TOWARDS UNDERSTANDING HOW DEEP TRANSFORMERS PERFORM CLASSIFICATION WITH UNLABELED CONTEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers achieve remarkable performance in various tasks in natural language processing and computer vision. Despite the empirical success of transformers, the impact of model depth on performance remains not fully understood. In this paper, we consider a classification task with an unlabeled context, and theoretically investigate the performance of a deep transformer model trained by gradient descent in learning this task. Our analysis demonstrates that deeper transformer layers can produce features with higher signal-to-noise ratio (SNR), and as a result, linear classification based on features of deeper layers can yield higher prediction accuracy. Experiments on both synthetic and real-world datasets are conducted to validate our theory. Our study sheds light on the benefits and impact of transformer model depth, and provides insights into how unlabeled context can enhance transformer performance on supervised tasks.

1 INTRODUCTION

Transformer-based models achieve significant success and have become crucial models in a wide range of domains, including vision (Carion et al., 2020; Dosovitskiy et al., 2021), language comprehension and generation (Devlin et al., 2019; Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023; Bi et al., 2024), audio (Baevski et al., 2020), and multimodal (Radford et al., 2021) applications. Typically, these transformer models are built with deep architectures. However, the underlying reasons for the superior performance of deep transformers and the influence of model depth on their performance remain unclear.

Some efforts have been dedicated to investigating how the depth of transformers influences their performance. He & Su (2025) observed that every layer contributes to improved prediction accuracy in a wide range of open-source LLMs, including transformer-based models; however, there is no theoretical explanation for this phenomenon. Wang et al. (2025) proposed a network in which each layer consists solely of a multi-subspace self-attention architecture and proved that each layer enhances the signal-to-noise ratio in the token representations under the linear representation and superposition hypotheses. Li et al. (2025) analyzed the semi-supervised in-context learning and found that one-layer linear attention models do not benefit from unlabeled data, whereas multi-layer or looped transformers with linear attention can effectively leverage unlabeled data. However, training guarantees for deep transformers remain unaddressed in Wang et al. (2025) and Li et al. (2025).

This paper aims to provide a theoretical explanation of how the self-attention mechanism can effectively learn features from the unlabeled context in deep transformers. We investigate a classification problem on an attention-only deep transformer model. This classification problem involves an unlabeled context, where the context contains no label information. We prove the training convergence of the deep transformer with the gradient descent algorithm, and examine the noise reduction effect of the trained transformer by analyzing the signal-to-noise ratio (SNR) in the outputs across different layers. Our results reveal that the self-attention mechanism can reduce noise across all attention layers in the deep transformer without utilizing labels from the context. The major contributions of this paper are summarized as follows:

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

- We prove that gradient descent can train deep transformers to convergence when performing supervised classification tasks in the unlabeled context. To our knowledge, this is among the first theoretical works that establish precise convergence results for deep transformers.
- This paper offers a theoretical explanation for the denoising effect in deep transformers. We prove that the SNR of the outputs increases across all layers in the trained deep transformer, demonstrating every self-attention layer contributes to noise reduction by utilizing the unlabeled context.
- We also investigate a downstream logistic regression task based on features from the hidden layers of the transformers. The population risk bound for the downstream task demonstrates that denoised data leads to improved performance. This provides insight into the advantage of deep transformer models.
- Experiments on synthetic data adhering to our settings confirm our theoretical results. Moreover, experiments on more complex real-world datasets and general transformer-based models (GPT-2) yield similar results, corroborating the broader applicability of our theoretical findings.

Additional related works are presented in Appendix B.

2 Problem Setting

This section presents the problem setting considered in this paper. We introduce the classification task with unlabeled context, define the attention-only deep transformer model, and then specify the details of the training algorithm.

Definition 2.1. *Let $\boldsymbol{\mu} \in \mathbb{R}^d$ be the signal vector in each data point, then each data point $(\mathbf{x}, y)$ are generated from the following data model $\mathcal{D}$:*

$$\mathbf{x} = y \cdot \boldsymbol{\mu} + \boldsymbol{\xi} \in \mathbb{R}^d,$$

where $y \cdot \boldsymbol{\mu}$ refers to the signal part of the data point, in which the label $y \in \mathbb{R}$ is generated from the Rademacher distribution with $y \in \{-1, +1\}$; and the noise part $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{noise}}^2 \cdot (\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2}))$ follows the Gaussian distribution with σ_{noise} denoting the standard deviation of the noise. The whole data point $\mathbf{x}$ is given in the format of a mixture of the signal and noise parts. Similar Gaussian mixture data models have been considered in a series of theoretical works in this field (Pathak et al., 2024; Shen et al., 2025; He et al., 2025; Chen et al., 2026; Li et al., 2025).

