \overline{\mathbf{W}}_L \vec{\mathbf{Q}}_L - \overline{\mathbf{W}^\star}_L^\top \overline{\mathbf{W}^\star}_L \right\|_F \leq \sqrt{\Upsilon_L^3} (\mathsf{poly}_2)^7 \left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1} \right) \quad .$$

[33]We note here, to apply Claim P.10, one also needs to ensure $\varepsilon \leq \frac{\alpha_L}{(\mathsf{poly}_2)^8 \sqrt{\Upsilon_L^2}}$ and $\delta_{L-1} \leq \frac{1}{(\mathsf{poly}_2)^8 \sqrt{\Upsilon_L^2}}$; however, both of them are satisfied under the assumptions $\varepsilon \leq \frac{\alpha_L}{(D_L)^9 \Upsilon_L}$ and $\frac{\alpha_L}{\alpha_{L-1}} \leq \frac{1}{4 \Upsilon_L^3 (D_L)^{16} C_{L-1}}$, and the definition of δ_{L-1} from (L.4).

Note from the definition of $\mathbf{Sum}(F_L(x))$ and $\mathbf{Sum}(G_L^\star(x))$ (see (L.6)) we have

$$\mathbf{Sum}(G_L^\star(x)) = \left\|\overline{\mathbf{W}^\star}_L(S_{L-1}^\star(x) * S_{L-1}^\star(x), \dots)\right\|^2$$
$$\mathbf{Sum}(F_L(x)) = \left\|\overline{\mathbf{W}}_L(S_{L-1}(x) * S_{L-1}(x), \dots)\right\|^2$$

so using a similar derivation as (L.10), we have

$$\mathbb{E}_{x\sim\mathcal{D}} \left(\mathbf{Sum}(F_L(x)) - \mathbf{Sum}(G_L^\star(x))\right)^2 \leq \Upsilon_L^3(\mathsf{poly}_2)^{15}\left(\frac{\varepsilon}{\alpha_L} + \delta_{L-1}\right)^2 \quad . \tag{L.24}$$

L.9 Recursion

We can now put (L.24) back to the bound of $\mathbf{Obj}_{L-1}$ (see (L.1)) and derive that

$$\begin{aligned}\mathbf{Obj}_{L-1} &\leq 2\alpha_L^2 \mathbb{E}_{x\sim\mathcal{D}} \left(\mathbf{Sum}(F_L(x)) - \mathbf{Sum}(G_L^\star(x))\right)^2 + 2\mathbf{Obj} \\ &\leq \Upsilon_L^3(\mathsf{poly}_2)^{16}\left(\delta_{L-1}^2\alpha_L^2 + \varepsilon^2\right) \quad .\end{aligned} \tag{L.25}$$

Note this is a tighter upper bound on $\mathbf{Obj}_{L-1}$ comparing to the previously used one in (L.3). Therefore, we can apply the induction hypothesis again and replace (L.4) also with a tighter bound

$$\forall \ell = 2,3,\dots,L-1: \quad \mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_\ell S_\ell^\star(x) - S_\ell(x)\|_2^2 \leq \left(\frac{\varepsilon + \delta_{L-1}\alpha_L}{\sqrt{\alpha_\ell \alpha_{\ell+1}}}\right)^2 \Upsilon_L^3(\mathsf{poly}_2)^{16} C_{L-1} \quad . \tag{L.26}$$

In other words, we can replace our previous crude bound on δ_{L-1} (see (L.3)) with this tighter bound (L.26), and repeat. By our assumption, $\frac{\alpha_L}{\alpha_{L-1}} \leq \frac{1}{4\Upsilon_L^3(D_L)^{16}C_{L-1}}$, this implies that the process ends when[34]

$$\delta_{L-1}^2 = \left(\frac{\varepsilon}{\sqrt{\alpha_{L-1}\alpha_L}}\right)^2 \cdot 2\Upsilon_L^3(\mathsf{poly}_2)^{16} C_{L-1} \quad . \tag{L.27}$$

Plugging this choice back to (L.26), we have for every $\ell = 2,3,\dots,L-1$

$$\mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_\ell S_\ell^\star(x) - S_\ell(x)\|_2^2 \leq \left(\frac{\varepsilon}{\sqrt{\alpha_\ell\alpha_{\ell+1}}}\right)^2 \cdot 2\Upsilon_L^3(\mathsf{poly}_2)^{16} C_{L-1} \leq \left(\frac{\varepsilon}{\sqrt{\alpha_\ell\alpha_{\ell+1}}}\right)^2 C_L$$

As for the case of $\ell = L$, we derive from (L.22) that

$$\mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_L S_L^\star(x) - S_L(x)\|_2^2 \leq 2\Upsilon_L^2(\mathsf{poly}_2)^{17}\left(\frac{\varepsilon}{\alpha_L}\right)^2 \leq \left(\frac{\varepsilon}{\sqrt{\alpha_L\alpha_{L+1}}}\right)^2 C_L$$

This completes the proof of Theorem L.1. ■

L.10 Proof of Corollary L.3

Proof of Corollary L.3. As for Corollary L.3, we first note that our final choice of δ_{L-1} (see (L.27)), when plugged into (L.13), (L.15), (L.20) and (L.22), respectively give us

$$\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L\triangleleft}\mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F^2 \leq 2(D_L)^2\left(\frac{\varepsilon}{\alpha_L}\right)^2$$
$$\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L\triangleleft}\mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F^2 \leq 2\Upsilon_L(D_L)^4\left(\frac{\varepsilon}{\alpha_L}\right)^2$$
$$\left\|\vec{\mathbf{Q}}_L^\top \overline{\mathbf{K}}_L^\top \overline{\mathbf{K}}_L \vec{\mathbf{Q}}_L - \overline{\mathbf{W}^\star}_L^\top \overline{\mathbf{W}^\star}_L\right\|_F^2 \leq 2\Upsilon_L^2(D_L)^{14}\left(\frac{\varepsilon}{\alpha_L}\right)^2$$
$$\mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_L S_L^\star(x) - S_L(x)\|_2^2 \leq 2\Upsilon_L^2(D_L)^{17}\left(\frac{\varepsilon}{\alpha_L}\right)^2$$

[34]To be precise, we also need to verify that this new $\delta_{L-1} \leq \frac{1}{(\mathsf{poly}_2)^8}$ as before, but this is ensured from our assumptions $\varepsilon \leq \frac{\alpha_L}{(D_L)^9\Upsilon_L}$ and $\frac{\alpha_L}{\alpha_{L-1}} \leq \frac{1}{4\Upsilon_L^3(D_L)^{16}C_{L-1}}$.

So far this has only given us bounds for the L-th layer. As for other layers $\ell = 2, 3, \dots, L-1$, we note that our final choice of δ_{L-1} (see (L.27)), when plugged into the formula of $\mathbf{Obj}_{L-1}$ (see (L.25)), in fact gives

$$\mathbf{Obj}_{L-1} \leq 2\Upsilon_L^3 (D_L)^{16} \varepsilon^2 < \left(2\sqrt{\Upsilon_L^3}(D_L)^8 \varepsilon\right)^2 \ll \left(\frac{\alpha_{L-1}}{(D_{L-1})^9 \Upsilon_{L-1}}\right)^2 .$$

using our assumptions $\varepsilon \leq \frac{\alpha_L}{(D_L)^9 \Upsilon_L}$ and $\frac{\alpha_L}{\alpha_{L-1}} \leq \frac{1}{4\Upsilon_L^3 (D_L)^{16} C_{L-1}}$. Therefore, we can recurse to the case of $L-1$ with ε^2 replaced with $4\Upsilon_L^3 (D_L)^{16} \varepsilon^2$. Continuing in this fashion gives the desired bounds.

Finally, our assumption $\varepsilon \leq \frac{\alpha_L}{(D_L)^9 \Upsilon_L}$ implies $\mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_L S_L^\star(x) - S_L(x)\|_2^2 \leq 1$, and using gap assumption it also holds for previous layers:

$$\forall \ell < L: \qquad \mathbb{E}_{x\sim\mathcal{D}} \|\mathbf{U}_\ell S_\ell^\star(x) - S_\ell(x)\|_2^2 \leq 2\Upsilon_\ell^2 (D_\ell)^{17} \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot \frac{C_L}{C_\ell} \leq 1$$

They also imply $\mathbb{E}_{x\sim\mathcal{D}} \|S_\ell(x)\|_2^2 \leq 2B_\ell$ using $\mathbb{E}_{x\sim\mathcal{D}} \|S_\ell^\star(x)\|_2^2 \leq B_\ell$. □

L.11 Proof of Corollary L.4

Proof of Corollary L.4. This time, we begin by recalling that from (L.3):

$$\mathbf{Obj}_{L-1} \leq \alpha_L^2 \cdot (k_L L \overline{B}_L \tau_L)^8 c_3(2^L) + 2\varepsilon^2 \leq \alpha_L^2 \cdot D_L$$

Therefore, we can use $\varepsilon^2 = \alpha_L^2 \cdot D_L$ and apply Theorem L.1 and Corollary L.3 for the case of $L-1$. This is why we choose $\varepsilon_0 = \alpha_L \cdot \sqrt{D_L}$ for $\ell < L$.

As for the case of $\ell = L$, we first note the $L-1$ case tells us

$$\mathbb{E}_{x\sim\mathcal{D}} \left\|\mathbf{U}_{L-1} S_{L-1}^\star(x) - S_{L-1}(x)\right\|_2^2 \leq \delta_{L-1}^2 := 6\Upsilon_{L-1}^2 (D_{L-1})^{17} \left(\frac{\varepsilon}{\alpha_{L-1}}\right)^2 \ll \left(\frac{\varepsilon}{\alpha_L}\right)^2$$

Therefore, we can plug in this choice of δ_{L-1} into (L.13), (L.15) and (L.20) to derive

$$\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{W}}_{L,L-1}^\top \overline{\mathbf{W}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F^2 \leq 2(D_L)^2 \left(\frac{\varepsilon}{\alpha_L}\right)^2$$

$$\left\|\mathbf{Q}_{L-1}^\top \overline{\mathbf{K}}_{L,L-1}^\top \overline{\mathbf{K}}_{L\triangleleft} \mathbf{Q}_{L\triangleleft} - \overline{\mathbf{W}^\star}_{L,L-1}^\top \overline{\mathbf{W}^\star}_{L\triangleleft}\right\|_F^2 \leq 2\Upsilon_L (D_L)^4 \left(\frac{\varepsilon}{\alpha_L}\right)^2$$

$$\left\|\vec{\mathbf{Q}}_L^\top \overline{\mathbf{K}}_L^\top \overline{\mathbf{K}}_L \vec{\mathbf{Q}}_L - \overline{\mathbf{W}^\star}_L^\top \overline{\mathbf{W}^\star}_L\right\|_F^2 \leq 2\Upsilon_L^2 (D_L)^{14} \left(\frac{\varepsilon}{\alpha_L}\right)^2$$

Note that the three equations (L.13), (L.15) and (L.20) have only required the weaker requirement $\varepsilon \leq \frac{\alpha_L}{(D_L)^3 \sqrt{\Upsilon_L}}$ on ε comparing to the full Theorem L.1 (the stronger requirement was $\varepsilon \leq \frac{\alpha_L}{(D_L)^9 \Upsilon_L}$, but it is required only starting from equation (L.21)). □

M Construction of Descent Direction

Let $\mathbf{U}_\ell$ be defined as in Theorem L.1. Let us construct $\mathbf{V}_{\ell,j}^\star \in \mathbb{R}^{k_\ell \times \binom{k_j+1}{2}}$ or $\mathbb{R}^{k_\ell \times d}$ that satisfies

$$\forall j > 2: \mathbf{V}_{\ell,j}^\star \sigma(\mathbf{R}_j \mathbf{U}_j z) = \mathbf{W}_{\ell,j}^\star \sigma(z), \quad \forall j' \in [2], \mathbf{V}_{\ell,j'}^\star = \mathbf{W}_{\ell,j'}^\star \tag{M.1}$$

and the singular values of $\mathbf{V}_{\ell,j}^\star$ are between $[\frac{1}{O(k_\ell^4 L^2 \kappa)}, O(L^2\kappa)]$. (This can be done by defining $\mathbf{V}_{\ell,j}^\star = \mathbf{W}_{\ell,j}^\star (\mathbf{I} * \mathbf{I})(\mathbf{R}_j \mathbf{U}_j * \mathbf{R}_j \mathbf{U}_j)^{-1} \in \mathbb{R}^{k_\ell \times \binom{k_j+1}{2}}$, and the singular value bounds are due to Fact I.7, Lemma I.5 and Lemma I.6.) Let us also introduce notations

$$\mathbf{E}_\ell := \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_\ell - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_\ell^\star = (\mathbf{E}_{\ell,\ell-1}, \mathbf{E}_{\ell\triangleleft})$$

$$\mathbf{E}_{\ell\triangleleft} := \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell\triangleleft}^\star$$

$$\mathbf{E}_{\ell,\ell-1} := \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star$$
$$\widehat{\mathbf{E}}_\ell := \mathbf{K}_\ell^\top \mathbf{K}_\ell - \left(\mathbf{V}_\ell^\star\right)^\top \mathbf{V}_\ell^\star$$

Let us consider updates (for some $\eta_2 \geq \eta_1$):

$$\mathbf{W}_\ell \leftarrow \sqrt{1-\eta_1}\mathbf{W}_\ell + \sqrt{\eta_1}\mathbf{D}_\ell \mathbf{V}_\ell^{\star,w}$$
$$\mathbf{K}_{\ell\lhd} \leftarrow \left(1+\frac{\eta_1}{2}\right)\mathbf{K}_{\ell\lhd} - \eta_1 \mathbf{Q}_\ell \mathbf{K}_{\ell\lhd} - \eta_2 \mathbf{K}_{\ell,\ell-1}\mathbf{E}_{\ell\lhd}$$
$$\mathbf{K}_{\ell,\ell-1} \leftarrow \left(1-\frac{\eta_1}{2}\right)\mathbf{K}_{\ell,\ell-1} + \eta_1 \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1} - \eta_2 \mathbf{K}_{\ell\lhd}\mathbf{E}_{\ell\lhd}^\top$$

where $\mathbf{V}_\ell^{\star,w} \in \mathbb{R}^{m\times *}$ is defined as $(\mathbf{V}_\ell^{\star,w})^\top = \frac{\sqrt{k_\ell}}{\sqrt{m}}((\mathbf{V}_\ell^\star)^\top, \dots (\mathbf{V}_\ell^\star)^\top)$ which contains $\frac{m}{k_\ell}$ identical copies of $\mathbf{V}_\ell^\star$, and $\mathbf{D}_\ell \in \mathbb{R}^{m\times m}$ is a diagonal matrix with diagonals as random ± 1, and $\mathbf{Q}_\ell$ is a symmetric matrix given by

$$\mathbf{Q}_\ell = \frac{1}{2}\left(\mathbf{K}_{\ell,\ell-1}\mathbf{K}_{\ell,\ell-1}^\top\right)^{-1}\mathbf{K}_{\ell,\ell-1}\left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star \mathbf{K}_{\ell,\ell-1}^\top \left(\mathbf{K}_{\ell,\ell-1}\mathbf{K}_{\ell,\ell-1}^\top\right)^{-1}$$

M.1 Simple Properties

Fact M.1. *Suppose we know $\|\mathbf{W}_\ell\|_F \leq \widetilde{\kappa}_\ell$. Then,*

$$(\mathbf{W}_\ell^{(\mathsf{new})})^\top(\mathbf{W}_\ell^{(\mathsf{new})}) = (1-\eta_1)(\mathbf{W}_\ell)^\top \mathbf{W}_\ell + \eta_1(\mathbf{V}_\ell^\star)^\top \mathbf{V}_\ell^\star + \sqrt{\eta_1}\xi$$

for some error matrix ξ with

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell}[\xi] = \mathbf{0} \quad \textit{and} \quad \Pr_{\mathbf{D}_\ell}\left[\|\xi\|_F > \frac{\log\delta^{-1}\cdot \mathsf{poly}(\widetilde{\kappa}_\ell)}{\sqrt{m}}\right] \leq \delta \quad \textit{and} \quad \mathop{\mathbb{E}}_{\mathbf{D}_\ell}[\|\xi\|_F^2] \leq \frac{\mathsf{poly}(\widetilde{\kappa}_\ell)}{m}$$

Proof. Trivial from vector version of Hoeffding's inequality. □

Claim M.2. *Suppose $\sigma_{\min}(\mathbf{K}_{\ell,\ell-1}), \sigma_{\min}(\mathbf{K}_{\ell\lhd}) \geq \frac{1}{2\widetilde{\kappa}}$ and $\|\mathbf{K}_\ell\|_2 \leq 2\widetilde{\kappa}$ for some $\widetilde{\kappa} \geq \kappa + k_\ell + L$, we have:*

$$\langle \mathbf{E}_{\ell\lhd}, \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1}\mathbf{E}_{\ell\lhd} + \mathbf{E}_{\ell\lhd}\mathbf{K}_{\ell\lhd}^\top \mathbf{K}_{\ell\lhd}\rangle \geq \frac{1}{\mathsf{poly}(\widetilde{\kappa})}\|\mathbf{E}_{\ell\lhd}\|_F^2$$

Proof of Claim M.2. We first note the left hand side

$$LHS = \|\mathbf{K}_{\ell,\ell-1}\mathbf{E}_{\ell\lhd}\|_F^2 + \|\mathbf{K}_{\ell\lhd}\mathbf{E}_{\ell\lhd}^\top\|_F^2$$

Without loss of generality (by left/right multiplying with a unitary matrix), let us write $\mathbf{K}_{\ell,\ell-1} = (\mathbf{K}_1, \mathbf{0})$ and $\mathbf{K}_{\ell\lhd} = (\mathbf{K}_2, \mathbf{0})$ for square matrices $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{R}^{k_\ell\times k_\ell}$. Accordingly, let us write $\mathbf{E}_{\ell\lhd} = \left(\begin{smallmatrix}\mathbf{E}_1 & \mathbf{E}_2\\ \mathbf{E}_3 & \mathbf{E}_4\end{smallmatrix}\right)$ for $\mathbf{E}_1 \in \mathbb{R}^{k_\ell\times k_\ell}$. We have

$$LHS = \|(\mathbf{K}_1\mathbf{E}_1, \mathbf{K}_1\mathbf{E}_2)\|_F^2 + \|(\mathbf{K}_2\mathbf{E}_1^\top, \mathbf{K}_2\mathbf{E}_3^\top)\|_F^2 \geq \frac{1}{\mathsf{poly}(\widetilde{\kappa})}(\|\mathbf{E}_1\|_F^2 + \|\mathbf{E}_2\|_F^2 + \|\mathbf{E}_3\|_F^2) \ .$$

Note also $\|\mathbf{E}_{\ell\lhd}\|_F \leq \mathsf{poly}(\widetilde{\kappa})$. Let us write $\mathbf{V}_{\ell,\ell-1}^\star = (\mathbf{V}_1, \mathbf{V}_2)$ and $\mathbf{V}_{\ell\lhd}^\star = (\mathbf{V}_3, \mathbf{V}_4)$ for square matrices $\mathbf{V}_1, \mathbf{V}_3 \in \mathbb{R}^{k_\ell\times k_\ell}$. Then we have

$$\mathbf{E}_{\ell\lhd} = \begin{pmatrix}\mathbf{E}_1 & \mathbf{E}_2\\ \mathbf{E}_3 & \mathbf{E}_4\end{pmatrix} = \begin{pmatrix}\mathbf{K}_1^\top\mathbf{K}_2 - \mathbf{V}_1^\top\mathbf{V}_3 & -\mathbf{V}_1^\top\mathbf{V}_4\\ -\mathbf{V}_2^\top\mathbf{V}_3 & -\mathbf{V}_2^\top\mathbf{V}_4\end{pmatrix} \tag{M.2}$$

Recall we have $\|\mathbf{V}_{\ell,\ell-1}^\star\|_2, \|\mathbf{V}_{\ell\lhd}^\star\|_2 \leq L^2\kappa$. Consider two cases.