In this paper, we investigate how the self-attention mechanism achieves the denoising effect on noisy input data across layers in deep transformers. The data model $\mathcal{D}$ here represents the noisy data input $\mathbf{x}$, which is a natural choice for this study, as it enables us to compare how much signal and noise are preserved. We analyze the denoising effect of the transformer by comparing the relative preserved signal and noise in the outputs across different layers.

To capture the relative signal and noise preserved by the outputs across layers in transformers, we define the SNR as $\text{SNR}_0 = \|\boldsymbol{\mu}\|_2/(\sigma_{\text{noise}}\sqrt{d}) \approx \|\boldsymbol{\mu}\|_2/\|\boldsymbol{\xi}\|_2$, where the approximation is accurate when d is large.

Deep Transformer Model. Transformer-based models are usually composed of multiple layers, with self-attention mechanism serving as the key component. In this paper, our attention-only deep transformer model $f(\mathbf{X}) : \mathbb{R}^{d\times N} \to \mathbb{R}^{N\times 1}$ is defined as follows.

$$\begin{aligned} f(\mathbf{X}) =& \mathbf{v}^\top[\mathbf{H}_L + \mathbf{H}_L \cdot \text{Softmax}(\mathbf{H}_L^\top \mathbf{W}_L \mathbf{H}_L)], \\ \mathbf{H}_{l+1} =& \mathbf{H}_l + \mathbf{H}_l \cdot \text{Softmax}(\mathbf{H}_l^\top \mathbf{W}_l \mathbf{H}_l),\ l = 0, \ldots, L-1, \end{aligned} \tag{2.1}$$

where L is the number of layers of the transformer, and $\mathbf{v} \in \mathbb{R}^{d\times 1}$ is a parameter vector. Moreover, we consider re-parameterizing the key and query matrices in each self-attention module into a single matrix, and denote it as $\mathbf{W}_l \in \mathbb{R}^{d\times d}$ for layer l, $l = 0, \ldots, L$. This re-parameterization is frequently considered in recent theoretical studies of transformers (Zhang et al., 2024a; Wang et al., 2024b; Zhang et al., 2025). The function $\text{Softmax}(\cdot) : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ denotes the column-wise Softmax function given as $\text{Softmax}(\mathbf{U})_{ij} = \frac{\exp(u_{ij})}{\sum_{s=1}^N \exp(u_{sj})}$.

The input matrix $\mathbf{H}_0$ of the model (2.1) is in the format of unlabeled context and query, given as

$$\mathbf{H}_0 = \mathbf{X} = \big[\underbrace{\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}}_{\text{context}}, \underbrace{\mathbf{x}_N}_{\text{query}}\big] \in \mathbb{R}^{d\times N}, \tag{2.2}$$

where N is the number of data points in the input matrix, every data point $\mathbf{x}_i \in \mathbb{R}^d,\ i \in [N]$ follows the data model $\mathcal{D}$ in Definition 2.1.

The primary goal of the learning task is to predict the label of $\mathbf{x}_N$, which constitutes a supervised classification task. In $\mathbf{H}$, $\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}$ serve as the unlabeled context. With this setting, the transformer (2.1) leverages the *unlabeled context* in $\mathbf{H}$ to facilitate the supervised learning task.

Training. In the training, the population loss function we consider is defined as

$$L(\mathbf{v}, \{\mathbf{W}_l\}_{l=0}^L) = \mathbb{E}\big[\ell\big(y_N \cdot f(\mathbf{X}) \cdot \mathbf{e}_N\big)\big], \tag{2.3}$$

where $\ell(z) = \log(1 + \exp(-z))$ is the cross-entropy loss, and the constant vector $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top \in \mathbb{R}^N$, where only the i-th entry in $\mathbf{e}_i$ equals 1.

The gradient descent (GD) algorithm is adopted to minimize the loss function (2.3), with the initialization $\mathbf{v}^{(0)} = \mathbf{0}_d,\ \mathbf{W}_l^{(0)} = \mathbf{0}_{d\times d}, l = 0, \ldots, L$. Zero-initialization is a commonly used initialization in many related papers studying training guarantees of transformers (Von Oswald et al., 2023; Wang et al., 2024b; Shi & Cao, 2025). For iteration $t = 0, 1, \ldots,$ the gradient descent updates of $\mathbf{v}$ and $\{\mathbf{W}_l\}_{l=0}^L$ are given as

$$\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} - \eta \frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(t)}, \{\mathbf{W}_l^{(t)}\}_{l=0}^L\big). \tag{2.4}$$

$$\mathbf{W}_l^{(t+1)} = \mathbf{W}_l^{(t)} - \eta \frac{\partial L}{\partial \mathbf{W}_l}\big(\mathbf{v}^{(t)}, \{\mathbf{W}_l^{(t)}\}_{l=0}^L\big). \tag{2.5}$$