In the first case, $\sigma_{\min}(\mathbf{V}_1) \leq \frac{1}{16L^2\kappa(\widetilde{\kappa})^2}$. Then, it satisfies $\|\mathbf{E}_1\|_F \geq \frac{1}{2}\|\mathbf{K}_1^\top\mathbf{K}_2\|_F \geq \frac{1}{8(\widetilde{\kappa})^2}$ so we are done. In the second case, $\sigma_{\min}(\mathbf{V}_1) \geq \frac{1}{16L^2\kappa(\widetilde{\kappa})^2}$. We have

$$\|\mathbf{E}_2\|_F = \|\mathbf{V}_1^\top\mathbf{V}_4\|_F \geq \sigma_{\min}(\mathbf{V}_1)\|\mathbf{V}_4\|_F \geq \frac{\sigma_{\min}(\mathbf{V}_1)}{\sigma_{\max}(\mathbf{V}_2)}\|\mathbf{V}_2^\top\mathbf{V}_4\|_F \geq \frac{1}{\mathsf{poly}(\widetilde{\kappa})}\|\mathbf{E}_4\|_F$$

so we are also done. □

Claim M.3. *Suppose $\sigma_{\min}(\mathbf{K}_{\ell,\ell-1}) \geq \frac{1}{\widetilde{\kappa}}$ and $\|\mathbf{K}_\ell\|_2 \leq \widetilde{\kappa}$ for some $\widetilde{\kappa} \geq \kappa + k_\ell + L$, we have*

$$\left\|2\mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star\right\|_F \leq \mathsf{poly}(\widetilde{\kappa}) \left\|\mathbf{E}_{\ell\triangleleft}\right\|_F$$
$$\textit{and} \quad \|2\mathbf{Q}_\ell - \mathbf{I}\|_F \leq (\widetilde{\kappa})^2 \|\mathbf{E}_{\ell,\ell-1}\|_F$$

Proof of Claim M.3. Without loss of generality (by applying a unitary transformation), let us write $\mathbf{K}_{\ell,\ell-1} = (\mathbf{K}, \mathbf{0})$ for square matrix $\mathbf{K} \in \mathbb{R}^{k_\ell \times k_\ell}$, and let us write $\mathbf{V}_{\ell,\ell-1}^\star = (\mathbf{V}_1, \mathbf{V}_2)$ for square matrix $\mathbf{V}_1 \in \mathbb{R}^{k_\ell \times k_\ell}$. From (M.2), we have

$$\|\mathbf{V}_2\|_F \leq \frac{\|\mathbf{E}_{\ell\triangleleft}\|_F}{\sigma_{\min}(\mathbf{V}_{\ell\triangleleft}^\star)} \leq \mathsf{poly}(k_\ell, \kappa, L) \cdot \|\mathbf{E}_{\ell\triangleleft}\|_F \ .$$

From the definition of $\mathbf{Q}_\ell$ we have

$$2\mathbf{Q}_\ell = (\mathbf{K}\mathbf{K}^\top)^{-1}(\mathbf{K}, \mathbf{0})\left(\mathbf{V}_1, \mathbf{V}_2\right)^\top \left(\mathbf{V}_1, \mathbf{V}_2\right)\left(\mathbf{K}, \mathbf{0}\right)^\top (\mathbf{K}\mathbf{K}^\top)^{-1} = \mathbf{K}^{-\top}\mathbf{V}_1^\top \mathbf{V}_1 \mathbf{K}^{-1} \tag{M.3}$$

It is easy to verify that

$$2\mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star = \begin{pmatrix} \mathbf{V}_1^\top \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star = \begin{pmatrix} \mathbf{0} & \mathbf{V}_1^\top \mathbf{V}_2 \\ \mathbf{V}_2^\top \mathbf{V}_1 & \mathbf{V}_2^\top \mathbf{V}_2 \end{pmatrix}$$

which shows that

$$\left\|2\mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star\right\|_F \leq 2\|\mathbf{V}_1\|_F \|\mathbf{V}_2\|_F + \|\mathbf{V}_2\|_F^2 \leq \mathsf{poly}(\widetilde{\kappa}) \cdot \|\mathbf{E}_{\ell\triangleleft}\|_F \ .$$

Next, we consider $\|2\mathbf{Q}_\ell - \mathbf{I}\|_F^2$, since

$$\|\mathbf{K}^\top \mathbf{K} - \mathbf{V}_1^\top \mathbf{V}_1\|_F \leq \left\|\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell,\ell-1}^\star\right\|_F = \|\mathbf{E}_{\ell,\ell-1}\|_F \ ,$$

we immediately have

$$\|2\mathbf{Q}_\ell - \mathbf{I}\|_F \leq \frac{1}{\sigma_{\min}(\mathbf{K})^2}\|\mathbf{K}^\top \mathbf{K} - \mathbf{V}_1^\top \mathbf{V}_1\|_F \leq (\widetilde{\kappa})^2 \|\mathbf{E}_{\ell,\ell-1}\|_F \ .$$

□

M.2 Frobenius Norm Updates

Consider the F-norm regularizers given by

$$\begin{aligned}
\mathbf{R}_{6,\ell} &= \|\mathbf{K}_\ell\|_F^2 = \mathbf{Tr}(\mathbf{K}_\ell^\top \mathbf{K}_\ell) \\
&= \mathbf{Tr}(\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1}) + 2\mathbf{Tr}(\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft}) + \mathbf{Tr}(\mathbf{K}_{\ell\triangleleft}^\top \mathbf{K}_{\ell\triangleleft}) \\
\mathbf{R}_{7,\ell} &= \|\mathbf{W}_\ell\|_F^2 = \mathbf{Tr}(\mathbf{W}_\ell^\top \mathbf{W}_\ell)
\end{aligned}$$

Lemma M.4. *Suppose for some parameter $\widetilde{\kappa}_\ell \geq \kappa + L + k_\ell$ it satisfies*

$$\sigma_{\min}(\mathbf{K}_{\ell,\ell-1}) \geq \frac{1}{2\widetilde{\kappa}_\ell} \textit{ and } \|\mathbf{K}_\ell\|_2 \leq 2\widetilde{\kappa}_\ell,\ \eta_1, \eta_2 < \frac{1}{\mathsf{poly}(\widetilde{\kappa}_\ell)}, \textit{ and } \|\mathbf{E}_{\ell\triangleleft}\|_F \leq \frac{1}{(2\widetilde{\kappa}_\ell)^2}$$

then

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{D}_\ell}\left[\mathbf{R}_{7,\ell}^{(\mathsf{new})}\right] &\leq (1-\eta_1)\mathbf{R}_{7,\ell} + \eta_1 \cdot \mathsf{poly}(k_\ell, L, \kappa) \\
\mathbf{R}_{6,\ell}^{(\mathsf{new})} &\leq (1-\eta_1)\mathbf{R}_{6,\ell} + \eta_1 \cdot \mathsf{poly}(k_\ell, \kappa, L) + (\eta_1^2 + \eta_2\|\mathbf{E}_{\ell\triangleleft}\|_F) \cdot \mathsf{poly}(\widetilde{\kappa}_\ell)
\end{aligned}$$

Proof of Lemma M.4. Our updates satisfy

$$\begin{aligned}
\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} &\leftarrow (1-\eta_1)\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} + 2\eta_1 \mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1} + \xi_1 \\
\mathbf{K}_{\ell\triangleleft}^\top \mathbf{K}_{\ell\triangleleft} &\leftarrow (1+\eta_1)\mathbf{K}_{\ell\triangleleft}^\top \mathbf{K}_{\ell\triangleleft} - 2\eta_1 \mathbf{K}_{\ell\triangleleft}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell\triangleleft} + \xi_2 \\
\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft} &\leftarrow \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft} + \xi_3 \\
\mathbf{W}_\ell^\top \mathbf{W}_\ell &\leftarrow (1-\eta_1)(\mathbf{W}_\ell)^\top \mathbf{W}_\ell + \eta_1 (\mathbf{V}_\ell^\star)^\top \mathbf{V}_\ell^\star + \sqrt{\eta_1}\xi_4
\end{aligned}$$

where error matrices $\|\xi_1\|_F, \|\xi_2\|_F, \|\xi_3\|_F \leq (\eta_1^2 + \eta_2\|\mathbf{E}_{\ell\triangleleft}\|_F) \cdot \mathsf{poly}(\widetilde{\kappa}_\ell)$ and $\mathbb{E}_{\mathbf{D}_\ell}[\xi_4] = 0$.

The $\mathbf{R}_{7,\ell}$ part is now trivial and the $\mathbf{R}_{6,\ell}$ part is a direct corollary of Claim M.5. □

Claim M.5. *The following is always true*

$$\mathbf{Tr}\left(-\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} + 2\mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1}\right) \leq -\|\mathbf{K}_{\ell,\ell-1}\|_F^2 + O\left(k_\ell^2 \kappa^2\right)$$

Furthermore, suppose $\sigma_{\min}(\mathbf{K}_{\ell,\ell-1}) \geq \frac{1}{2\widetilde{\kappa}_\ell}$ and $\|\mathbf{K}_\ell\|_2 \leq 2\widetilde{\kappa}_\ell$ for $\widetilde{\kappa}_\ell \geq \kappa + L + k_\ell$, we have that as long as $\|\mathbf{E}_{\ell\triangleleft}\|_F \leq \frac{1}{(2\widetilde{\kappa}_\ell)^2}$ then

$$\mathbf{Tr}\left(\mathbf{K}_{\ell\triangleleft}^\top \mathbf{K}_{\ell\triangleleft} - 2\mathbf{K}_{\ell\triangleleft}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell\triangleleft}\right) \leq -\|\mathbf{K}_{\ell\triangleleft}\|_F^2 + O((L^2\kappa)^2 k_\ell)$$

Proof of Claim M.5. For the first bound, it is a direct corollary of the bound $\|2\mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1}\|_F \leq \mathsf{poly}(\kappa, L)$ (which can be easily verified from formulation (M.3)).

As for the second bound, let us assume without loss of generality (by left/right multiplying with a unitary matrix) that $\mathbf{K}_{\ell,\ell-1} = (\mathbf{K}_1, \mathbf{0})$ and $\mathbf{K}_{\ell\triangleleft} = (\mathbf{K}_2, \mathbf{0})$ for square matrices $\mathbf{K}_1, \mathbf{K}_2 \in \mathbb{R}^{k_\ell \times k_\ell}$. Let us write $\mathbf{V}_{\ell,\ell-1}^\star = (\mathbf{V}_1, \mathbf{V}_2)$ and $\mathbf{V}_{\ell\triangleleft}^\star = (\mathbf{V}_3, \mathbf{V}_4)$ for square matrices $\mathbf{V}_1, \mathbf{V}_3 \in \mathbb{R}^{k_\ell \times k_\ell}$. Then we have,

$$\mathbf{E}_{\ell\triangleleft} = \begin{pmatrix} \mathbf{E}_1 & \mathbf{E}_2 \\ \mathbf{E}_3 & \mathbf{E}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{K}_1^\top \mathbf{K}_2 - \mathbf{V}_1^\top \mathbf{V}_3 & -\mathbf{V}_1^\top \mathbf{V}_4 \\ -\mathbf{V}_2^\top \mathbf{V}_3 & -\mathbf{V}_2^\top \mathbf{V}_4 \end{pmatrix}$$

We have

$$\begin{aligned} \|\mathbf{K}_1^\top \mathbf{K}_2 - \mathbf{V}_1^\top \mathbf{V}_3\|_F \leq \|\mathbf{E}_{\ell\triangleleft}\|_F &\implies \|\mathbf{K}_2 - \mathbf{K}_1^{-\top} \mathbf{V}_1^\top \mathbf{V}_3\|_F \leq 2\widetilde{\kappa}_\ell \cdot \|\mathbf{E}_{\ell\triangleleft}\|_F \ . \\ &\implies \left\|\mathbf{K}_2 \mathbf{K}_2^\top - \mathbf{K}_1^{-\top} \mathbf{V}_1^\top \mathbf{V}_3 \mathbf{V}_3^\top \mathbf{V}_1 \mathbf{K}_1^{-1}\right\|_F \leq (2\widetilde{\kappa}_\ell)^2 \cdot \|\mathbf{E}_{\ell\triangleleft}\|_F \end{aligned}$$

Translating this into the spectral dominance formula (recalling $\mathbf{A} \succeq \mathbf{B}$ means $\mathbf{A} - \mathbf{B}$ is positive semi-definite), we have

$$\begin{aligned} \mathbf{K}_2 \mathbf{K}_2^\top &\preceq \mathbf{K}_1^{-\top} \mathbf{V}_1^\top \mathbf{V}_3 \mathbf{V}_3^\top \mathbf{V}_1 \mathbf{K}_1^{-1} + (2\widetilde{\kappa}_\ell)^2 \cdot \|\mathbf{E}_{\ell\triangleleft}\|_F \cdot \mathbf{I} \\ &\preceq (L^2\kappa)^2 \cdot \mathbf{K}_1^{-\top} \mathbf{V}_1^\top \mathbf{V}_1 \mathbf{K}_1^{-1} + (2\widetilde{\kappa}_\ell)^2 \cdot \|\mathbf{E}_{\ell\triangleleft}\|_F \cdot \mathbf{I} \qquad (\text{using } \|\mathbf{V}_{\ell\triangleleft}^\star\|_2 \leq L^2\kappa) \end{aligned}$$

On the other hand, from (M.3) one can verify that

$$2\mathbf{K}_{\ell\triangleleft}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell\triangleleft} = \mathbf{K}_2^\top \mathbf{K}_1^{-\top} \mathbf{V}_1^\top \mathbf{V}_1 \mathbf{K}_1^{-1} \mathbf{K}_2$$

Combining the two formula above, we have

$$\begin{aligned} 2\mathbf{K}_{\ell\triangleleft}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell\triangleleft} &\succeq \frac{1}{(L^2\kappa)^2} \mathbf{K}_2^\top \mathbf{K}_2 \mathbf{K}_2^\top \mathbf{K}_2 - (2\widetilde{\kappa}_\ell)^2 \|\mathbf{E}_{\ell\triangleleft}\|_F \cdot \mathbf{K}_2^\top \mathbf{K}_2 \\ &\succeq 2\mathbf{K}_2^\top \mathbf{K}_2 - O((L^2\kappa)^2) \cdot \mathbf{I} \qquad (\text{using } \mathbf{A}^2 \succeq 2\mathbf{A} - \mathbf{I} \text{ for symmetric } \mathbf{A}) \end{aligned}$$

Taking trace on both sides finish the proof. □

M.3 Regularizer Updates

Let us consider three regularizer

$$\begin{aligned} \mathbf{R}_{3,\ell} &= \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\triangleleft} - \mathbf{W}_{\ell,\ell-1}^\top \mathbf{W}_{\ell\triangleleft} \\ \mathbf{R}_{4,\ell} &= \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} - \mathbf{W}_{\ell,\ell-1}^\top \mathbf{W}_{\ell,\ell-1} \\ \mathbf{R}_{5,\ell} &= \mathbf{K}_\ell^\top \mathbf{K}_\ell - \mathbf{W}_\ell^\top \mathbf{W}_\ell \end{aligned}$$

Lemma M.6. *Suppose for some parameter $\widetilde{\kappa} \geq \kappa + L + k_\ell$ it satisfies*

$$\sigma_{\min}(\mathbf{K}_{\ell,\ell-1}) \geq \frac{1}{2\widetilde{\kappa}},\ \sigma_{\min}(\mathbf{K}_{\ell\triangleleft}) \geq \frac{1}{2\widetilde{\kappa}},\ \|\mathbf{K}_\ell\|_2, \|\mathbf{W}_\ell\|_2 \leq 2\widetilde{\kappa},\ \eta_2 < \frac{1}{\mathsf{poly}(\widetilde{\kappa})},\ \eta_1 \leq \frac{\eta_2}{\mathsf{poly}(\widetilde{\kappa})}$$

then, suppose $\mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2$ *and suppose Corollary L.3 holds for* $L \geq \ell$*, then*

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{3,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{3,\ell}\|_F^2 + \eta_1^3 \cdot \mathsf{poly}(\widetilde{\kappa}) + (\eta_2 \frac{\varepsilon^2}{\alpha_\ell^2}) \cdot (D_\ell)^4 \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$
$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{4,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{4,\ell}\|_F^2 + \eta_2 \frac{\varepsilon^2}{\alpha_\ell^2} \Upsilon_\ell \cdot (D_\ell)^6 \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$
$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{5,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{5,\ell}\|_F^2 + \eta_2 \frac{\varepsilon^2}{\alpha_\ell^2} \Upsilon_\ell^2 \cdot (D_\ell)^{16} \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

Proof of Lemma M.6. Let us check how these matrices get updated.

$$\mathbf{R}_{3,\ell} \leftarrow (1-\eta_1)\mathbf{R}_{3,\ell} + \eta_1 \mathbf{E}_{\ell\lhd} - \eta_2 \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} \mathbf{E}_{\ell\lhd} - \eta_2 \mathbf{E}_{\ell\lhd} \mathbf{K}_{\ell\lhd}^\top \mathbf{K}_{\ell\lhd} + \xi_3 + \zeta_3$$
$$(\text{using } \mathbf{E}_{\ell\lhd} = \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\lhd} - (\mathbf{V}^\star{}_{\ell,\ell-1})^\top \mathbf{V}^\star{}_{\ell\lhd})$$
$$\mathbf{R}_{4,\ell} \leftarrow (1-\eta_1)\mathbf{R}_{4,\ell} + \eta_1 \left(2\mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1} - (\mathbf{V}^\star{}_{\ell,\ell-1})^\top \mathbf{V}^\star{}_{\ell,\ell-1} \right) + \xi_4 + \zeta_4$$
$$\mathbf{R}_{5,\ell} \leftarrow (1-\eta_1)\mathbf{R}_{5,\ell} + \eta_1 \left(\mathbf{K}_\ell^\top \mathbf{K}_\ell - (\mathbf{V}_\ell^\star)^\top \mathbf{V}_\ell^\star \right) - \eta_1 \mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell,\ell-1} + 2\eta_1 \mathbf{K}_{\ell,\ell-1}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell,\ell-1}$$
$$+ \eta_1 \mathbf{K}_{\ell\lhd}^\top \mathbf{K}_{\ell\lhd} - 2\eta_1 \mathbf{K}_{\ell\lhd}^\top \mathbf{Q}_\ell \mathbf{K}_{\ell\lhd} + \xi_5 + \zeta_5$$

where error matrices $\mathbb{E}_{\mathbf{D}_\ell}[\zeta_3] = 0, \mathbb{E}_{\mathbf{D}_\ell}[\zeta_4] = 0, \mathbb{E}_{\mathbf{D}_\ell}[\zeta_5] = 0$ and

$$\|\xi_3\|_F \leq (\eta_1^2 + \eta_2^2 \|\mathbf{E}_{\ell\lhd}\|_F^2) \cdot \mathsf{poly}(\widetilde{\kappa})$$
$$\|\xi_4\|_F, \|\xi_5\|_F \leq (\eta_1^2 + \eta_2 \|\mathbf{E}_{\ell\lhd}\|_F) \cdot \mathsf{poly}(\widetilde{\kappa})$$
$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \|\zeta_3\|_F^2, \mathop{\mathbb{E}}_{\mathbf{D}_\ell} \|\zeta_4\|_F^2, \mathop{\mathbb{E}}_{\mathbf{D}_\ell} \|\zeta_5\|_F^2 \leq \frac{\eta_1}{m} \cdot \mathsf{poly}(\widetilde{\kappa})$$

The update on $\mathbf{R}_{3,\ell}$ now tells us (by applying Claim M.2)

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{3,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 2\eta_1) \|\mathbf{R}_{3,\ell}\|_F^2 + 2\eta_1 \|\mathbf{R}_{3,\ell}\|_F \|\mathbf{E}_{\ell\lhd}\|_F - \frac{\eta_2}{\mathsf{poly}(\widetilde{\kappa})} \|\mathbf{E}_{\ell\lhd}\|_F^2$$
$$+ \eta_2 \mathsf{poly}(\widetilde{\kappa}) \left\| \mathbf{W}_\ell^\top \mathbf{W}_{\ell,\ell-1} - (\mathbf{V}_\ell^\star)^\top \mathbf{V}_{\ell,\ell-1}^\star \right\|_F \|\mathbf{E}_{\ell\lhd}\|_F$$
$$+ (\eta_1^2 \|\mathbf{R}_{3,\ell}\|_F + \eta_1^2 \|\mathbf{E}_{\ell\lhd}\|_F + \eta_2^2 \|\mathbf{E}_{\ell\lhd}\|_F^2 + \frac{\eta_1}{m}) \cdot \mathsf{poly}(\widetilde{\kappa})$$

As for $\mathbf{R}_{4,\ell}$ and $\mathbf{R}_{5,\ell}$, applying Claim M.3 and using the notation $\widehat{\mathbf{E}}_\ell = \mathbf{K}_\ell^\top \mathbf{K}_\ell - (\mathbf{V}_\ell^\star)^\top \mathbf{V}_\ell^\star$, we can further simplify them to

$$\mathbf{R}_{4,\ell} \leftarrow (1-\eta_1)\mathbf{R}_{4,\ell} + \xi_4' + \zeta_4 \qquad \text{for } \|\xi_4'\|_F \leq (\eta_1 \|\mathbf{E}_{\ell\lhd}\|_F + \eta_2 \|\mathbf{E}_{\ell\lhd}\|_F) \cdot \mathsf{poly}(\widetilde{\kappa})$$
$$\mathbf{R}_{5,\ell} \leftarrow (1-\eta_1)\mathbf{R}_{5,\ell} + \eta_1 \widehat{\mathbf{E}}_\ell + \xi_5' + \zeta_5 \qquad \text{for } \|\xi_5'\|_F \leq (\eta_1 \|\mathbf{E}_\ell\|_F + \eta_2 \|\mathbf{E}_{\ell\lhd}\|_F) \cdot \mathsf{poly}(\widetilde{\kappa})$$

As a result,

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{4,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.9\eta_1) \|\mathbf{R}_{4,\ell}\|_F^2 + \|\mathbf{R}_{4,\ell}\|_F \cdot (\eta_1 \|\mathbf{E}_{\ell\lhd}\|_F + \eta_2 \|\mathbf{E}_{\ell\lhd}\|_F + \frac{\eta_1}{m}) \cdot \mathsf{poly}(\widetilde{\kappa})$$
$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{5,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.9\eta_1) \|\mathbf{R}_{5,\ell}\|_F^2 + \|\mathbf{R}_{5,\ell}\|_F \cdot (\eta_1 \|\widehat{\mathbf{E}}_\ell\|_F + \eta_2 \|\mathbf{E}_{\ell\lhd}\|_F + \frac{\eta_1}{m}) \cdot \mathsf{poly}(\widetilde{\kappa})$$

Since $\mathbf{Obj} = \varepsilon^2$, by applying Corollary L.3, we have

$$\text{Corollary L.3a :} \qquad \|\mathbf{W}_\ell^\top \mathbf{W}_{\ell,\ell-1} - (\mathbf{V}_\ell^\star)^\top \mathbf{V}_{\ell,\ell-1}^\star\|_F^2 \leq \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot (D_\ell)^3 \cdot \frac{C_L}{C_\ell}$$
$$\text{Corollary L.3b :} \qquad \|\mathbf{E}_{\ell\lhd}\|_F^2 = \|\mathbf{K}_{\ell,\ell-1}^\top \mathbf{K}_{\ell\lhd} - \left(\mathbf{V}_{\ell,\ell-1}^\star\right)^\top \mathbf{V}_{\ell\lhd}^\star\|_F^2 \leq \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot (D_\ell)^5 \Upsilon_\ell \cdot \frac{C_L}{C_\ell}$$
$$\text{Corollary L.3c :} \qquad \|\widehat{\mathbf{E}}_\ell\|_F^2 = \|\mathbf{K}_\ell^\top \mathbf{K}_\ell - (\mathbf{V}_\ell^\star)^\top \mathbf{V}_\ell^\star\|_F^2 \leq \left(\frac{\varepsilon}{\alpha_\ell}\right)^2 \cdot (D_\ell)^{15} \Upsilon_\ell^2 \cdot \frac{C_L}{C_\ell} \tag{M.4}$$

Plugging these into the bounds above, and using $\eta_2 \geq \eta_1 \cdot \mathsf{poly}(\widetilde{\kappa})$ and $\eta_2 \leq \frac{1}{\mathsf{poly}(\widetilde{\kappa})}$, and repeatedly using $2ab \leq a^2 + b^2$, we have

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{3,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{3,\ell}\|_F^2 + \eta_1^3 \cdot \mathsf{poly}(\widetilde{\kappa}) + (\eta_2 \frac{\varepsilon^2}{\alpha_\ell^2}) \cdot (D_\ell)^4 \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{4,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{4,\ell}\|_F^2 + \eta_2 \frac{\varepsilon^2}{\alpha_\ell^2} \Upsilon_\ell \cdot (D_\ell)^6 \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{5,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{5,\ell}\|_F^2 + (\eta_1 \frac{\varepsilon^2}{\alpha_\ell^2} \Upsilon_\ell^2 + \eta_2 \frac{\varepsilon^2}{\alpha_\ell^2} \Upsilon_\ell) \cdot (D_\ell)^{16} \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

□

Lemma M.7. *In the same setting as Lemma M.6, suppose the weaker Corollary L.4 holds for $L \geq \ell$ instead of Corollary L.3. Then, for every $\ell < L$,*

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{3,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{3,\ell}\|_F^2 + \eta_1^3 \cdot \mathsf{poly}(\widetilde{\kappa}) + (\eta_2 \frac{\alpha_L^2 D_L}{\alpha_\ell^2}) \cdot (D_\ell)^4 \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{4,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{4,\ell}\|_F^2 + \eta_2 \frac{\alpha_L^2 D_L}{\alpha_\ell^2} \Upsilon_\ell \cdot (D_\ell)^6 \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_\ell} \left\| \mathbf{R}_{5,\ell}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{5,\ell}\|_F^2 + \eta_2 \frac{\alpha_L^2 D_L}{\alpha_\ell^2} \Upsilon_\ell^2 \cdot (D_\ell)^{16} \cdot \frac{C_L}{C_\ell} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_L} \left\| \mathbf{R}_{3,L}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{3,L}\|_F^2 + \eta_1^3 \cdot \mathsf{poly}(\widetilde{\kappa}) + (\eta_2 \frac{\varepsilon^2}{\alpha_L^2}) \cdot (D_L)^4 + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_L} \left\| \mathbf{R}_{4,L}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{4,L}\|_F^2 + \eta_2 \frac{\varepsilon^2}{\alpha_L^2} \Upsilon_L \cdot (D_L)^6 + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

$$\mathop{\mathbb{E}}_{\mathbf{D}_L} \left\| \mathbf{R}_{5,L}^{(\mathsf{new})} \right\|_F^2 \leq (1 - 1.8\eta_1) \|\mathbf{R}_{5,L}\|_F^2 + \eta_2 \frac{\varepsilon^2}{\alpha_L^2} \Upsilon_L^2 \cdot (D_L)^{16} + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa})}{m}$$

Proof. Proof is identical to Lemma M.6 but replacing the use of Corollary L.3 with Corollary L.4. □

M.4 Loss Function Update

For analysis purpose, let us denote by

$$\widetilde{\mathbf{Loss}}_{\leq \ell}(x; \mathbf{W}, \mathbf{K}) := \Big(G^\star(x) - \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K})) \Big)^2$$

$$\mathbf{Loss}_{\leq \ell}(x; \mathbf{W}, \mathbf{K}) := \Big(G^\star(x) - \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(F_j(x; \mathbf{W}, \mathbf{K})) \Big)^2$$

$$\mathsf{OPT}_{\leq \ell} = \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[\Big(G^\star(x) - \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(G_j^\star(x)) \Big)^2 \right]$$

Lemma M.8. *Suppose the sampled set $\mathcal{Z}$ satisfies the event of Proposition J.2, Proposition J.8, Proposition J.7 (for $\varepsilon_{\mathsf{s}} \leq \varepsilon^2/100$). Suppose for some parameter $\widetilde{\kappa}_\ell \geq \kappa + L + \overline{k}_\ell$ and $\tau_\ell \geq \widetilde{\kappa}_\ell$ it satisfies*

$$\sigma_{\min}(\mathbf{K}_{\ell,\ell-1}) \geq \frac{1}{2\widetilde{\kappa}_\ell},\ \sigma_{\min}(\mathbf{K}_{\ell \lhd}) \geq \frac{1}{2\widetilde{\kappa}_\ell},\ \|\mathbf{K}_\ell\|_F, \|\mathbf{W}_\ell\|_F \leq \widetilde{\kappa}_\ell,\ \eta_2 < \frac{1}{\mathsf{poly}(\widetilde{\kappa})},\ \eta_1 \leq \frac{\eta_2}{\mathsf{poly}(\widetilde{\kappa})}$$

Suppose parameters are set to satisfy Definition H.4. Suppose the assumptions of Theorem L.1 hold for some $L = \ell - 1$, then for every constant $\gamma > 1$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{D}}[\widetilde{\mathbf{Loss}}_{\le \ell}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})})] \\
&\le (1 - 0.99\eta_1)\widetilde{\mathbf{Loss}}_{\le \ell}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1 \Big(0.04\varepsilon^2 + \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m} + (1 + \frac{1}{\gamma})^2 \mathsf{OPT}_{\le \ell}\Big)
\end{aligned}$$

Proof of Lemma M.8. Let us first focus on

$$\mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K})) = \|\mathbf{W}_j(\sigma(\mathbf{R}_{j-1}\widetilde{S}_{j-1}(x; \mathbf{K})), \dots)\|^2$$

and first consider only the movement of $\mathbf{W}$. Recall from Fact M.1 that

$$(\mathbf{W}_j^{(\mathsf{new})})^\top (\mathbf{W}_j^{(\mathsf{new})}) \leftarrow (1 - \eta_1)(\mathbf{W}_j)^\top \mathbf{W}_j + \eta_1 (\mathbf{V}_j^\star)^\top \mathbf{V}_j^\star + \sqrt{\eta_1}\xi_j$$

for some $\mathbb{E}_{\mathbf{D}}[\xi_j] = 0$ and $\mathbb{E}_{\mathbf{D}}[\|\xi_j\|_F^2] \le \mathsf{poly}(\widetilde{\kappa}_j)/m$. Therefore,

$$\mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}^{(\mathsf{new})}, \mathbf{K})) = (1 - \eta_1)\mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K})) + \eta_1 \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{V}^\star, \mathbf{K})) + \sqrt{\eta_1}\xi_{j,1} \tag{M.5}$$

for some $\xi_{j,1} = (\sigma(\mathbf{R}_{j-1}\widetilde{S}_{j-1}), \dots)^\top \xi(\sigma(\mathbf{R}_{j-1}\widetilde{S}_{j-1}), \dots)$ satisfying $\mathbb{E}[\xi_{j,1}] = 0$ and $|\xi_{j,1}| \le (\mathsf{poly}(\widetilde{\kappa}_j, \overline{B}'_j) + \|x\|^2 + \|S_1(x)\|^2)\|\xi_j\|_F$. Therefore, for every x,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{D}}[\widetilde{\mathbf{Loss}}_{\le \ell}(x; \mathbf{W}^{(\mathsf{new})}, \mathbf{K})] \\
&= \mathbb{E}_{\mathbf{D}}\Big[\Big(G^\star(x) - (1 - \eta_1)\sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K})) - \eta_1 \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{V}^\star, \mathbf{K})) + \sum_{j=2}^{\ell} \alpha_j \sqrt{\eta_1}\xi_{j,1}\Big)^2\Big] \\
&\overset{①}{=} \Big(G^\star(x) - (1 - \eta_1)\sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K})) - \eta_1 \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{V}^\star, \mathbf{K}))\Big)^2 + \eta_1 \mathbb{E}_{\mathbf{D}}\Big[\sum_{j=2}^{\ell} \alpha_j^2 \xi_{j,1}^2\Big] \\
&\overset{②}{\le} (1 - \eta_1)\Big(G^\star(x) - \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K}))\Big)^2 + \eta_1 \Big(G^\star(x) - \eta_1 \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{V}^\star, \mathbf{K}))\Big)^2 \\
&\qquad + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m} \\
&= (1 - \eta_1)\widetilde{\mathbf{Loss}}_{\le \ell}(x; \mathbf{W}, \mathbf{K}) + \eta_1 \widetilde{\mathbf{Loss}}_{\le \ell}(x; \mathbf{V}^\star, \mathbf{K}) + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m}
\end{aligned}$$

Above, ① uses the fact that $\mathbb{E}_{\mathbf{D}}[\xi_{j,1}] = 0$ and the fact that $\xi_{j,1}$ and $\xi_{j,1}$ are independent for $j \ne j$; and ② uses $((1 - \eta)a + \eta b)^2 \le (1 - \eta)a^2 + \eta b^2$, as well as the bound on $\mathbb{E}_{\mathbf{D}}[\|\xi_j\|_F^2]$ from Fact M.1.

Applying expectation with respect to $x \sim \mathcal{Z}$ on both sides, we have

$$\mathbb{E}_{\mathbf{D}}[\widetilde{\mathbf{Loss}}_{\le \ell}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K})] \le (1 - \eta_1)\widetilde{\mathbf{Loss}}_{\le \ell}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1 \widetilde{\mathbf{Loss}}_{\le \ell}(\mathcal{Z}; \mathbf{V}^\star, \mathbf{K}) + \eta_1 \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m}$$

On the other hand, for the update in $\mathbf{K}_j$ in every $j < \ell$, we can apply $\|2\mathbf{Q}_j - \mathbf{I}\|_F \le (\widetilde{\kappa}_j)^2 \|\mathbf{E}_{j,j-1}\|_F$ from Claim M.3 and apply the bounds in (M.4) to derive that (using our lower bound assumption on $\lambda_{3,j}, \lambda_{4,j}$ from Theorem L.1)

$$\|\mathbf{K}_j^{(\mathsf{new})} - \mathbf{K}_j\|_F \le \eta_1 \|\mathbf{E}_j\|_F + \eta_2 \|\mathbf{E}_{j\lhd}\|_F \cdot \mathsf{poly}(\widetilde{\kappa}_j) \le \frac{1}{\alpha_j}\big(\eta_1 \varepsilon + \eta_2 \varepsilon\big) \cdot (D_j)^8 \sqrt{\Upsilon_j^2} \cdot \frac{\sqrt{C_L}}{\sqrt{C_j}} \tag{M.6}$$

Putting this into Claim J.4 (for $L = \ell$), and using the gap assumption on $\frac{\alpha_{\ell+1}}{\alpha_\ell}$ from Definition H.4, we derive that

$$\widetilde{\mathbf{Loss}}_{\le \ell}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})})$$

$$\leq (1+0.01\eta_1)\widetilde{\mathbf{Loss}}_{\leq \ell}(\mathcal{Z};\mathbf{W}^{(\mathsf{new})},\mathbf{K}) + \eta_1 \frac{\varepsilon^2 \cdot \alpha_\ell^2}{\alpha_{\ell-1}^2}(D_{\ell-1})^{16}\Upsilon_{\ell-1}^2 \frac{C_L}{C_{\ell-1}}$$
$$\leq (1+0.01\eta_1)\widetilde{\mathbf{Loss}}_{\leq \ell}(\mathcal{Z};\mathbf{W}^{(\mathsf{new})},\mathbf{K}) + \eta_1 \frac{\varepsilon^2}{100}$$

Finally, we calculate that

$$\widetilde{\mathbf{Loss}}_{\leq \ell}(\mathcal{Z};\mathbf{V}^\star,\mathbf{K}) \overset{①}{\leq} \widetilde{\mathbf{Loss}}_{\leq \ell}(\mathcal{D};\mathbf{V}^\star,\mathbf{K}) + 0.01\varepsilon^2 \overset{②}{\leq} \big(1+\frac{1}{\gamma}\big)\mathbf{Loss}_{\leq \ell}(\mathcal{D};\mathbf{V}^\star,\mathbf{K}) + 0.02\varepsilon^2$$
$$\overset{③}{\leq} (1+\frac{1}{\gamma})^2 \mathsf{OPT}_{\leq \ell} + 0.03\varepsilon^2 \qquad \text{(M.7)}$$

where ① uses Proposition J.8 and $\gamma > 1$ is a constant, ② uses Claim J.1, and ③ uses Claim M.9 below. Combining all the inequalities we finish the proof. □

M.4.1 AUXILIARY

Claim M.9. *Suppose parameters are set to satisfy Definition H.4, and the assumptions of Theorem L.1 hold for some $L = \ell - 1$. Then, for the $\mathbf{V}^\star = (\mathbf{V}_2^\star, \dots, \mathbf{V}_\ell^\star)$ that we constructed from (M.1), and suppose $\{\alpha_j\}_j$ satisfies the gap assumption from Definition H.4, it satisfies for every constant $\gamma > 1$,*

$$\mathbf{Loss}_{\leq \ell}(\mathcal{D};\mathbf{V}^\star,\mathbf{K}) \leq \frac{\varepsilon^2}{100} + (1+\frac{1}{\gamma})\mathsf{OPT}_{\leq \ell}$$

Proof. Recalling that

$$F(x;\mathbf{W},\mathbf{K}) = \sum_\ell \alpha_\ell \mathbf{Sum}(F_\ell(x)) = \sum_\ell \alpha_\ell \left\|\mathbf{W}_\ell(\sigma(\mathbf{R}_{\ell-1}S_{\ell-1}(x)),\dots)\right\|^2$$

Using the conclusion that for every $j < \ell$, $\mathbb{E}_{x\sim\mathcal{D}}\left\|\mathbf{U}_j S_j^\star(x) - S_j(x)\right\|_2^2 \leq \delta_j^2 := (D_j)^{18}\left(\frac{\varepsilon}{\alpha_j}\right)^2 \cdot \frac{C_L}{C_\ell}$ from Corollary L.3d, one can carefully verify that (using an analogous proof to (L.11)) for every $j \leq \ell$,

$$\left\|\mathbf{V}_j^\star(\sigma(\mathbf{R}_{j-1}\mathbf{U}_{j-1}S_{j-1}^\star(x)),\dots)\right\|^2 = \left\|\mathbf{V}_j^\star(\sigma(\mathbf{R}_{j-1}S_{j-1}(x)),\dots)\right\|^2 + \xi_j$$

for some

$$\mathbb{E}[(\xi_j)^2] \leq \mathsf{poly}(\widetilde{\kappa}_j, B_j, c_3(2^j))\delta_{j-1}^2 \leq D_j(D_{j-1})^{18}\left(\frac{\varepsilon}{\alpha_{j-1}}\right)^2 \cdot \frac{C_L}{C_j}$$