We introduce some notations needed for our analysis. Denote the input matrix of layer l at iteration t as $\mathbf{H}_l^{(t)}$, i.e., $\mathbf{H}_l^{(t)} = [\mathbf{h}_{l,1}^{(t)}, \ldots, \mathbf{h}_{l,N}^{(t)}] \in \mathbb{R}^{d\times N},\ l = 0, \ldots, L$, where $\mathbf{h}_{l,i}^{(t)} \in \mathbb{R}^d$ represents l-th layer, i-th token in $\mathbf{H}_l^{(t)}$ at t-th iteration. With this notation, we have the $\mathbf{H}_0^{(t)} \equiv \mathbf{X}$.

By our data model $\mathcal{D}$ in Definition 2.1, the signal-to-noise ratio (SNR) of the original data input $\mathbf{X}$ in the transformer is defined as $\mathrm{SNR}_0 = \|\boldsymbol{\mu}\|_2/(\sigma_{\text{noise}}\sqrt{d}) \approx \|\boldsymbol{\mu}\|_2/\|\boldsymbol{\xi}\|_2$. For the outputs of a general layer in deep transformer (2.1), denote $\mathrm{SNR}_l^{(t)}$ as the SNR of the output of layer $l,\ l = 0, \ldots, L$ at iteration t. The corresponding SNR in the output of layer l is defined as the ratio of the norm of signal part and noise part, namely

$$\mathrm{SNR}_l^{(t)} = \frac{|\langle \mathbf{h}_{l,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2}\rangle|}{\|(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2})\mathbf{h}_{l,i}^{(t)}\|_2}, \tag{2.6}$$

where $\big\langle \mathbf{h}_{l,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2}\big\rangle$ is the projection of $\mathbf{h}_{l,i}^{(t)}$ onto the direction of $\boldsymbol{\mu}$, and $(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2})\mathbf{h}_{l,i}^{(t)}$ is the projection of $\mathbf{h}_{l,i}^{(t)}$ onto the perpendicular space of $\boldsymbol{\mu}$. This general definition of SNR allows us to analyze how the signals preserved in features evolve across layers in the transformer.

3 Main results

This section presents the main theoretical results. We first give the training convergence guarantee of deep attention-only transformers (2.1) and analyze its denoising effect across layers. Then, inspired by He & Su (2025), we investigate the performance of downstream logistic regression based on the hidden layer features of the trained transformer.

Main results of this paper are established under the following conditions on the dimension d, the learning rate η, the number of layers L in transformer model (2.1).

Condition 3.1. *Suppose that*

1. *The number of layers L satisfies $L = \Theta(\log(N))$ and $L \geq \Omega\big(\log\log(\|\boldsymbol{\mu}\|_2^2 \cdot \max\{\mathrm{SNR}_0^2, 1\})\big)$.*

2. *The learning rate η satisfies $\eta = \Theta(\|\boldsymbol{\mu}\|_2^{-2})$.*
3. *The dimension d satisfies $d \geq \Omega(\max\{N^2, 9^L \cdot \|\boldsymbol{\mu}\|_2^2 \cdot \mathrm{SNR}_0^{-4}\})$.*

These assumptions in Condition 3.1 are mild and reasonable. The number of layers L is assumed to be of order $\log(N)$, ensuring that the model depth is not extremely large. This mild depth assumption avoids specifying an exact number of shallow layers, as is common in prior work. The learning rate η is chosen to be of order $\|\boldsymbol{\mu}\|_2^{-2}$, which excludes excessively large learning rates and ensures efficient training. We assume the dimension d is sufficiently large so that the model operates in the over-parameterized regime. This setting is standard in the theoretical study of signal learning and underlies many analyses of linear and one-layer models (Wu et al., 2023; Song et al., 2024).

We also make an assumption on a maximum admissible number of training iterations, and denote it as $T^* = \mathrm{poly}(\|\boldsymbol{\mu}\|_2, N, 2^L, d) \geq 2$. This is a mild assumption as the training in practice cannot be exponentially long and typically requires only a polynomial number of iterations. In the following, we analyze the training convergence and SNR across layers within this maximum admissible number of iterations T^*.

Convergence and SNR analysis. The following Theorem 3.2 demonstrates convergence of gradient descent training, and presents the increase of SNR across layer-wise outputs of the trained transformer model.