Since our definition of $\mathbf{V}^\star$ satisfies (M.1), we also have for every $j \leq \ell$

$$\left\|\mathbf{V}_j^\star(\sigma(\mathbf{R}_{j-1}\mathbf{U}_{j-1}S_{j-1}^\star(x)),\dots)\right\|^2 = \mathbf{Sum}(G_j^\star(x))$$

Putting them together, and using the gap assumption on $\frac{\alpha_j}{\alpha_{j-1}}$ from Definition H.4,

$$\mathbb{E}_{x\sim\mathcal{D}}(\sum_{j=2}^{\ell}\alpha_j\mathbf{Sum}(F_j(x;\mathbf{V}^\star,\mathbf{K})) - \alpha_j\mathbf{Sum}(G_j^\star(x)))^2 \leq L\sum_{j=2}^{\ell}\alpha_j^2 D_j(D_{j-1})^{19}\left(\frac{\varepsilon}{\alpha_{j-1}}\right)^2 \cdot \frac{C_L}{C_j} \leq \frac{\varepsilon^2}{100(1+\gamma)} \; .$$

Finally, using Young's inequality that

$$\mathbf{Loss}_{\leq \ell}(x;\mathbf{V}^\star,\mathbf{K}) \leq (1+\frac{1}{\gamma})\left(\sum_{\ell=2}^{\ell}\alpha_\ell\mathbf{Sum}(G_\ell^\star(x)) - G^\star(x)\right)^2$$
$$+ (1+\gamma)\left(\sum_{\ell=2}^{L}\alpha_\ell\mathbf{Sum}(F_\ell(x;\mathbf{V}^\star,\mathbf{K})) - \alpha_\ell\mathbf{Sum}(G_\ell^\star(x))\right)^2$$

we finish the proof. □

M.5 Objective Decrease Direction: Stage $\ell^{\triangle}$

Theorem M.10. *Suppose we are in stage $\ell^{\triangle}$, meaning that $\lambda_{3,j} = \lambda_{4,j} = \lambda_{5,j} = 0$ for $j \geq \ell$ and the trainable parameters are $\mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{K}_1, \ldots, \mathbf{K}_{\ell-1}$. Suppose it satisfies*

$$\varepsilon^2 := \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \left(\frac{\alpha_{\ell-1}}{(D_{\ell-1})^9 \Upsilon_{\ell-1}}\right)^2 \quad \textit{and} \quad \left\{\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|_2^2] \leq \tau_j\right\}_{j<\ell}$$

Suppose the sampled set $\mathcal{Z}$ satisfies the event of Proposition J.2, Proposition J.8, Proposition J.7 (for $\varepsilon_{\mathsf{s}} \leq \varepsilon^2/100$). Suppose parameters are set to satisfy Definition H.4. Then, for every $\eta_2 < \frac{1}{\mathsf{poly}(\widetilde{\kappa})}$ and $\eta_1 \leq \frac{\eta_2}{\mathsf{poly}(\widetilde{\kappa})}$,

$$\mathbb{E}_{\mathbf{D}} \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}) \leq (1 - 0.7\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + 2\eta_1 \alpha_{\ell+1}^2$$

And also we have $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|^2] \leq 2B_j$ for every $j < \ell$.

Proof of Theorem M.10. We first verify the prerequisites of many of the lemmas we need to invoke.

Prerequisite 1. Using $\lambda_{6,\ell} \geq \frac{\varepsilon^2}{(\widetilde{\kappa}_\ell)^2}$ and $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2$, we have

$$\|\mathbf{K}_\ell\|_F, \|\mathbf{W}_\ell\|_F \leq \widetilde{\kappa}_\ell$$

which is a prerequisite for Lemma M.4, Lemma M.6, Lemma M.8 that we need to invoke.

Prerequisite 2. Applying Proposition J.7, we have

$$\widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2 \xRightarrow{\text{Proposition J.7}} \widetilde{\mathbf{Loss}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 2\varepsilon^2 \tag{M.8}$$

Since $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|^2] \leq \tau_j$ for all $j < \ell$, we can apply Claim J.1 and get

$$\widetilde{\mathbf{Loss}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 2\varepsilon^2 \xRightarrow{\text{Claim J.1 and choice } B'} \mathbf{Loss}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 3\varepsilon^2 \tag{M.9}$$

Next, consider a dummy loss function against only the first $\ell - 1$ layers

$$\mathbf{Loss}_{dummy}(\mathcal{D}; \mathbf{W}, \mathbf{K}) := \sum_{x\sim\mathcal{D}} \Big[\Big(\sum_{j=2}^{\ell-1} \alpha_j \mathbf{Sum}(F_j(x)) - \alpha_j \mathbf{Sum}(G_j^\star(x))\Big)^2\Big] \leq 1.1\mathbf{Loss}(\mathcal{D}; \mathbf{W}, \mathbf{K}) + O(\alpha_\ell^2) \leq 4\varepsilon^2$$

so in the remainder of the proof we can safely apply Theorem L.1 and Corollary L.3 for $L = \ell - 1$. Note that this is also a prerequisite for Lemma M.8 with ℓ layers that we want to invoke. As a side note, we can use Corollary L.3d to derive

$$\forall j < \ell: \qquad \mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|^2] \leq 2B_j \ .$$

Prerequisite 3. Corollary L.3b tells us for every $j < \ell$,

$$\begin{aligned}\left\|\mathbf{Q}_{j-1}^\top \overline{\mathbf{K}}_{j,j-1}^\top \overline{\mathbf{K}}_{j\lhd} \mathbf{Q}_{j\lhd} - \overline{\mathbf{W}^\star}_{j,j-1}^\top \overline{\mathbf{W}^\star}_{j\lhd}\right\|_F^2 &\leq \Upsilon_j (D_j)^4 \left(\frac{\varepsilon}{\alpha_j}\right)^2 \frac{C_\ell}{C_j} \\ &\overset{①}{\leq} \frac{\Upsilon_j (D_j)^4}{\Upsilon_{\ell-1}^2 (D_{\ell-1})^{18}} \left(\frac{\alpha_{\ell-1}}{\alpha_j}\right)^2 \frac{C_\ell}{C_j} \overset{②}{\leq} \frac{1}{(D_j)^{14}}\end{aligned} \tag{M.10}$$

Above, inequality ① uses the assumption $\varepsilon \leq \frac{\alpha_{\ell-1}}{(D_{\ell-1})^9 \Upsilon_{\ell-1}}$. Inequality ② holds when $j = \ell - 1$ by using $\frac{1}{\Upsilon_{\ell-1}} \frac{C_\ell}{C_{\ell-1}} \ll 1$ from our sufficiently large choice of $\Upsilon_{\ell+1}$, and ineuqliaty ② holds when $j < \ell - 1$ using the gap assumption on $\frac{\alpha_j}{\alpha_{j-1}}$ when $j < \ell - 1$.

Note that the left hand side of (M.10) is identical to (since $\overline{\mathbf{K}}_{j,i} \mathbf{Q}_i = \mathbf{K}_{j,i}(\mathbf{R}_i \mathbf{U}_i * \mathbf{R}_i \mathbf{U}_i)$)

$$\left\|\mathbf{A}\mathbf{K}_{j,j-1}^\top \mathbf{K}_{j\lhd} \mathbf{B} - \mathbf{C}(\mathbf{W}_{j,j-1}^\star)^\top \mathbf{W}_{j\lhd}^\star \mathbf{D}\right\|_F^2$$

for some well-conditioned sqaure matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ with singular values between $[\frac{1}{\mathsf{poly}(\overline{k}_j, L)}, O(\mathsf{poly}(\overline{k}_j, L))]$ (see Lemma I.6 and Lemma I.5). Therefore, combining the facts that

(1) $\mathbf{K}_{j,j-1}^\top \mathbf{K}_{j\lhd}$ and $(\mathbf{W}^\star_{j,j-1})^\top \mathbf{W}^\star_{j\lhd}$ are both of rank exactly k_j, (2) $\|\mathbf{K}_j\| \leq \widetilde{\kappa}_j$, (3) minimal singular value $\sigma_{\min}(\mathbf{W}^\star_{j,i}) \geq 1/\kappa$, we must have

$$\sigma_{\min}(\mathbf{K}_{j,j-1}) \geq \frac{1}{\widetilde{\kappa}_j \cdot \mathsf{poly}(\overline{k}_j, \kappa, L)} \quad \text{and} \quad \sigma_{\min}(\mathbf{K}_{j\lhd}) \geq \frac{1}{\widetilde{\kappa}_j \cdot \mathsf{poly}(\overline{k}_j, \kappa, L)}$$

as otherwise this will contract to (M.10). This lower bound on the minimum singular value is a prerequisite for Lemma M.4, Lemma M.6 that we need to invoke.

Prerequisite 4. Using Corollary L.3b, we also have for every $j < \ell$ (see the calculation in (M.4))

$$\begin{aligned}\|\mathbf{E}_{j\lhd}\|_F^2 = \|\mathbf{K}_{j,j-1}^\top \mathbf{K}_{j\lhd} - \left(\mathbf{V}^\star_{j,j-1}\right)^\top \mathbf{V}^\star_{j\lhd}\|_F^2 &\leq \left(\frac{\varepsilon}{\alpha_j}\right)^2 \Upsilon_j \cdot (D_j)^5 \cdot \frac{C_\ell}{C_j} \\ &\leq \left(\frac{\alpha_{\ell-1}}{\alpha_j}\right)^2 \cdot \frac{\Upsilon_j (D_j)^5}{\Upsilon_{\ell-1}(D_{\ell-1})^{18}} \cdot \frac{C_\ell}{C_j} \leq \frac{1}{(D_j)^{13}}\end{aligned}$$

which is a prerequisite for Lemma M.4 that we need to invoke.

Main Proof Begins. Now we are fully prepared and can begin the proof. In the language of this section, our objective

$$\begin{aligned}\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) = \widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) &+ \sum_{j<\ell} \left(\lambda_{3,j} \|\mathbf{R}_{3,j}\|_F^2 + \lambda_{4,j} \|\mathbf{R}_{4,j}\|_F^2 + \lambda_{5,j} \|\mathbf{R}_{5,j}\|_F^2 + \lambda_{6,j} \mathbf{R}_{6,j}\right) \\ &+ \sum_{j \leq \ell} \lambda_{6,j} \left(\mathbf{R}_{7,j}\right)\end{aligned}$$

We can apply Lemma M.4 to bound the decrease of $\mathbf{R}_{6,j}$ for $j < \ell$ and $\mathbf{R}_{7,j}$ for $j \leq \ell$, apply Lemma M.6 to bound the decrease of $\mathbf{R}_{3,j}, \mathbf{R}_{4,j}, \mathbf{R}_{5,j}$ for $j < \ell$, and apply Lemma M.8 to bound the decrease of $\widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$ (with the choice $\mathsf{OPT}_{\leq \ell} \leq 2\alpha_{\ell+1}^2$). By combining all the lemmas, we have (using $\eta_2 = \eta_1/\mathsf{poly}(\widetilde{\kappa})$ and sufficiently small choice of η_1)

$$\begin{aligned}&\mathbb{E}_{\mathbf{D}} \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}) \\ &\overset{①}{\leq} (1 - 0.9\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1(\varepsilon_{sample} + \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m}) + \eta_1 \sum_{j \leq \ell} \lambda_{6,j} \mathsf{poly}(k_j, L, \kappa) + 2\eta_1 \alpha_{\ell+1}^2 \\ &\qquad + \eta_1 \sum_{j<\ell} \left(\frac{1}{\Upsilon_j} + \frac{\Upsilon_j}{\Upsilon_j^2} + \frac{\Upsilon_j^2}{\Upsilon_j^3}\right) \varepsilon^2 (D_j)^4 \frac{C_\ell}{C_j} \\ &\overset{②}{\leq} (1 - 0.8\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1(\varepsilon_{sample} + \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m}) + \eta_1 \sum_{j \leq \ell} \lambda_{6,j} \mathsf{poly}(k_j, L, \kappa) + 2\eta_1 \alpha_{\ell+1}^2 \\ &\overset{③}{\leq} (1 - 0.7\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + 2\eta_1 \alpha_{\ell+1}^2\end{aligned}$$

Above, inequality ① uses our parameter choices that $\lambda_{3,j} = \frac{\alpha_j^2}{(D_j)\Upsilon_j}$, $\lambda_{4,j} = \frac{\alpha_j^2}{(D_j)^7 \Upsilon_j^2}$, and $\lambda_{5,j} = \frac{\alpha_j^2}{\Upsilon_j^3 (D_j)^{13}}$. Inequality ② uses our choices of Υ_j (see Definition H.4). Inequality ③ uses $m \geq \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{\varepsilon^2}$ from Definition H.4, $\varepsilon_{\mathsf{s}} \leq 0.01\varepsilon^2$, and $\lambda_{6,j} = \frac{\varepsilon^2}{\widetilde{\kappa}_j^2} \leq \frac{\varepsilon^2}{\mathsf{poly}(k_j, L, \kappa)}$ from Definition H.4. □

M.6 Objective Decrease Direction: Stage $\ell^\triangledown$

Theorem M.11. *Suppose we are in stage $\ell^\triangledown$, meaning that $\lambda_{3,j} = \lambda_{4,j} = \lambda_{5,j} = 0$ for $j > \ell$ and the trainable parameters are $\mathbf{W}_1, \dots, \mathbf{W}_\ell, \mathbf{K}_1, \dots, \mathbf{K}_\ell$. Suppose it satisfies*

$$\left(\frac{\alpha_\ell}{(D_\ell)^9 \Upsilon_\ell}\right)^2 \leq \varepsilon^2 := \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) \leq \left(\frac{\alpha_\ell}{(D_\ell)^3 \sqrt{\Upsilon_\ell}}\right)^2 \quad \textit{and} \quad \left\{\mathbb{E}_{x \sim \mathcal{D}}[\|S_j(x)\|_2^2] \leq \tau\right\}_{j<\ell}$$

Suppose the sampled set $\mathcal{Z}$ satisfies the event of Proposition J.2, Proposition J.8, Proposition J.7 (for $\varepsilon_{\mathsf{s}} \leq \varepsilon^2/100$). Suppose parameters are set to satisfy Definition H.4. Then, for every $\eta_2 < \frac{1}{\mathsf{poly}(\widetilde{\kappa})}$

and $\eta_1 \leq \frac{\eta_2}{\mathsf{poly}(\widetilde{\kappa})}$,

$$\mathbb{E}_{\mathbf{D}} \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}) \leq (1 - 0.7\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + 2\eta_1 \alpha_{\ell+1}^2$$

And also we have $\mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|^2] \leq 2B_j$ *for every* $j < \ell$. *Furthermore, if* $\varepsilon^2 \leq \left(\frac{\alpha_\ell}{(D_\ell)^9 \Upsilon_\ell}\right)^2$ *then we also have* $\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2] \leq 2B_\ell$.

Proof of Theorem M.11. The proof is analogous to Theorem M.10 but with several changes.

Prerequisite 1. For analogous reasons, we have

$$\|\mathbf{K}_\ell\|_F, \|\mathbf{W}_\ell\|_F \leq \widetilde{\kappa}_\ell$$

which is a prerequisite for Lemma M.4, Lemma M.7, Lemma M.8 that we need to invoke.

Prerequisite 2. This time, we have $\varepsilon^2 \leq \frac{\alpha_\ell}{(D_\ell)^3\sqrt{\Upsilon_\ell}}$. This means the weaker assumption of Corollary L.4 has been satisfied for $L = \ell$, and as a result Theorem L.1 and Corollary L.3 hold with $L = \ell - 1$. This is a prerequisite for Lemma M.8 with ℓ layers that we want to invoke. Note in particular, Corollary L.3d implies

$$\forall j < \ell: \qquad \mathbb{E}_{x\sim\mathcal{D}}[\|S_j(x)\|^2] \leq 2B_j \ .$$

Note also, if $\varepsilon^2 \leq \left(\frac{\alpha_\ell}{(D_\ell)^9 \Upsilon_\ell}\right)^2$, then Corollary L.3 holds with $L = \ell$, so we can invoke Corollary L.3e to derive the above bound for $j = \ell$.

$$\mathbb{E}_{x\sim\mathcal{D}}[\|S_\ell(x)\|^2] \leq 2B_\ell$$

Prerequisite 3. Again using Corollary L.3b for $L = \ell - 1$, we can derive for all $j < \ell$

$$\sigma_{\min}(\mathbf{K}_{j,j-1}) \geq \frac{1}{\widetilde{\kappa}_j \cdot \mathsf{poly}(k_j, \kappa, L)} \quad \text{and} \quad \sigma_{\min}(\mathbf{K}_{j\triangleleft}) \geq \frac{1}{\widetilde{\kappa}_j \cdot \mathsf{poly}(k_j, \kappa, L)}$$

This time, one can also use Corollary L.4b with $L = \ell$ to derive that the above holds also for $j = \ell$. This is a prerequisite for Lemma M.4, Lemma M.7 that we need to invoke.

Prerequisite 4. Using Corollary L.3b, we also have for every $j < \ell$ (see the calculation in (M.4))

$$\|\mathbf{E}_{j\triangleleft}\|_F^2 = \|\mathbf{K}_{j,j-1}^\top \mathbf{K}_{j\triangleleft} - \left(\mathbf{V}^\star_{j,j-1}\right)^\top \mathbf{V}^\star_{j\triangleleft}\|_F^2 \leq \frac{1}{(D_j)^{13}}$$

This time, one can also use Corollary L.4b with $L = \ell$ to derive that the above holds also for $j = \ell$.

Main Proof Begins. Now we are fully prepared and can begin the proof. In the language of this section, our objective

$$\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) = \widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \sum_{j<\ell} \left(\lambda_{3,j}\|\mathbf{R}_{3,j}\|_F^2 + \lambda_{4,j}\|\mathbf{R}_{4,j}\|_F^2 + \lambda_{5,j}\|\mathbf{R}_{5,j}\|_F^2 + \lambda_{6,j}\mathbf{R}_{6,j}\right)$$
$$+ \sum_{j\leq\ell} \lambda_{6,j}\left(\mathbf{R}_{7,j}\right)$$

We can apply Lemma M.4 to bound the decrease of $\mathbf{R}_{6,j}, \mathbf{R}_{7,j}$ for $j \leq \ell$, apply Lemma M.7 to bound the decrease of $\mathbf{R}_{3,j}, \mathbf{R}_{4,j}, \mathbf{R}_{5,j}$ for $j \leq \ell$, and apply Lemma M.8 to bound the decrease of $\widetilde{\mathbf{Loss}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$ (with the choice $\mathsf{OPT}_{\leq\ell} \leq 2\alpha_{\ell+1}^2$). By combining all the lemmas, we have (using $\eta_2 = \eta_1/\mathsf{poly}(\widetilde{\kappa})$ and sufficiently small choice of η_1)

$$\mathbb{E}_{\mathbf{D}} \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})})$$
$$\overset{①}{\leq} (1 - 0.9\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1(\varepsilon_{sample} + \frac{\mathsf{poly}(\widetilde{\kappa}, B')}{m}) + \eta_1 \sum_{j\leq\ell} \lambda_{6,j}\mathsf{poly}(k_j, L, \kappa) + 2\eta_1\alpha_{\ell+1}^2$$

$$+\eta_1\left(\frac{1}{\Upsilon_\ell}+\frac{\Upsilon_\ell}{\Upsilon_\ell^2}+\frac{\Upsilon_\ell^2}{\Upsilon_\ell^3}\right)\varepsilon^2(D_\ell)^4+\eta_1\sum_{j<\ell}\left(\frac{1}{\Upsilon_j}+\frac{\Upsilon_j}{\Upsilon_j^2}+\frac{\Upsilon_j^2}{\Upsilon_j^3}\right)(\alpha_\ell)^2 D_\ell (D_j)^4\frac{C_\ell}{C_j}$$

$$\overset{②}{\le}(1-0.9\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z};\mathbf{W},\mathbf{K})+\eta_1(\varepsilon_{sample}+\frac{\mathsf{poly}(\widetilde{\kappa},B')}{m})+\eta_1\sum_{j\le\ell}\lambda_{6,j}\mathsf{poly}(k_j,L,\kappa)+2\eta_1\alpha_{\ell+1}^2$$

$$+\eta_1\left(\frac{1}{\Upsilon_\ell}+\frac{\Upsilon_\ell}{\Upsilon_\ell^2}+\frac{\Upsilon_\ell^2}{\Upsilon_\ell^3}\right)\varepsilon^2(D_\ell)^4+\eta_1\sum_{j<\ell}\left(\frac{1}{\Upsilon_j}+\frac{\Upsilon_j}{\Upsilon_j^2}+\frac{\Upsilon_j^2}{\Upsilon_j^3}\right)\varepsilon^2(D_\ell)^{19}\Upsilon_\ell^2(D_j)^4\frac{C_\ell}{C_j}$$

$$\overset{③}{\le}(1-0.8\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z};\mathbf{W},\mathbf{K})+\eta_1(\varepsilon_{sample}+\frac{\mathsf{poly}(\widetilde{\kappa},B')}{m})+\eta_1\sum_{j\le\ell}\lambda_{6,j}\mathsf{poly}(k_j,L,\kappa)+2\eta_1\alpha_{\ell+1}^2$$

$$\overset{④}{\le}(1-0.7\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z};\mathbf{W},\mathbf{K})+2\eta_1\alpha_{\ell+1}^2$$

Above, inequality ① uses our parameter choices that $\lambda_{3,j}=\frac{\alpha_j^2}{(D_j)\Upsilon_j}$, $\lambda_{4,j}=\frac{\alpha_j^2}{(D_j)^7\Upsilon_j}$, and $\lambda_{5,j}=\frac{\alpha_j^2}{(D_j)^{13}}$. Inequality ② uses our assumption that $\varepsilon\ge\frac{\alpha_\ell}{(D_\ell)^9\Upsilon_\ell}$. Inequality ③ uses our choices of Υ_j (see Definition H.4). Inequality ④ uses $m\ge\frac{\mathsf{poly}(\widetilde{\kappa},B')}{\varepsilon^2}$ from Definition H.4, $\varepsilon_{\mathsf{s}}\le 0.01\varepsilon^2$, and $\lambda_{6,j}=\frac{\varepsilon^2}{\widetilde{\kappa}_j^2}\le\frac{\varepsilon^2}{\mathsf{poly}(k_j,L,\kappa)}$ from Definition H.4. □

N Extension to Classification

Let us assume without loss of generality that $\mathbf{Var}[G^\star(x)]=\frac{1}{C\cdot c_3(2^L)}$ for some sufficiently large constant $C>1$. We have the following proposition that relates the ℓ_2 and cross entropy losses. (Proof see Appendix N.2.)

Proposition N.1. *For every function $F(x)$ and $\varepsilon\ge 0$, we have*

1. *If $F(x)$ is a polynomial of degree 2^L and $\mathbb{E}_{(x_0,x)\sim\mathcal{D}}\,\mathsf{CE}\left(Y(x_0,x),v(x_0+F(x))\right)\le\varepsilon$ for some there $v\ge 0$, then*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left(F(x)-G^\star(x)\right)^2=O(c_3(2^L)^2\varepsilon^2)$$

2. *If $\mathbb{E}_{x\sim\mathcal{D}}\left(F(x)-G^\star(x)\right)^2\le\varepsilon^2$ and $v\ge 0$, then*

$$\mathop{\mathbb{E}}_{(x_0,x)\sim\mathcal{D}}\mathsf{CE}\left(Y(x_0,x),v(x_0+F(x))\right)\le O\left(v\varepsilon^2+\frac{\log^2 v}{v}\right)$$

At a high level, when setting $v=\frac{1}{\varepsilon}$, Proposition N.1 implies, up to small factors such as $c_3(2^L)$ and $\log(1/\varepsilon)$, it satisfies

$$\ell_2\text{-loss}=\varepsilon^2\iff\text{cross-entropy loss}=\varepsilon$$

Therefore, applying SGD on the ℓ_2 loss (like we do in this paper) should behave very similarly to applying SGD on the cross-entropy loss.

Of course, to turn this into an actual rigorous proof, there are subtleties. Most notably, we cannot naively convert back and forth between cross-entropy and ℓ_2 losses for *every SGD step*, since doing so we losing a multiplicative factor per step, killing the objective decrease we obtain. Also, one has to deal with truncated activation vs. quadratic activation. In the next subsection, we sketch perhaps the simplest possible way to prove our classification theorem by reducing its proof to that of our ℓ_2 regression theorem.

N.1 Detail Sketch: Reduce the Proof to Regression

Let us use the same parameters in Definition H.4 with minor modifications:

- additionally require one $\log(1/\varepsilon)$ factor in the gap assumption $\frac{\alpha_{\ell+1}}{\alpha_\ell}$,[35]
- additionally require one $1/\varepsilon$ factor in the over-parameterization m, and
- additionally require one $\mathsf{poly}(d)$ factor in the sample complexity N.

Recall from Theorem M.10 and Theorem M.11 that the main technical statement for the convergence in the regression case was to construct some $\mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}$ satisfying

$$\mathop{\mathbb{E}}_{\mathbf{D}} \widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}) \leq (1-0.7\eta_1)\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + 2\eta_1 \alpha_{\ell+1}^2 \ .$$

We show that the same construction $\mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}$ also satisfies, denoting by $\varepsilon = \widetilde{\mathbf{Obj}}^{\mathsf{xE}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$,

$$\mathop{\mathbb{E}}_{\mathbf{D}} \widetilde{\mathbf{Obj}}^{\mathsf{xE}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}) \leq (1-0.7\eta_1)\widetilde{\mathbf{Obj}}^{\mathsf{xE}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1 \cdot O(\frac{\log^2(1/\varepsilon)}{\varepsilon}) \cdot \alpha_{\ell+1}^2 \ . \tag{N.1}$$

This means the objective can sufficiently decrease at least until $\varepsilon \approx \alpha_{\ell+1} \cdot \log \frac{1}{\alpha_{\ell+1}}$ (or to arbitrarily small when $\ell = L$). The rest of the proof will simplify follow from here.

Quick Observation. Let us assume without loss of generality that $v = \frac{\log(1/\varepsilon)}{100\varepsilon}$ always holds.[36] Using an analogous argument to Proposition J.7 and Claim J.1, we also have

$$\widetilde{\mathbf{Obj}}^{\mathsf{xE}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 2\varepsilon \quad \text{and} \quad \mathbf{Obj}^{\mathsf{xE}}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq 3\varepsilon \ .$$

Applying Lemma N.1, we immediately know $\mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq O(c_3(2^L)^2\varepsilon^2)$ for the original ℓ_2 objective. Therefore, up to a small factor $c_3(2^L)^2$, the old inequality $\mathbf{Obj}(\mathcal{D}; \mathbf{W}, \mathbf{K}) \leq \varepsilon^2$ remains true. This ensures that we can still apply many of the technical lemmas (especially the critical Lemma L.1 and the regularizer update Lemma M.6).

Going back to (N.1). In order to show sufficient objective value decrease in (N.1), in principle one needs to look at loss function decrease as well as regularizer decrease. This is what we did in the proofs of Theorem M.10 and Theorem M.11 for the regression case.

Now for classification, the regularizer decrease *remains the same as before* since we are using the same regularizer. The only technical lemma that requires non-trivial changes is Lemma M.8 which talks about loss function decrease from $\mathbf{W}, \mathbf{K}$ to $\mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}$. As before, let us write for notational simplicity

$$\widetilde{F}_{\leq \ell}(x; \mathbf{W}, \mathbf{K}) := \sum_{j=2}^{\ell} \alpha_j \mathbf{Sum}(\widetilde{F}_j(x; \mathbf{W}, \mathbf{K}))$$

$$\widetilde{\mathbf{Loss}}_{\leq \ell}^{\mathsf{xE}}(x_0, x; \mathbf{W}, \mathbf{K}) := \mathsf{CE}(Y(x_0, x), v(x_0 + \widetilde{F}_{\leq \ell}(x; \mathbf{W}, \mathbf{K})))$$

One can show that the following holds (proved in Appendix N.1.1):

Lemma N.2 (classification variant of Lemma M.8)**.**

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{D}} \widetilde{\mathbf{Loss}}_{\leq \ell}^{\mathsf{xE}}(\mathcal{Z}; \mathbf{W}^{(\mathsf{new})}, \mathbf{K}^{(\mathsf{new})}) \\
&\leq (1-\eta_1)\widetilde{\mathbf{Loss}}_{\leq \ell}^{\mathsf{xE}}(\mathcal{Z}; \mathbf{W}, \mathbf{K}) + \eta_1 \left(\frac{O(\log^2(1/\varepsilon))}{\varepsilon} \mathsf{OPT}_{\leq \ell} + 0.1\varepsilon + \frac{v^2 \cdot \mathsf{poly}(\widetilde{\kappa}, B')}{m} \right)
\end{aligned}$$

Combining this with the regularizer decrease lemmas, we arrive at (N.1).

[35] We need this log factor because there is a logarithmic factor loss when translating between cross-entropy and the ℓ_2 loss (see Lemma N.1). This log factor prevents us from working with extremely small $\varepsilon > 0$, and therefore we have required $\varepsilon > \frac{1}{d^{100 \log d}}$ in the statement of Theorem 4.

[36] This can be done by setting $v = \frac{\log(1/\varepsilon_0)}{100\varepsilon_0}$ where ε_0 is the current target error in Algorithm 1. Since ε and ε_0 are up to a factor of at most 2, the equation $v = \frac{\log(1/\varepsilon)}{100\varepsilon}$ holds up to a constant factor. Also, whenever ε_0 shrinks by a factor of 2 in Algorithm 1, we also increase v accordingly. This is okay, since it increases the objective value $\widetilde{\mathbf{Obj}}(\mathcal{Z}; \mathbf{W}, \mathbf{K})$ by more than a constant factor.

N.1.1 PROOF OF LEMMA N.2

Sketched proof of Lemma N.2. Let us rewrite

$$\widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K}^{(\mathsf{new})}) = (1-\eta_1)\widetilde{F}_{\leq\ell}(x;\mathbf{W},\mathbf{K}) + \eta_1 H(x) + Q(x) \tag{N.2}$$

$$\text{for } H(x) := \frac{\widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K}^{(\mathsf{new})}) - \widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K})}{\eta_1} + \widetilde{F}_{\leq\ell}(x;\mathbf{V}^\star,\mathbf{K})$$

$$\text{for } Q(x) := \widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K}) - \eta_1\widetilde{F}_{\leq\ell}(x;\mathbf{V}^\star,\mathbf{K}) - (1-\eta_1)\widetilde{F}_{\leq\ell}(x;\mathbf{W},\mathbf{K})$$

We make two observations from here.

- First, we can calculate the ℓ_2 loss of the auxilary function $H(x)$. The original proof of Lemma M.8 can be modified to show the following (proof in Appendix N.1.2)

  Claim N.3. $\mathbb{E}_{x\sim\mathcal{D}}(G^\star(x)-H(x))^2 \leq 0.00001\frac{\varepsilon^2}{\log^2(1/\varepsilon)} + 6\mathsf{OPT}_{\leq\ell}$.

  Using Lemma N.1, and our choice of $v = \frac{100\log^2(1/\varepsilon)}{\varepsilon}$, we can connect this back to the cross entropy loss:

  $$\mathbb{E}_{(x_0,x)\sim\mathcal{D}}\mathsf{CE}\left(Y(x_0,x), v(x_0+H(x))\right) \leq \frac{O(\log^2(1/\varepsilon))}{\varepsilon}\mathsf{OPT}_{\leq\ell} + 0.09\varepsilon$$

  Through a similar treatment to Proposition J.8 we can also translate this to the training set

  $$\mathbb{E}_{(x_0,x)\sim\mathcal{Z}}\mathsf{CE}\left(Y(x_0,x), v(x_0+H(x))\right) \leq \frac{O(\log^2(1/\varepsilon))}{\varepsilon}\mathsf{OPT}_{\leq\ell} + 0.1\varepsilon \tag{N.3}$$

- Second, recall from (M.5) in the original proof of Lemma M.8 that we have

  $$\begin{aligned}\mathbb{E}_{\mathbf{D}}[(Q(x))^2] &= \mathbb{E}_{\mathbf{D}}\left(\widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K}) - \eta_1\widetilde{F}_{\leq\ell}(x;\mathbf{V}^\star,\mathbf{K}) - (1-\eta_1)\widetilde{F}_{\leq\ell}(x;\mathbf{W},\mathbf{K})\right)^2\\ &= \mathbb{E}_{\mathbf{D}}\left(\sum_{j=2}^{\ell}\alpha_j\xi_{j,1}\right)^2 \leq \eta_1\frac{\mathsf{poly}(\widetilde{\kappa},B')}{m}\ .\end{aligned} \tag{N.4}$$

  as well as $\mathbb{E}_{\mathbf{D}}[Q(x)] = 0$.

We are now ready to go back to (N.2), and apply convexity and the Lipscthiz smoothness of the cross-entropy loss function to derive:

$$\begin{aligned}\mathbb{E}_{\mathbf{D}}\widetilde{\mathbf{Loss}}^{\mathsf{xE}}_{\leq\ell}(\mathcal{Z};\mathbf{W}^{(\mathsf{new})},\mathbf{K}^{(\mathsf{new})}) \leq (1-\eta_1)\widetilde{\mathbf{Loss}}^{\mathsf{xE}}_{\leq\ell}(\mathcal{Z};\mathbf{W},\mathbf{K}) + \eta_1\mathbb{E}_{(x_0,x)\sim\mathcal{Z}}[\mathsf{CE}(Y(x_0,x),v(x_0+H(x)))]\\ + v^2\cdot\mathbb{E}_{\mathbf{D}}[(Q(x))^2]\end{aligned}$$

Plugging (N.3) and (N.4) into the above formula, we finish the proof. □

N.1.2 PROOF OF CLAIM N.3

Proof of Claim N.3. Let us write

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{Z}}(G^\star(x)-H(x))^2 \leq \frac{2}{(\eta_1)^2}\mathbb{E}_{x\sim\mathcal{Z}}\left(\widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K}^{(\mathsf{new})}) - \widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K})\right)^2\\ + 2\mathbb{E}_{x\sim\mathcal{Z}}\left(G^\star(x) - \widetilde{F}_{\leq\ell}(x;\mathbf{V}^\star,\mathbf{K})\right)^2\end{aligned}$$

- For the first term, the same analysis of Claim J.4 gives

  $$\begin{aligned}&\mathbb{E}_{x\sim\mathcal{Z}}\left(\widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K}^{(\mathsf{new})}) - \widetilde{F}_{\leq\ell}(x;\mathbf{W}^{(\mathsf{new})},\mathbf{K})\right)^2\\ &\leq \alpha_\ell^2\mathsf{poly}(\widetilde{\kappa}_{\ell-1},\overline{B}'_{\ell-1})\|\mathbf{K}^{(\mathsf{new})}-\mathbf{K}\|_F^2 \leq (\eta_1)^2\frac{\varepsilon^2}{1000000\log^2(1/\varepsilon)}\end{aligned}$$

where the last inequality has used the upper bound on $\|\mathbf{K}_j^{(\mathsf{new})} - \mathbf{K}_j\|_F$ for $j < \ell$ — see (M.6) in the original proof of Lemma M.8 — as well as the gap assumption on $\frac{\alpha_\ell}{\alpha_{\ell-1}}$ (with an additional $\log(1/\varepsilon)$ factor).

- For the second term, the original proof of Lemma M.8 — specifically (M.7) — already gives

$$\mathop{\mathbb{E}}_{x\sim\mathcal{Z}}\left(G^\star(x) - \widetilde{F}_{\leq\ell}(x;\mathbf{V}^\star,\mathbf{K})\right)^2 = \widetilde{\mathbf{Loss}}_{\leq\ell}(\mathcal{Z};\mathbf{V}^\star,\mathbf{K}) \leq (1+\frac{1}{\gamma})^2\mathsf{OPT}_{\leq\ell} + \frac{\varepsilon^2}{1000000\log^2(1/\varepsilon)}$$

where the additional $\log(1/\varepsilon)$ factor comes from the gap assumption on $\frac{\alpha_\ell}{\alpha_{\ell-1}}$.

Putting them together, and applying a similar treatment to Proposition J.7 to go from the training set $\mathcal{Z}$ to the population $\mathcal{D}$, we have the desired bound. □

N.2 Proof of Proposition N.1

Proof of Proposition N.1.

1. Suppose by way of contradiction that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}(F(x) - G^\star(x))^2 = \Omega\left(c_3(2^L)^2\varepsilon^2\right)$$

Let us recall a simple probability fact. Given any random variable $X \geq 0$, it satisfies[37]

$$\mathbf{Pr}[X > \frac{1}{2}\sqrt{\mathbb{E}[X^2]}] \geq \frac{9}{16}\frac{(\mathbb{E}[X^2])^2}{\mathbb{E}[X^4]}$$

Let us plug in $X = |F(x) - G^\star(x)|$, so by the hyper-contractivity Property D.3, with probability at least $\Omega\left(\frac{1}{c_3(2^L)}\right)$ over $x \sim \mathcal{D}$,

$$|F(x) - G^\star(x)| = \Omega(c_3(2^L)\varepsilon)$$

Also by the hyper-contractivity Property D.3 and Markov's inequality, with probability at least $1 - O\left(\frac{1}{c_3(2^L)}\right)$,

$$G^\star(x) \leq \mathbb{E}[G^\star(x)] + O(c_3(2^L)) \cdot \sqrt{\mathbf{Var}[G^\star(x)]} \leq \mathbb{E}[G^\star(x)] + 1$$

When the above two events over x both take place — this happens with probability $\Omega(\frac{1}{c_3(2^L)})$ — we further have with probability at least $\Omega(c_3(2^L)\varepsilon)$ over x_0, it satisfies $\mathsf{sgn}(x_0 + F(x)) \neq \mathsf{sgn}(x_0 + G^\star(x)) = Y(x_0, x)$. This implies $\mathbb{E}_{(x_0,x)\sim\mathcal{D}}\,\mathsf{CE}\left(Y(x_0,x), v(x_0 + F(x))\right) > \varepsilon$ using the definition of cross entropy, giving a contradiction.