Theorem 3.2. *Under Condition 3.1, for all $2 \leq t \leq T^*$, it holds that*

- *the population loss function* (2.3) *after iteration t satisfies*

$$L\big(\mathbf{v}^{(t)}, \{\mathbf{W}_l^{(t)}\}_{l=0}^L\big) \leq \exp(-c_8 \cdot 2^L),$$

  where $c_8 > 0$ is a positive constant.
- *for any input matrix $\mathbf{X}$ generated following* (2.2) *and any $l = 0, \ldots, L$, with probability at least $1 - \exp(-c_2 \cdot 2^L) - 2\exp(-\sqrt{N})$, the ratio of SNRs between the outputs of layer $l+1$ and layer l in the trained transformer at t-th iteration have the following lower bound,*

$$\frac{\mathrm{SNR}_{l+1}^{(t)}}{\mathrm{SNR}_l^{(t)}} \geq \frac{2 - \epsilon(N, d, L)}{1 + p^2}, \tag{3.1}$$

  where $p = \exp(-\Theta(\|\boldsymbol{\mu}\|_2^2))$ and

$$\epsilon(N, d, L) = O\Big(\max\Big\{\frac{N^{\frac{1}{4}}}{\sqrt{d}}, \frac{1}{\sqrt{N}}, \exp(-2^L), d^{-\frac{3}{8}}\Big\}\Big).$$

Theorem 3.2 presents the lower bound for the ratio of SNRs between layer $l+1$ and layer l. Under Condition 3.1, it is easy to see that $\epsilon(N, d, L)$ in (3.1) is relatively small, leading to a ratio of SNRs across layers that is larger than one whenever $\|\boldsymbol{\mu}\|_2 = \Theta(1)$, demonstrating the trained transformer layers can enhance SNR. We can also observe in (3.1) that a larger number of tokens N lead to a greater increase in SNR across layers, which implies that advantage of having more unlabeled context data.

The key reason for the denoising effect is that, after training, we observed that the trained $\mathbf{W}_l^* = m_1\boldsymbol{\mu}\boldsymbol{\mu}^\top + m_2\mathbf{I}_d$ (as shown in (E.1)) with the first term in $\mathbf{W}_l^*$ being dominant. At layer l, the matrix $\mathbf{H}_l{}^\top\mathbf{W}_l^*\mathbf{H}_l = [\mathbf{h}_{l,j}{}^\top\mathbf{W}_l^*\mathbf{h}_{l,i}]_{ji} = m_1\mathbf{h}_{l,j}{}^\top\boldsymbol{\mu}\boldsymbol{\mu}^\top\mathbf{h}_{l,i} + m_2\mathbf{h}_{l,j}{}^\top\mathbf{h}_{l,i}$ in Softmax function shows that the attention assigns higher scores to pairs of tokens $\mathbf{h}_{l,i}$ and $\mathbf{h}_{l,j}$ from the same class. Such "within-class averaging" is the primary reason why self-attention exhibits the denoising effect.

Two recent works explored semi-supervised in-context learning with transformers, emulating the expectation-maximization (EM) algorithm. Li et al. (2025) analyzed multilayer and looped transformers with linear attention, showing how each linear attention module can update label estimates by mimicking EM algorithm. This study reveals that transformer depth enables effective use of unlabeled samples in the context by allowing the leading polynomial degree to grow exponentially with the number of layers. Liu & Yang (2025) focused on a multi-class linear classification problem and demonstrated that Softmax attention in four-layer transformers with chain-of-thought (CoT) prompting approximates an EM estimator. In terms of training, Li et al. (2025) did not cover the training guarantees of the multi-layer looped transformers, while Liu & Yang (2025) proved that

a four-layer transformer converges when trained with teacher forcing, provided that only the first layer's parameters are updated during training while the last three layers are frozen with specifically constructed weights. Compared with these existing results, our work gives a more thorough convergence guarantee of deep transformers. More importantly, our result offers a novel perspective concerning Softmax attention in deep transformers: the self-attention mechanism can effectively reduce noise across layers by performing "within-class averaging" over the input features.

Downstream tasks. Theorem 3.2 characterizes the denoising effect of the transformer (2.1) by comparing the ratios of SNRs across layers, and demonstrates that the SNR increases with depth. To demonstrate how this noise reduction corresponds to performance improvement, in this paper, we consider a logistic regression problem as the downstream task. We train the logistic regression model with the outputs of different layers in transformer (2.1). We compare the prediction accuracy of the logistic regression model by demonstrating the upper bound of the population risk.