2. By the Lipschitz continuity of the cross-entropy loss, we have that

$$\begin{aligned}\mathsf{CE}\left(Y(x_0,x), v(x_0+F(x))\right) &\leq \mathsf{CE}\left(Y(x_0,x), v(x_0+G^\star(x))\right) + O\left(v|G^\star(x) - F(x)|\right)\\ &\leq O\left(1 + v|G^\star(x) - F(x)|\right)\end{aligned}$$

Now, for a fixed x, we know that if $x_0 \geq -G^\star(x) + |G^\star(x) - F(x)| + 10\frac{\log v}{v}$ or $x_0 \leq -G^\star(x) - |G^\star(x) - F(x)| - 10\frac{\log v}{v}$, then $\mathsf{CE}\left(Y(x_0,x), v(x_0+F(x))\right) \leq \frac{1}{v}$. This implies

$$\begin{aligned}&\mathop{\mathbb{E}}_{x_0}\mathsf{CE}\left(Y(x_0,x), v(x_0+F(x))\right)\\ &\leq \frac{1}{v} + \mathop{\mathbf{Pr}}_{x_0}\left[x_0 \in -G^\star(x) \pm \left(|G^\star(x) - F(x)| + 10\frac{\log v}{v}\right)\right] \times O\left(1 + v|G^\star(x) - F(x)|\right)\\ &\leq \frac{1}{v} + \left(|G^\star(x) - F(x)| + 10\frac{\log v}{v}\right) \times O\left(1 + v|G^\star(x) - F(x)|\right)\end{aligned}$$

[37] The proof is rather simple. Denote by $\mathbb{E}[X^2] = a^2$ and let $\mathcal{E} = \{X \geq \frac{1}{2}a\}$ and $p = \mathbf{Pr}\left[X \geq \frac{1}{2}a\right]$. Then, we have

$$a^2 = \mathbb{E}[X^2] \leq \frac{1}{4}(1-p)a^2 + p\,\mathbb{E}[X^2 \mid \mathcal{E}] \leq \frac{1}{4}a^2 + p\sqrt{\mathbb{E}[X^4 \mid \mathcal{E}]} = \frac{1}{4}a^2 + \sqrt{p}\sqrt{p\,\mathbb{E}[X^4 \mid \mathcal{E}]} \leq \frac{1}{4}a^2 + \sqrt{p}\sqrt{\mathbb{E}[X^4]}$$

$$\leq \frac{1}{v} + O\left(\log v \times |G^\star(x) - F(x)| + v|G^\star(x) - F(x)|^2 + \frac{\log v}{v}\right)$$

Taking expectation over x we have

$$\mathop{\mathbb{E}}_{(x_0,x)\sim\mathcal{D}} \mathsf{CE}\left(Y(x_0,x), v(x_0 + F(x))\right)$$
$$\leq \frac{1}{v} + O\left(\log v \mathop{\mathbb{E}}_{x\sim\mathcal{D}} |G^\star(x) - F(x)| + v \mathop{\mathbb{E}}_{x\sim\mathcal{D}} |G^\star(x) - F(x)|^2 + \frac{\log v}{v}\right) \leq O(v\varepsilon^2 + \frac{\log^2 v}{v}) \ .$$

□

O LOWER BOUNDS FOR KERNELS, FEATURE MAPPINGS AND TWO-LAYER NETWORKS

O.1 LOWER BOUND: KERNEL METHODS AND FEATURE MAPPINGS

This subsection is a direct corollary of (Allen-Zhu & Li, 2019a) with simple modifications.

We consider the following L-layer target network as a separating hard instance for any kernel method. Let us choose $k = 1$ with each $\mathbf{W}^\star_{\ell,0}, \mathbf{W}^\star_{\ell,1} \in \mathbb{R}^d$ sampled i.i.d. uniformly at random from $\mathcal{S}_{2^{L-1}}$, and other $\mathbf{W}^\star_{\ell,j} = 1$. Here, the set $\mathcal{S}_p$ is given by:

$$\mathcal{S}_p = \left\{\forall w \in \mathbb{R}^d \mid \|w\|_0 = p, w_i \in \left\{0, \frac{1}{\sqrt{p}}\right\}\right\} \ .$$

We assume input x follows from the d-dimensional standard Gaussian distribution.

Recall Theorem 1 says that, for every d and $L = o(\log\log d)$, under appropriate gap assumptions for $\alpha_1, \dots, \alpha_L$, for every $\varepsilon > 0$, the neural network defined in our paper requires only $\mathsf{poly}(d/\varepsilon)$ time and samples to learn this target function $G^\star(x)$ up to accuracy ε.

In contrast, we show the following theorem of the sample complexity lower bound for kernel methods:

Theorem O.1 (kernel lower bound). *For every $d > 1$, every $L \leq \frac{\log\log d}{100}$, every $\alpha_L < 0.1$, every (Mercer) kernels $K : \mathbb{R}^{d\times d} \to \mathbb{R}$, and $N \leq \frac{1}{1000}\binom{d}{2^{L-1}}$, for every N i.i.d. samples $x^{(1)}, \dots, x^{(N)} \sim \mathcal{N}(0,1)$, the following holds for at least 99% of the target functions $G^\star(x)$ in the aforementioned class (over the choice in $\mathcal{S}_p$). For all kernel regression functions*

$$\mathfrak{K}(x) = \textstyle\sum_{n\in[N]} K(x, x^{(n)}) \cdot v_n$$

where weights $v_i \in \mathbb{R}$ can depend on $\alpha_1, \cdots, \alpha_L, x^{(1)}, \dots, x^{(N)}, K$ and the training labels $\{y^{(1)}, \cdots, y^{(N)}\}$, it must suffer population risk

$$\mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,\mathbf{I}_{d\times d})} \left(G^\star(x) - \mathfrak{K}(x)\right)^2 = \Omega(\alpha_L^2 \log^{-2^{L+2}}(d)) \ .$$

Remark O.2. Let us compare this to our positive result in Theorem 1 for $L = o(\log\log d)$. Recall from Section 3 that α_L can be as large as for instance $d^{-0.001}$ in order for Theorem 1 to hold. When this holds, neural network achieves for instance $1/d^{100}$ error with $\mathsf{poly}(d)$ samples and time complexity. In contrast, Theorem O.1 says, unless there are more than $\frac{1}{1000}\binom{d}{2^{L-1}} = d^{\omega(1)}$ samples, no kernel method can achieve a regression error of even $1/d^{0.01}$.

Sketch proof of Theorem O.1. The proof is almost a direct application of (Allen-Zhu & Li, 2019a), and the main difference is that we have Gaussian input distribution here (in order to match the upper bound), and in (Allen-Zhu & Li, 2019a) the input distribution is uniform over $\{-1,1\}^d$. We sketch the main ideas below.

First, randomly sample $|x_i|$ for each coordinate of x, then we have that $x_i = |x_i|\tau_i$ where each τ_i i.i.d. uniformly on $\{-1,1\}$. The target function $G^\star(x)$ can be re-written as $G^\star(x) = \widetilde{G^\star}(\tau)$ for

$\tau = (\tau_i)_{i\in[d]} \in \{-1,1\}^d$, where $\widetilde{G^\star}(\tau)$ is a degree $p = 2^{L-1}$ polynomial over τ, of the form:

$$\widetilde{G^\star}(\tau) = \alpha_L \langle w, \tau\rangle^p + \widehat{G^\star}(\tau)$$

where (for $a \circ b$ being the coordinate product of two vectors)

$$w = \mathbf{W}^\star_{2,0} \circ |x| \quad \text{and} \quad \deg(\widehat{G^\star}(\tau)) \le p-1$$

For every function f, let us write the Fourier Boolean decomposition of f:

$$f(\tau) = \sum_{\mathcal{S}\subset[d]} \lambda_{\mathcal{S}} \prod_{j\in\mathcal{S}} \tau_j$$

and for any fixed w, write the decomposition of $\widetilde{G^\star}(\tau)$:

$$\widetilde{G^\star}(\tau) = \sum_{\mathcal{S}\subset[d]} \lambda'_{\mathcal{S}} \prod_{j\in\mathcal{S}} \tau_j$$

Let us denote the set of p non-zero coordinates of $\mathbf{W}^\star_{2,0}$ as $\mathcal{S}_w$. Using basic Fourier analysis of boolean variables, we must have that conditioning on the ≥ 0.999 probability event that $\prod_{i\in\mathcal{S}_w}|x_i| \ge \left(\log^{0.9} d\right)^{-2^L}$, it satisfies

$$|\lambda'_{\mathcal{S}_w}| = \left(\frac{1}{\sqrt{p}}\right)^p \alpha_L \prod_{i\in\mathcal{S}_w}|x_i| \ge \left(\frac{1}{\sqrt{p}}\right)^p \alpha_L \left(\log^{0.9} d\right)^{-2^L} \ge \alpha_L \log^{-2^L}(d) \ .$$

Moreover, since $\deg(\widehat{G^\star}(\tau)) \le p-1$, we must have $\lambda'_{\mathcal{S}} = 0$ for any other $\mathcal{S} \ne \mathcal{S}_w$ with $|\mathcal{S}| = p$. This implies that for any function $f(\tau)$ with

$$f(\tau) = \sum_{\mathcal{S}\subset[d]} \lambda_{\mathcal{S}} \prod_{j\in\mathcal{S}} \tau_j \quad \text{and} \quad \mathbb{E}_\tau \left(f(\tau) - \widetilde{G^\star}(\tau)\right)^2 = O(\alpha_L^2 \log^{-2^{L+2}}(d)) \ ,$$

it must satisfy

$$\lambda^2_{\mathcal{S}_w} = \Omega(\alpha_L^2 \log^{-2^{L+1}}(d)) > \sum_{\mathcal{S}\subseteq[d], |\mathcal{S}|=p, \mathcal{S}\ne\mathcal{S}_w} \lambda^2_{\mathcal{S}} = O(\alpha_L^2 \log^{-2^{L+2}}(d))$$

Finally, using $\mathbb{E}_{x\sim\mathcal{N}(0,\mathbf{I})} \left(G^\star(x) - \mathfrak{K}(x)\right)^2 = \mathbb{E}_{|x|}\mathbb{E}_\tau \left(\mathfrak{K}(|x|\circ\tau) - \widetilde{G^\star}(\tau)\right)^2$, we have with probability at least 0.999 over the choice of $|x|$, it holds that

$$\mathbb{E}_\tau \left(\mathfrak{K}(|x|\circ\tau) - \widetilde{G^\star}(\tau)\right)^2 = O(\alpha_L^2 \log^{-2^{L+2}}(d)) \ .$$

From here, we can select $f(\tau) = \mathfrak{K}(|x|\circ\tau)$. The rest of the proof is a direct application of (Allen-Zhu & Li, 2019a, Lemma E.2) (as the input τ is now uniform over the Boolean cube $\{-1,1\}^d$). (The precise argument also uses the observation that if for > 0.999 fraction of w, event $\mathcal{E}_w(x)$ holds for > 0.999 fraction of x, then there is an x such that $\mathcal{E}_w(x)$ holds for > 0.997 fraction of w.) $\square$

For similar reason, we also have the number of features lower bound for linear regression over feature mappings:

Theorem O.3 (feature mapping lower bound)**.** *For every $d > 1$, every $L \le \frac{\log\log d}{100}$, every $d \ge 0$, every $\alpha_L \le 0.1$, every $D \le \frac{1}{1000}\binom{d}{2^{L-1}}$, and every feature mapping $\phi\colon \mathbb{R}^d \to \mathbb{R}^D$, the following holds for at least 99% of the target functions $G^\star(x)$ in the aforementioned class (over the choice in $\mathcal{S}_p$). For all linear regression functions*

$$\mathfrak{F}(x) = w^\top \phi(x),$$

where weights $w \in \mathbb{R}^D$ can depend on $\alpha_1, \cdots, \alpha_L$ and ϕ, it must suffer population risk

$$\mathbb{E}_{x\sim\mathcal{N}(0,\mathbf{I})} \|G^\star(x) - \mathfrak{F}(x)\|_2^2 = \Omega\left(\alpha_L^2 \log^{-2^{L+2}}(d)\right) \ .$$

Remark O.4. In the same setting as Remark O.2, we see that neural network achieves for instance $1/d^{100}$ regression error with poly(d) time complexity, but to achieve even just $1/d^{0.01}$ error, Theorem O.3 says that any linear regression over feature mappings must use at least $D = d^{\omega(1)}$ features. This usually needs $\Omega(D) = d^{\omega(1)}$ time complexity.[38]

O.2 Lower Bound: Certain Two-Layer Polynomial Neural Networks

We also give a preliminary result separating our positive result (for L-layer quadratic DenseNet) from two-layer neural networks with polynomial activations (of degree 2^L). The lower bound relies on the following technical lemma which holds for some absolute constant $C > 1$:

Lemma O.5. *For $1 \leq d_1 \leq d$, consider inputs (x, y) where $x \in \mathbb{R}^{d_1}$ follows from $\mathcal{N}(0, \mathbf{I}_{d_1 \times d_1})$ and $y \in \mathbb{R}^{d-d_1}$ follows from an arbitrary distribution independent of x. We have that for every $p \geq 1$,*

- *for every function $f(x, y) = \left(\frac{\|x\|_4^4}{d_1}\right)^p + g(x, y)$ where $g(x, y)$ is a polynomial and its degree over x is at most $4p - 1$, and*
- *for every function $h(x, y) = \sum_{i=1}^r a_i \widetilde{\sigma}_i(\langle w_i, (x, x^2, y) + b_i \rangle)$ with $r = \frac{1}{C}(d_1/p)^p$ and each $\widetilde{\sigma}_i$ is an arbitrary polynomial of maximum degree $2p$,*

it must satisfy $\mathbb{E}_{x,y}(h(x, y) - f(x, y))^2 \geq \frac{1}{p^{C \cdot p}}$.

Before we prove Lemma O.5 in Section O.2.1, let us quickly point out how it gives our lower bound theorem. We can for instance consider target functions with $k_2 = d$, $k_3 = \cdots = k_L = 1$, $\mathbf{W}^\star_{2,1} = \mathbf{I}_{d \times d}$ and $\mathbf{W}^\star_{\ell,0}, \mathbf{W}^\star_{\ell,1}, \mathbf{W}^\star_{\ell,2} = \left(\frac{1}{\sqrt{d}}, \cdots, \frac{1}{\sqrt{d}}\right)$, and other $\mathbf{W}^\star_{\ell,j} = 1$ for $j > 2$.

For such target functions, when $L = o(\log \log d)$, our positive result Theorem 1 shows that the (hierarchical) DenseNet learner considered in our paper only need $\mathsf{poly}(d/\varepsilon)$ time and sample complexity to learn it to an arbitrary $\varepsilon > 0$ error (where the degree of the $\mathsf{poly}(d/\varepsilon)$ does not depend on L).

On the other hand, since the aforementioned target $G^\star(x)$ can be written in the form $\alpha_L \left(\frac{\|x\|_4^4}{d_1}\right)^{2^{L-2}} + g(x)$ for some $g(x)$ of degree at most $2^L - 1$, Lemma O.5 directly implies the following:

Theorem O.6. *For any two-layer neural network of form $h(x) = \sum_{i=1}^r a_i \widetilde{\sigma}_i(\langle w_i, (x, S_1(x)) + b_i \rangle)$, with $r \leq d^{2^{o(L)}}$ and each $\widetilde{\sigma}_i$ is any polynomial of maximum degree 2^{L-1}, we have that*

$$\mathop{\mathbb{E}}_{x \sim \mathcal{N}(0,\mathbf{I})} (h(x) - G^\star(x))^2 \geq \frac{\alpha_L^2}{2^{2^{O(L)}}} \ .$$

(Since $\widetilde{\sigma}_i$ is degree 2^{L-1} over $S_1(x)$, the final degree of $h(x)$ is 2^L in x; this is the same as our L-layer DenseNet in the positive result.)

To compare this with the upper bound, let us recall again (see Section 3) that when $L = o(\log \log d)$, parameter α_L can be as large as for instance $d^{-0.001}$ in order for Theorem 1 to hold. When this holds, neural network achieves for instance $1/d^{100}$ error with $\mathsf{poly}(d)$ samples and time complexity. In contrast, Theorem O.1 says, unless there are more than $d^{2^{\Omega(L)}} = d^{\omega(1)}$ neurons, the two-layer polynomial network cannot achieve regression error of even $1/d^{0.01}$. To conclude, the hierarchical neural network can learn this function class more efficiently.

Finally, we also remark here after some simple modifications to Lemma O.5, we can also obtain the following theorem when $k_2 = k_3 = \cdots = k_L = 1$, $\mathbf{W}^\star_{\ell,1}, \mathbf{W}^\star_{\ell,0} = \left(\frac{1}{\sqrt{d}}, \cdots, \frac{1}{\sqrt{d}}\right)$ and other

[38] One might argue that feature mapping can be implemented to run faster than $O(D)$ time. However, those algorithms are very complicated and may require a lot of work to design. It can be unfair to compare to them for a "silly" reason. One can for instance cheat by defining an infinitely-large feature mapping where each feature corresponds to a different neural network; then, one can train a neural network and just set the weight of the feature mapping corresponding to the final network to be 1. Therefore, we would tend to assume that a linear regression over feature mapping requires at least $\Omega(D)$ running time to implement, where D is the total number of features.

$\mathbf{W}^\star_{\ell,j} = 1$.

Theorem O.7. *For every function of form $h(x) = \sum_{i=1}^r a_i \widetilde{\sigma}'_i(\langle w_i, x + b_i\rangle)$ with $r \leq d^{2^{o(L)}}$ and each $\widetilde{\sigma}'_i$ is any polynomial of maximum degree 2^L, we have*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{N}(0,\mathbf{I})} (h(x) - G^\star(x))^2 \geq \frac{\alpha_L^2}{2^{2^{O(L)}}} \enspace .$$

O.2.1 Proof of Lemma O.5

Proof of Lemma O.5. Suppose by way of contradiction that for some sufficiently large constant $C > 1$,

$$\mathop{\mathbb{E}}_{x,y}(h(x,y) - f(x,y))^2 \leq \frac{1}{p^{C\cdot p}}$$

This implies that

$$\mathop{\mathbb{E}}_{x}\left(\mathop{\mathbb{E}}_{y} h(x,y) - \mathop{\mathbb{E}}_{y} f(x,y)\right)^2 \leq \frac{1}{p^{C\cdot p}} \tag{O.1}$$

We break x into p parts: $x = (x^{(1)}, x^{(2)}, \cdots, x^{(p)})$ where each $x^{(j)} \in \mathbb{R}^{d_1/p}$. We also decompose w_i into $(w_i^{(1)}, w_i^{(2)}, \cdots, w_i^{(p)}, w'_i)$ accordingly. We can write

$$\left(\frac{\|x\|_4^4}{d_1}\right)^p = \left(\frac{\sum_{j\in[p]}\|x^{(j)}\|_4^4}{d_1}\right)^p \tag{O.2}$$

Since $\widetilde{\sigma}_i$ is of degree at most $2p$, we can write for some coefficients $a_{i,q}$:

$$\mathop{\mathbb{E}}_{y} a_i\widetilde{\sigma}_i(\langle w_i, (x, x^2, y) + b_i\rangle) = \sum_{q\in[2p]} a_{i,q}\left(\sum_{j\in[p]}\langle x^{(j)}, w_i^{(j)}\rangle + \langle \left(x^{(j)}\right)^2, w_i^{(j)}\rangle\right)^q \tag{O.3}$$

Let us now go back to (O.1). We know that $\mathbb{E}_y f(x,y)$ and $\mathbb{E}_y h(x,y)$ are both polynomials over $x \in \mathbb{R}^{d_1}$ with maximum degree $4p$.

- The only $4p$-degree monomials of $\mathbb{E}_y f(x,y)$ come from (O.2) which is $\frac{1}{(d_1)^p}\left(\sum_{j\in[p]}\|x^{(j)}\|_4^4\right)^p$. Among them, the only ones with homogeneous degree 4 for each $x^{(j)}$ is $\frac{1}{(d_1)^p}\prod_{j\in[p]}\|x^{(j)}\|_4^4$.
- The only $4p$-degree monomials of $\mathbb{E}_y h(x,y)$ come from (O.3) which is $a_{i,2p}\left(\sum_{j\in[p]}\langle\left(x^{(j)}\right)^2, w_i^{(j)}\rangle\right)^{2p}$. Among them, the only ones with homogeneous degree 4 for each $x^{(j)}$ can be written as $\frac{a'_i}{(d_1)^p}\prod_{j\in[p]}\left(\langle\left(x^{(j)}\right)^2, w_i^{(j)}\rangle\right)^2$.

Applying the degree-preserving Property D.4 for Gaussian polynomials:

$$\mathcal{C}_x\left(\sum_i a'_i\prod_{j\in[p]}\langle\left(x^{(j)}\right)^2, w_i^{(j)}\rangle^2 - \prod_{j\in[p]}\|x^{(j)}\|_4^4\right) \leq \frac{(d_1)^{2p}}{p^{(C-10)p}} \enspace .$$

Let us denote $\prod_{j\in[p]}\langle\left(x^{(j)}\right)^2, w_i^{(j)}\rangle = \langle\widetilde{w}_i, \widetilde{x}\rangle$ where $\widetilde{x}, \widetilde{w}_i \in \mathbb{R}^{(d_1/p)^p}$ are given as:

$$\widetilde{x} = \left(\prod_{j\in[p]}\left(x_{i_j}^{(j)}\right)^2\right)_{i_1,\cdots,i_p\in[d_1/p]} \quad\text{and}\quad \widetilde{w}_i = \left(\prod_{j\in[p]}[w_i^{(j)}]_{i_j}\right)_{i_1,\cdots,i_p\in[d_1/p]}$$

Under this notation, we have

$$\prod_{j\in[p]}\|x^{(j)}\|_4^4 = \|\widetilde{x}\|_2^2, \quad \sum_i a'_i\prod_{j\in[p]}\langle\left(x^{(j)}\right)^2, w_i^{(j)}\rangle^2 = \widetilde{x}^\top\sum_i a'_i\widetilde{w}_i(\widetilde{w}_i)^\top\widetilde{x}^\top$$

This implies that for $\mathbf{M} = \sum_i a'_i \widetilde{w}_i (\widetilde{w}_i)^\top \in \mathbb{R}^{(d_1/p)^p \times (d_1/p)^p}$, we have

$$\mathcal{C}_x \left(\widetilde{x}^\top (\mathbf{M} - \mathbf{I}) \widetilde{x}^\top \right) = \frac{(d_1)^{2p}}{p^{(C-10)p}}$$

By the special structure of $\mathbf{M}$ where $\mathbf{M}_{(i_1, i'_1), (i_2, i'_2), \cdots, (i_j, i'_j)} = \mathbf{M}_{\{i_1, i'_1\}, \{i_2, i'_2\}, \cdots, \{i_j, i'_j\}}$ does not depend on the order of (i_j, i'_j) (since each $\widetilde{w}_i (\widetilde{w}_i)^\top$ has this property), we further know that

$$\|\mathbf{I} - \mathbf{M}\|_F^2 = \frac{(d_1)^{2p}}{p^{(C-10)p}} \ll (d_1/p)^p \times (d_1/p)^p$$

This implies that the rank r of $\mathbf{M}$ must satisfy $r = \Omega((d_1/p)^p)$ using (Allen-Zhu & Li, 2019a, Lemma E.2). $\square$

P MATHEMATICAL PRELIMINARIES

P.1 CONCENTRATION OF GAUSSIAN POLYNOMIALS

Lemma P.1. *Suppose $f \colon \mathbb{R}^m \to \mathbb{R}$ is a degree q homogenous polynomial, and let $\mathcal{C}(f)$ be the sum of squares of all the monomial coefficients of f. Suppose $g \sim \mathcal{N}(0, \mathbf{I})$ is standard Gaussian, then for every $\varepsilon \in (0, \frac{1}{10})$,*

$$\Pr_{g \sim \mathcal{N}(0, \mathbf{I})} \left[|f(g)| \leq \varepsilon \sqrt{\mathcal{C}(f)} \right] \leq O(q) \cdot \varepsilon^{1/q}$$

Proof. Recall from the anti-concentration of Gaussian polynomial (see Lemma P.2a)

$$\Pr_{g \sim \mathcal{N}(0, \mathbf{I})} \left[|f(g) - t| \leq \varepsilon \sqrt{\mathbf{Var}[f(g)]} \right] \leq O(q) \cdot \varepsilon^{1/q}$$

Next, one can verify when f is degree-q homogenous for $q \geq 1$, we have $\mathbf{Var}[f(g)] \geq \mathcal{C}(f)$. This can be seen as follows, first, we write $\mathbf{Var}[f(g)] = \mathbb{E}[(f(g) - \mathbb{E} f(g))^2]$. Next, we rewrite the polynomial $f(g) - \mathbb{E} f(g)$ in the Hermite basis of g. For instance, $g_1^5 g_2^2$ is replaced with $(H_5(g_1) + \cdots)(H_2(g_2) + \cdots)$ where $H_k(x)$ is the (probabilists') k-th order Hermite polynomial and the "$\cdots$" hides lower-order terms. This transformation does not affect the coefficients of the highest degree monomials. (For instance, the coefficient in front of $H_5(g_1) H_2(g_2)$ is the same as the coefficient in front of $g_1^5 g_2^2$. By the orthogonality of Hermite polynomials with respect to the Gaussian distribution, we immediately have $\mathbb{E}[(f(g) - \mathbb{E} f(g))^2] \geq \mathcal{C}(f)$. $\square$

Lemma P.2. *Let $f \colon \mathbb{R}^m \to \mathbb{R}$ be a degree q polynomial.*

(a) Anti-concentration (see e.g. (Lovett, 2010, Eq. (1))): for every $t \in \mathbb{R}$ and $\varepsilon \in (0, 1)$,

$$\Pr_{g \sim \mathcal{N}(0, \mathbf{I})} \left[|f(g) - t| \leq \varepsilon \sqrt{\mathbf{Var}[f(g)]} \right] \leq O(q) \cdot \varepsilon^{1/q}$$

(b) Hypercontractivity concentration (see e.g. (Schudy & Sviridenko, 2012, Thm 1.9)): there exists constant $R > 0$ so that

$$\Pr_{g \sim \mathcal{N}(0, \mathbf{I})} \left[|f(g) - \mathbb{E}[f(g)]| \geq \lambda \right] \leq e^2 \cdot e^{-\left(\frac{\lambda^2}{R \cdot \mathbf{Var}[f(g)]} \right)^{1/q}}$$

P.2 RANDOM INITIALIZATION

Lemma I.6. *Let $\mathbf{R}_\ell \in \mathbb{R}^{\binom{k_\ell+1}{2} \times k_\ell}$ be a random matrix such that each entry is i.i.d. from $\mathcal{N}\left(0, \frac{1}{k_\ell^2}\right)$, then with probability at least $1-p$, $\mathbf{R}_\ell * \mathbf{R}_\ell$ has singular values between $[\frac{1}{O(k_\ell^4 p^2)}, O(1 + \frac{1}{k_\ell^2} \log \frac{k_\ell}{p})]$, and $\|\mathbf{R}_\ell\|_2 \leq O(1 + \frac{\sqrt{\log(1/p)}}{k_\ell})$.*

As a result, with probability at least 0.99, it satisfies for all $\ell = 2, 3, \ldots, L$, the square matrices $\mathbf{R}_\ell * \mathbf{R}_\ell$ have singular values between $[\frac{1}{O(k_\ell^4 L^2)}, O(1 + \frac{\log(L k_\ell)}{k_\ell})]$ and $\|\mathbf{R}_\ell\|_2 \leq O(1 + \frac{\sqrt{\log L}}{k_\ell})$.

Proof. Let us drop the subscript ℓ for simplicity, and denote by $m = \binom{k+1}{2}$. Consider any unit vector $u \in \mathbb{R}^m$. Define $v^{(i)}$ to (any) unit vector orthogonal to all the rows of $\mathbf{R}$ except its i-th row. We have

$$|u^\top(\mathbf{R} * \mathbf{R})v^{(i)}| = |u_i(\mathbf{R}_{i,:} * \mathbf{R}_{i,:})v^{(i)}| = |u_i| \left| \sum_{p \leq q} a_{p,q} \mathbf{R}_{i,p} \mathbf{R}_{i,q} v^{(i)}_{p,q} \right|$$

Now, we have that $v^{(i)}$ is independent of the randomness of $\mathbf{R}_{i,:}$, and therefore, by anti-concentration of Gaussian homogenous polynomials (see Lemma P.1),

$$\mathop{\mathbf{Pr}}_{\mathbf{R}_{i,:}} \left[\left| \sum_{p \leq q} a_{p,q} \mathbf{R}_{i,p} \mathbf{R}_{i,q} v^{(i)}_{p,q} \right| \leq \varepsilon \|v^{(i)}\| \cdot \frac{1}{k} \right] \leq O(\varepsilon^{1/2}) \ .$$

Therefore, given any fixed i, with probability at least $1 - O(\varepsilon^{1/2})$, it satisfies that for *every* unit vector u ,

$$|u^\top(\mathbf{R} * \mathbf{R})v^{(i)}| \geq \frac{\varepsilon}{k}|u_i| \ .$$

By union bound, with probability at least $1 - O(k\varepsilon^{1/2})$, the above holds for all i and all unit vectors u. Since $\max_i |u_i| \geq \frac{1}{k}$ for any unit vector $u \in \mathbb{R}^{\binom{k+1}{2}}$, we conclude that $\sigma_{\min}(\mathbf{R} * \mathbf{R}) \geq \frac{\varepsilon}{k^2}$ with probability at least $1 - O(k\varepsilon^{1/2})$.

As for the upper bound, we can do a crude calculation by using $\|\mathbf{R} * \mathbf{R}|_2 \leq \|\mathbf{R} * \mathbf{R}\|_F$.

$$\|\mathbf{R} * \mathbf{R}\|_F^2 = \sum_{i, p \leq q} a_{p,q}^2 \mathbf{R}_{i,p}^2 \mathbf{R}_{i,q}^2 = \sum_i \left(\sum_{p \in [k]} \mathbf{R}_{i,p}^2 \right)^2 \ .$$

By concentration of chi-square distribution (and union bound), we know that with probability at least $1 - p$, the above summation is at most $O(k^2) \cdot (\frac{1}{k} + \frac{\log(k/p)}{k^2})^2$.

Finally, the bound on $\|\mathbf{R}\|_2$ can be derived from any asymptotic bound for the maximum singular value of Gaussian random matrix: $\mathbf{Pr}[\|k\mathbf{R}\|_2 > tk] \leq e^{-\Omega(t^2 k^2)}$ for every $t \geq \Omega(1)$. □

P.3 Property on Symmetric Tensor

Lemma I.5. *If $\mathbf{U} \in \mathbb{R}^{p \times p}$ is unitary and $\mathbf{R} \in \mathbb{R}^{s \times p}$ for $s \geq \binom{p+1}{2}$, then there exists some* unitary *matrix $\mathbf{Q} \in \mathbb{R}^{\binom{p+1}{2} \times \binom{p+1}{2}}$ so that $\mathbf{RU} * \mathbf{RU} = (\mathbf{R} * \mathbf{R})\,\mathbf{Q}$.*

Proof of Lemma I.5. For an arbitrary vector $w \in \mathbb{R}^s$, let us denote by $w^\top(\mathbf{R} * \mathbf{R}) = (b_{i,j})_{1 \leq i \leq j \leq p}$. Let $g \in \mathcal{N}(0, \mathbf{I}_{p \times p})$ be a Gaussian random vector so we have:

$$w^\top \sigma(\mathbf{R}g) = \sum_{i \in [s]} w_i (\mathbf{R}_i g)^2 = \sum_{i \in [s]} w_i \langle \mathbf{R}_i * \mathbf{R}_i, g * g \rangle = \sum_{i \in [p]} b_{i,i} g_i^2 + \sqrt{2} \sum_{1 \leq i < j \leq p} b_{i,j} g_i g_j \ .$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\left(w^\top \sigma(\mathbf{R}g) \right)^2 \right] &= \mathbb{E}\left[\left(\textstyle\sum_{i \in [p]} b_{i,i} g_i^2 + \sqrt{2} \sum_{1 \leq i < j \leq p} b_{i,j} g_i g_j \right)^2 \right] \\
&= 2 \textstyle\sum_{1 \leq i < j \leq p} b_{i,j}^2 + 2 \sum_{1 \leq i < j \leq p} b_{i,i} b_{j,j} + 3 \sum_{i \in [p]} b_{i,i}^2 \\
&= 2 \textstyle\sum_{1 \leq i < j \leq p} b_{i,j}^2 + \left(\sum_{i \in [p]} b_{i,i} \right)^2 + 2 \sum_{i \in [p]} b_{i,i}^2 \ .
\end{aligned}$$

On the other hand, we have $\mathbb{E}\left[w^\top \sigma(\mathbf{R}g) \right] = \sum_{i \in [p]} b_{i,i}$. Therefore, we have

$$\mathbf{Var}\left[w^\top \sigma(\mathbf{R}g) \right] = 2 \|w^\top (\mathbf{R} * \mathbf{R})\|_2^2 \ .$$

Note that $\mathbf{Var}[w^\top \sigma(\mathbf{R}g)] = \mathbf{Var}[w^\top \sigma(\mathbf{RU}g)]$ for a unitary matrix $\mathbf{U}$, therefore we conclude that

$$\|w^\top(\mathbf{RU} * \mathbf{RU})\|_2^2 = \|w^\top(\mathbf{R} * \mathbf{R})\|_2^2$$

for any vector w. Which implies that there exists some *unitary* matrix $\mathbf{Q} \in \mathbb{R}^{\binom{p+1}{2} \times \binom{p+1}{2}}$ so that $\mathbf{RU} * \mathbf{RU} = (\mathbf{R} * \mathbf{R})\,\mathbf{Q}$. $\square$

P.4 Properties On Homogeneous Polynomials

Given any degree-q homogenous polynomial $f(x) = \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I \prod_{j \in [n]} x_j^{I_j}$, recall we have defined

$$\mathcal{C}_x(f) := \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I^2$$

When it is clear from the context, we also denote $\mathcal{C}(f) = \mathcal{C}_x(f)$.

Definition P.3. *Given $f \colon \mathbb{R}^n \to \mathbb{R}$ and vector $y \in \mathbb{R}^n$, define the directional derivative*

$$(\Delta_y f)(x) := f(x+y) - f(x)$$

and given vectors $y^{(1)}, \dots, y^{(q)} \in \mathbb{R}^n$, define $\Delta_{y^{(1)},\dots,y^{(q)}} f = \Delta_{y^{(1)}} \Delta_{y^{(2)}} \dots \Delta_{y^{(q)}}$.

Lemma P.4. *Suppose $f \colon \mathbb{R}^n \to \mathbb{R}$ is a degree-q homogeneous polynomial. Then, the finite-differentiate polynomial*

$$\widehat{f}(y^{(1)}, \dots, y^{(q)}) = \Delta_{y^{(1)},\dots,y^{(q)}} f(x)$$

is also degree-q homogenous over $n \times q$ variables, and satisfies

- $\mathcal{C}(f) \cdot q! \le \mathcal{C}(\widehat{f}) \le \mathcal{C}(f) \cdot (q!)^2$.
- $\mathbb{E}_{y^{(1)},\dots,y^{(q)} \sim \mathcal{N}(0, \mathbf{I}_{n \times n})}[\big(\widehat{f}(y^{(1)}, \dots, y^{(q)})\big)^2] = \mathcal{C}(\widehat{f})$

Proof. Suppose $f(x) = \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I \prod_{j \in [n]} x_j^{I_j}$. Then, we have (see (Lovett, 2010, Claim 3.2))

$$\widehat{f}(y^{(1)}, \dots, y^{(q)}) = \sum_{J \in [n]^q} \widehat{a}_J \prod_{j \in [q]} y_{J_j}^{(j)}$$

where $\widehat{a}_J = a_{I(J)} \cdot \prod_{k=1}^n (I_k(J))!$ and $I_k(J) = |\{j \in [q] \colon J_j = k\}|$.

On the other hand, for every $I^* \in \mathbb{N}^q$ with $\|I^*\|_1 = q$, there are $\frac{q!}{\prod_{k=1}^n (I_k^*)!}$ different choices of $J \in [n]^q$ that maps $I(J) = I^*$. Therefore, we have

$$\mathcal{C}(\widehat{f}) = \sum_{J \in [n]^q} \widehat{a}_J^2 = \sum_{J \in [n]^q} a_{I(J)}^2 \cdot \big(\prod_{k=1}^n (I_k(J))!\big)^2 = \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I^2 \cdot \big(\prod_{k=1}^n (I_k)!\big)^2 \cdot \frac{q!}{\prod_{k=1}^n (I_k)!}$$

As a result,

$$\sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I^2 \cdot (q!) \le \mathcal{C}(\widehat{f}) \le \sum_{I \in \mathbb{N}^n \colon \|I\|_1 = q} a_I^2 \cdot (q!)^2$$

As for the second bullet, it is simple to verify. $\square$

Lemma P.5. *Suppose $f \colon \mathbb{R}^n \to \mathbb{R}$ is a degree-q homogeneous polynomial.*

- *If $g(x) = f(\mathbf{U}x)$ for $\mathbf{U} \in \mathbb{R}^{n \times m}$ being row orthonormal (with $n \le m$), then $\mathcal{C}(g) \ge \frac{\mathcal{C}(f)}{q!}$.*
- *If $g(x) = f(\mathbf{W}x)$ for $\mathbf{W} \in \mathbb{R}^{n \times m}$ with $n \le m$ and $\sigma_{\min}(\mathbf{W}) \ge \frac{1}{\kappa}$, then $\mathcal{C}(g) \ge \frac{\mathcal{C}(f)}{(q!)^2 \kappa^q}$.*

Proof.

- For every $y^{(1)}, \dots, y^{(q)} \in \mathbb{R}^m$,

$$\widehat{g}(y^{(1)}, \dots, y^{(q)}) = \Delta_{y^{(1)},\dots,y^{(q)}} g(x) = \Delta_{\mathbf{U}y^{(1)},\dots,\mathbf{U}y^{(q)}} f(\mathbf{U}x) = \widehat{f}(\mathbf{U}y^{(1)}, \dots, \mathbf{U}y^{(q)})$$

Since Gaussian is invariant under orthonormal transformation, we have

$$\mathcal{C}(\widehat{f}) = \mathop{\mathbb{E}}_{y^{(1)},\dots,y^{(q)}\sim\mathcal{N}(0,\mathbf{I}_{n\times n})}[\big(\widehat{f}(y^{(1)},\dots,y^{(q)})\big)^2] = \mathop{\mathbb{E}}_{y^{(1)},\dots,y^{(q)}\sim\mathcal{N}(0,\mathbf{I}_{m\times m})}[\big(\widehat{f}(\mathbf{U}y^{(1)},\dots,\mathbf{U}y^{(q)})\big)^2] = \mathcal{C}(\widehat{g})$$

- Suppose $\mathbf{W} = \mathbf{U\Sigma V}$ is its SVD decomposition. Define $f_1(x) = f(\mathbf{U}x)$, $f_2(x) = f_1(\mathbf{\Sigma}x)$, so that $g(x) = f_2(\mathbf{V}x)$. We have $\mathcal{C}(g) \geq \frac{1}{q!}\mathcal{C}(f_2) \geq \frac{1}{q!\kappa^q}\mathcal{C}(f_1) \geq \frac{1}{(q!)^2\kappa^q}\mathcal{C}(f)$.

□

Lemma P.6. *Suppose $f, g\colon \mathbb{R}^n \to \mathbb{R}$ are two homogeneous polynomials of degree p and q respectively, and denote by $h(x) = f(x)g(x)$. Then $\mathcal{C}_x(h) \leq \binom{p+q}{p}\mathcal{C}_x(f)\mathcal{C}_x(g)$.*

Proof. Let us write

$$f(x) = \sum_{I\in\mathbb{N}^k\colon \|I\|_1=p} a_I \prod_{j\in[k]} x_j^{I_j} \quad\text{and}\quad g(x) = \sum_{J\in\mathbb{N}^k\colon \|J\|_1=q} b_J \prod_{j\in[k]} x_j^{J_j} \ .$$

On one hand, we obviously have $\sum_{I\in\mathbb{N}^k\colon \|I\|_1=p}\sum_{J\in\mathbb{N}^k\colon \|J\|_1=q} a_I^2 b_J^2 = \mathcal{C}(f)\mathcal{C}(g)$. On the other hand, when multiplied together, each monomial in the multiplication $f(x)g(x)$ comes from at most $\binom{p+q}{p}$ pairs of (I, J). If we denote this set as S, then

$$\Big(\sum_{(I,J)\in S} a_I b_J\Big)^2 \leq \binom{p+q}{p}\sum_{(I,J)\in S} a_I^2 b_J^2 \ .$$

Putting the two together finishes the proof. □

Lemma P.7. *Suppose $f^{(1)}, f^{(2)}\colon \mathbb{R}^n \to \mathbb{R}^k$ are degree-p homogeneous polynomials and $g\colon \mathbb{R}^k \to \mathbb{R}$ is degree q homogenous. Denote by $h(x) = g(f^{(1)}(x)) - g(f^{(2)}(x))$. Then,*

$$\mathcal{C}_x(h) \leq k^q q^2 \cdot 2^{q-1}\cdot\binom{qp}{p,p,\dots,p}\cdot\mathcal{C}(g)\cdot(\max_i \mathcal{C}(f_i^{(1)}-f_i^{(2)}))\cdot(\max_i \mathcal{C}(f_i^{(1)})+\max_i \mathcal{C}(f_i^{(1)}-f_i^{(2)}))^{q-1} \ .$$

Proof. Let us write

$$g(y) = \sum_{I\in\mathbb{N}^k\colon \|I\|_1=q} a_I \prod_{j\in[k]} y_j^{I_j} \ .$$

For each monomial above, we need to bound $\mathcal{C}_x(h_I(x))$ for each

$$h_I(x) := \prod_{j\in[k]}(f_j^{(1)}(x))^{I_j} - \prod_{j\in[k]}(f_j^{(2)}(x))^{I_j} = \prod_{j\in S} f_j^{(1)}(x) - \prod_{j\in S} f_j^{(2)}(x)$$

where $S \subset [k]$ is a multiset that contains exactly I_j copies of j. Using the identity that $a_1a_2a_3a_4 - b_1b_2b_3b_4 = (a_1-b_1)a_2a_3a_4 + b_1(a_2-b_2)a_3a_4 + b_1b_2(a_3-b_3)a_4 + b_1b_2b_3(a_4-b_4)$, as well as applying Lemma P.6, one can derive that

$$\begin{aligned}
\mathcal{C}_x(h_I) &\leq q^2\cdot\binom{qp}{p,p,\dots,p}\cdot(\max_i \mathcal{C}(f_i^{(1)}-f_i^{(2)}))\cdot(\max_i\{\mathcal{C}(f_i^{(1)}),\mathcal{C}(f_i^{(2)})\})^{q-1}\\
&\leq q^2\cdot 2^{q-1}\cdot\binom{qp}{p,p,\dots,p}\cdot(\max_i \mathcal{C}(f_i^{(1)}-f_i^{(2)}))\cdot(\max_i \mathcal{C}(f_i^{(1)})+\max_i \mathcal{C}(f_i^{(1)}-f_i^{(2)}))^{q-1}
\end{aligned}$$

Summing up over all monomials finishes the proof. □

P.5 Properties on Matrix Factorization

Claim P.8. *Suppose we have matrices $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{k\times m_1}$ and $\mathbf{B}, \mathbf{D} \in \mathbb{R}^{k\times m_2}$ for some $m_1, m_2 \geq k$ and $\|\mathbf{A}^\top\mathbf{B} - \mathbf{C}^\top\mathbf{D}\|_F \leq \varepsilon$. Then, there exists some matrix $\mathbf{P} \in \mathbb{R}^{k\times k}$ so that:*

- $\|\mathbf{A}^\top - \mathbf{C}^\top\mathbf{P}\|_F \leq \frac{\varepsilon}{\sigma_{\min}(\mathbf{B})}$,
- $\|\mathbf{B} - \mathbf{P}^{-1}\mathbf{C}\|_F \leq \frac{2\varepsilon\cdot(\sigma_{\max}(\mathbf{B}))^2}{\sigma_{\min}(\mathbf{B})\sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{D})}$, *and*

- *the singular values of* $\mathbf{P}$ *are within* $\left[\frac{\sigma_{\min}(\mathbf{D})}{\sigma_{\max}(\mathbf{B})}, \frac{\sigma_{\max}(\mathbf{D})}{\sigma_{\min}(\mathbf{B})}\right]$.

Proof of Claim P.8. We also refer to (Allen-Zhu & Li, 2016) for the proof.

Suppose $\mathbf{A} = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1$, $\mathbf{B} = \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2$, $\mathbf{C} = \mathbf{U}_3\mathbf{\Sigma}_3\mathbf{V}_3$, $\mathbf{D} = \mathbf{U}_4\mathbf{\Sigma}_4\mathbf{V}_4$ are the SVD decompositions. We can write

$$\begin{aligned} &\|\mathbf{V}_1^\top\mathbf{\Sigma}_1^\top\mathbf{U}_1^\top\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2 - \mathbf{V}_3^\top\mathbf{\Sigma}_3^\top\mathbf{U}_3^\top\mathbf{U}_4\mathbf{\Sigma}_4\mathbf{V}_4\|_F \le \varepsilon \\ \implies &\|\mathbf{V}_3\mathbf{V}_1^\top\mathbf{\Sigma}_1^\top\mathbf{U}_1^\top\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2\mathbf{V}_4^\top - \mathbf{\Sigma}_3^\top\mathbf{U}_3^\top\mathbf{U}_4\mathbf{\Sigma}_4\|_F \le \varepsilon \end{aligned}$$

Now note that $\mathbf{\Sigma}_3^\top\mathbf{U}_3^\top\mathbf{U}_4\mathbf{\Sigma}_4$ is of dimension $m_1 \times m_2$ and only its top left $k \times k$ block is non-zero. Let us write $\mathbf{\Sigma}_4 = (\overline{\mathbf{\Sigma}}_4, \mathbf{0})$ for $\overline{\mathbf{\Sigma}}_4 \in \mathbb{R}^{k\times k}$. Let us write $\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2\mathbf{V}_4^\top = (\mathbf{E}, \mathbf{F})$ for $\mathbf{E} \in \mathbb{R}^{k\times k}$. Then, the above Frobenius bound also implies (by ignoring the last $m_2 - k$ columns)

$$\|\mathbf{V}_3\mathbf{V}_1^\top\mathbf{\Sigma}_1^\top\mathbf{U}_1^\top\mathbf{E} - \mathbf{\Sigma}_3^\top\mathbf{U}_3^\top\mathbf{U}_4\overline{\mathbf{\Sigma}}_4\|_F \le \varepsilon$$

Finally, using $\|\mathbf{M}\mathbf{N}\|_F \le \|\mathbf{M}\|_F \cdot \sigma_{\max}(\mathbf{N})$, we have

$$\|\mathbf{V}_1^\top\mathbf{\Sigma}_1^\top\mathbf{U}_1^\top - \mathbf{V}_3^\top\mathbf{\Sigma}_3^\top\mathbf{U}_3^\top\mathbf{U}_4\overline{\mathbf{\Sigma}}_4\mathbf{E}^{-1}\|_F \le \frac{\varepsilon}{\sigma_{\min}(\mathbf{E})} = \frac{\varepsilon}{\sigma_{\min}(\mathbf{B})}$$

Let us define $\mathbf{P} = \mathbf{U}_4\overline{\mathbf{\Sigma}}_4\mathbf{E}^{-1}$, so we have $\sigma_{\max}(\mathbf{P}) \le \frac{\sigma_{\max}(\mathbf{D})}{\sigma_{\min}(\mathbf{B})}$ and $\sigma_{\min}(\mathbf{P}) \ge \frac{\sigma_{\min}(\mathbf{D})}{\sigma_{\max}(\mathbf{B})}$.

From the above derivation we have

$$\|\mathbf{A}^\top\mathbf{B} - \mathbf{C}^\top\mathbf{P}\mathbf{B}\|_F \le \|\mathbf{A}^\top - \mathbf{C}^\top\mathbf{P}\|_F \cdot \sigma_{\max}(\mathbf{B}) \le \frac{\varepsilon\sigma_{\max}(\mathbf{B})}{\sigma_{\min}(\mathbf{B})}$$

By triangle inequality, this further implies

$$\|\mathbf{C}^\top\mathbf{P}\mathbf{B} - \mathbf{C}^\top\mathbf{P}\mathbf{P}^{-1}\mathbf{D}\|_F \le \varepsilon + \frac{\varepsilon\sigma_{\max}(\mathbf{B})}{\sigma_{\min}(\mathbf{B})} \implies \|\mathbf{B} - \mathbf{P}^{-1}\mathbf{D}\|_F \le \left(\varepsilon + \frac{\varepsilon\sigma_{\max}(\mathbf{B})}{\sigma_{\min}(\mathbf{B})}\right) \cdot \frac{1}{\sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{P})}$$

□

Claim P.9. *Suppose we have matrices* $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{k\times m_1}$ *and* $\mathbf{B}, \mathbf{D} \in \mathbb{R}^{k\times m_2}$ *for some* $m_1, m_2 \ge k$ *and* $\|\mathbf{A}^\top\mathbf{B} - \mathbf{C}^\top\mathbf{D}\|_F \le \varepsilon < \sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{D})$. *Then, there exists some matrix* $\mathbf{P} \in \mathbb{R}^{k\times k}$ *so that:*

- $\|\mathbf{A}^\top - \mathbf{C}^\top\mathbf{P}\|_F \le \frac{\varepsilon\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{D})-\varepsilon}$,
- $\|\mathbf{B} - \mathbf{P}^{-1}\mathbf{C}\|_F \le \frac{2\varepsilon\cdot(\sigma_{\max}(\mathbf{B}))^2\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{D})-\varepsilon)^2}$, *and*
- *the singular values of* $\mathbf{P}$ *are within* $\left[\frac{\sigma_{\min}(\mathbf{D})}{\sigma_{\max}(\mathbf{B})}, \frac{\sigma_{\max}(\mathbf{D})\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{D})-\varepsilon}\right]$.

Proof of Claim P.9. Without loss of generality (by left/right multiplying a unitary matrix), let us assume that $\mathbf{C} = (\overline{\mathbf{C}}, \mathbf{0})$ and $\mathbf{D} = (\overline{\mathbf{D}}, \mathbf{0})$ for $\overline{\mathbf{C}}, \overline{\mathbf{D}} \in \mathbb{R}^{k\times k}$. Let us write $\mathbf{A} = (\overline{\mathbf{A}}, *)$ and $\mathbf{B} = (\overline{\mathbf{B}}, *)$ for $\overline{\mathbf{A}}, \overline{\mathbf{B}} \in \mathbb{R}^{k\times k}$. We have the following relationships

$$\sigma_{\min}(\overline{\mathbf{C}}) = \sigma_{\min}(\mathbf{C}) \ , \quad \sigma_{\min}(\overline{\mathbf{D}}) = \sigma_{\min}(\mathbf{D}) \ , \quad \sigma_{\max}(\overline{\mathbf{A}}) \le \sigma_{\max}(\mathbf{A}) \ , \quad \sigma_{\min}(\overline{\mathbf{B}}) \le \sigma_{\min}(\mathbf{B}) \ .$$

Now, the bound $\|\mathbf{A}^\top\mathbf{B} - \mathbf{C}^\top\mathbf{D}\|_F \le \varepsilon$ translates to (by only looking at its top-left $k \times k$ block) $\|\overline{\mathbf{A}}^\top\overline{\mathbf{B}} - \overline{\mathbf{C}}^\top\overline{\mathbf{D}}\|_F \le \varepsilon$. Since these four matrices are square matrices, we immediately have $\sigma_{\min}(\overline{\mathbf{B}}) \ge \frac{\sigma_{\min}(\overline{\mathbf{C}})\sigma_{\min}(\overline{\mathbf{D}})-\varepsilon}{\sigma_{\max}(\overline{\mathbf{A}})}$. Plugging in the above relationships, the similar bound holds without the hat notion:

$$\sigma_{\min}(\mathbf{B}) \ge \frac{\sigma_{\min}(\mathbf{C})\sigma_{\min}(\mathbf{D}) - \varepsilon}{\sigma_{\max}(\mathbf{A})} \ .$$

Plugging this into the bounds of Claim P.8, we finish the proof. □

Claim P.10. *Suppose we have matrices $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{k\times m}$ for some $m \ge k$ and $\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{C}\|_F \le \varepsilon \le \frac{1}{2}(\sigma_{\min}(\mathbf{C}))^2$, then there exists some unitary matrix $\mathbf{U} \in \mathbb{R}^{k\times k}$ so that*

$$\|\mathbf{A}^\top - \mathbf{C}^\top\mathbf{U}\|_F \le \frac{7\varepsilon(\sigma_{\max}(\mathbf{A}) + \sigma_{\max}(\mathbf{C}))^2(\sigma_{\max}(\mathbf{C}))^3}{(\sigma_{\min}(\mathbf{C}))^6} \ .$$

Proof of Claim P.10. Applying Claim P.9, we know there exists matrix $\mathbf{P} \in \mathbb{R}^{k\times k}$ so that:

- $\|\mathbf{A}^\top - \mathbf{C}^\top\mathbf{P}\|_F \le \frac{2\varepsilon\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C}))^2}$,
- the singular values of $\mathbf{P}$ are within $\left[\frac{\sigma_{\min}(\mathbf{C})}{\sigma_{\max}(\mathbf{A})}, \frac{2\sigma_{\max}(\mathbf{C})\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C}))^2}\right]$.

They together imply

$$\begin{aligned}\|\mathbf{A}^\top\mathbf{A} - \mathbf{C}^\top\mathbf{P}\mathbf{P}^\top\mathbf{C}\|_F &\le \frac{2\varepsilon\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C}))^2} \cdot (\sigma_{\max}(\mathbf{A}) + \sigma_{\max}(\mathbf{C})\sigma_{\max}(\mathbf{P})) \\ &\le \frac{2\varepsilon\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C}))^2} \cdot \frac{3(\sigma_{\max}(\mathbf{C}))^2\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C}))^2} \le \frac{6\varepsilon(\sigma_{\max}(\mathbf{A}))^2(\sigma_{\max}(\mathbf{C}))^2}{(\sigma_{\min}(\mathbf{C}))^4}\end{aligned}$$

By triangle inequality we have

$$\|\mathbf{C}^\top\mathbf{C} - \mathbf{C}^\top\mathbf{P}\mathbf{P}^\top\mathbf{C}\|_F \le \frac{7\varepsilon(\sigma_{\max}(\mathbf{A}) + \sigma_{\max}(\mathbf{C}))^2(\sigma_{\max}(\mathbf{C}))^2}{(\sigma_{\min}(\mathbf{C}))^4}$$

Putting $\mathbf{C}$ into its SVD decomposition, one can easily verify that this implies

$$\|\mathbf{I} - \mathbf{P}\mathbf{P}^\top\|_F \le \frac{7\varepsilon(\sigma_{\max}(\mathbf{A}) + \sigma_{\max}(\mathbf{C}))^2(\sigma_{\max}(\mathbf{C}))^2}{(\sigma_{\min}(\mathbf{C}))^6}$$

Putting $\mathbf{P}$ into its SVD decomposition, one can easily verify that this implies the existence of some unitary matrix $\mathbf{U}$ so that[39]

$$\|\mathbf{U} - \mathbf{P}\|_F \le \frac{7\varepsilon(\sigma_{\max}(\mathbf{A}) + \sigma_{\max}(\mathbf{C}))^2(\sigma_{\max}(\mathbf{C}))^2}{(\sigma_{\min}(\mathbf{C}))^6} \ .$$

Finally, we replace $\mathbf{P}$ with $\mathbf{U}$ in the bound $\|\mathbf{A}^\top - \mathbf{C}^\top\mathbf{P}\|_F \le \frac{2\varepsilon\sigma_{\max}(\mathbf{A})}{(\sigma_{\min}(\mathbf{C}))^2}$, and finish the proof. □

P.6 Nonconvex Optimization Theory

Fact P.11. *For every B-second-order smooth function $f : \mathbb{R}^d \to \mathbb{R}$, every $\varepsilon > 0$, every fixed vectors $x \in \mathbb{R}^d$, suppose for every sufficiently small $\eta > 0$, there exists vector $x_1 \in \mathbb{R}^d$ and a random vector $x_2 \in \mathbb{R}^d$ with $\mathbb{E}[x_2] = 0$ satisfying $\|x_1\|_2 \le Q_1$, $\mathbb{E}[\|x_2\|_2^2] \le Q_2$ and*

$$\mathop{\mathbb{E}}_{x_2}[f(x + \eta x_1 + \sqrt{\eta}x_2)] \le f(x) - \eta\varepsilon \ .$$

Then, either $\|\nabla f(x)\| \ge \frac{\varepsilon}{2Q_1}$ or $\lambda_{\min}(\nabla^2 f(x)) \le -\frac{\varepsilon}{Q_2}$, where $\lambda_{\min}$ is the minimal eigenvalue.

Proof of Fact P.11. We know that

$$\begin{aligned}&f(x + \eta x_1 + \sqrt{\eta}x_2) \\ &= f(x) + \langle \nabla f(x), \eta x_1 + \sqrt{\eta}x_2\rangle + \frac{1}{2}(\eta x_1 + \sqrt{\eta}x_2)^\top \nabla^2 f(x)(\eta x_1 + \sqrt{\eta}x_2) \pm O(B\eta^{1.5}).\end{aligned}$$

Taking expectation, we know that

$$\mathbb{E}[f(x + \sqrt{\eta}x_2)] = f(x) + \eta\langle\nabla f(x), x_1\rangle + \eta\frac{1}{2}\mathbb{E}\left[x_2^\top\nabla^2 f(x)x_2\right] \pm O(B\eta^{1.5})$$

Thus, either $\langle\nabla f(x), x_1\rangle \le -\varepsilon/2$ or $\mathbb{E}\left[x_2^\top\nabla^2 f(x)x_2\right] \le -\varepsilon$, which completes the proof. □

[39]Indeed, if the singular values of $\mathbf{P}$ are $p_1, \dots, p_k$, then $\|\mathbf{I} - \mathbf{P}\mathbf{P}^\top\|_F \le \delta$ says $\sum_i(1 - p_i^2)^2 \le \delta^2$, but this implies $\sum_i(1 - p_i)^2 \le \delta^2$.