For M independent copies of input matrices $\mathbf{X}_1, \ldots, \mathbf{X}_M$ that are fed into the trained deep transformer, we use the last column token of each matrix and its corresponding label to perform logistic regression. Namely, for the input matrix $\mathbf{X}_m, m \in [M]$, the last column is $\mathbf{x}_{N,m}$ with its corresponding label $y_{N,m}$, and $\mathbf{h}_{l,N,m}$ is denoted as its last column output at l-th layer, $l = 0, \ldots, L$. Then for layer l, the dataset for downstream logistic regression is given as

$$\mathcal{S}_l = \{(\mathbf{h}_{l,N,m}, y_{N,m})\}_{m=1}^{M}. \tag{3.2}$$

We adopt linear logistic regression on $\mathcal{S}_l$. To compare the performance for features from different layers, we consider an independent test data copy $(\mathbf{X}, y)$ and denote the corresponding hidden layer output at a certain layer as $\mathbf{h}$. Then the population risk is defined as

$$R(\boldsymbol{\theta}) = \mathbb{P}(y \cdot \langle \boldsymbol{\theta}, \mathbf{h} \rangle < 0).$$

In linear logistic regression, we minimize the following empirical loss by gradient descent,

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} L_l(\boldsymbol{\theta}) := \frac{1}{M} \sum_{m=1}^{M} \log[1 + \exp(-y_m \cdot \langle \boldsymbol{\theta}, \mathbf{h}_{l,N,m} \rangle)]. \tag{3.3}$$

By Theorem 3 in Soudry et al. (2018), for this optimization task, the gradient descent iterate $\boldsymbol{\theta}^{(t)}$ converges to the direction of the maximum margin classifier. We denote the resulting maximum margin classifier obtained based on features from the l-th hidden layer as $\widehat{\boldsymbol{\theta}}_l$.

The following Theorem 3.3 presents the upper bounds for the population risk of the logistic regression task on our dataset $\mathcal{S}_l,\ l = 0, \ldots, L$, which are adapted from Theorem 3.1 in Cao et al. (2021).

Theorem 3.3. *Suppose that dimension $d \geq c_9 \cdot \max\{2M^{3/2}, M(\sqrt{d}+1)\}$ for some constant c_9. Then, for the training dataset $\mathcal{S}_l,\ l = 0, \ldots, L$, there exists an absolute constant c_{10} such that the maximum margin classifier $\widehat{\boldsymbol{\theta}}_l$ has the following risk bound: with probability at least $1 - M^{-1} - \exp(-c_2 \cdot 2^L)$, we have*

$$R_l(\widehat{\boldsymbol{\theta}}_l) \leq \exp\left(\frac{-c_{10} \cdot (\frac{2}{1+p^2})^{2l}}{1 + M^{-1}(1 + \mathrm{SNR}_0^{-4}) + d^{-2}\mathrm{SNR}_0^{-4}} \right),$$

where $p = \exp(-\Theta(\|\boldsymbol{\mu}\|_2^2))$.

Theorem 3.3 shows that, for the downstream logistic regression task, the population risk bound decreases with l. This demonstrates that higher prediction accuracy can be achieved by using deeper layer outputs of the transformer (2.1) as dataset. Combined this with our results of the SNR increases across layers in Theorem 3.2, Theorem 3.3 shows that the increase in SNR across layers corresponds to features with more effectively captured signals, resulting in improved performance on the downstream tasks. This downstream performance evaluation translates the relatively abstract SNR gains in each attention layer of deep transformers into more easily interpretable results, highlighting the importance of SNR as an indicator of learning quality.

4 PROOF SKETCH

This section gives a proof sketch of our main results, Theorem 3.2, with the full proof given in Appendix D and E.

Part 1. **Decomposition.** We first decompose hidden-layer tokens into a linear combination of the signal vector and noise vectors associated with the input data (Eq. (D.2)), where coefficients $\alpha_{l,i}^{(t)}$ and $\beta_{l,i,k}^{(t)}$ quantify the contributions of the signal and noise components respectively. This reformulation translates the analysis of SNR across layers into tracking the evolution of the coefficients $\alpha_{l,i}^{(t)}, \beta_{l,i,k}^{(t)}$ during training and propagation across layers.

Part 2. **Training dynamics and coefficient evolution.** Building on this decomposition, we explicitly characterize the model parameters and coefficient dynamics at a reference iteration. The resulting explicit, tractable expressions for attention weights already establish SNR improvement and achieve a sufficiently small population loss, and this serves as the base case for the subsequent analysis. We then track the evolution of the coefficient $\alpha_{l,i}^{(t)}$, $\beta_{l,i,k}^{(t)}$ (Lemma D.4) to show that the SNR improvement persists throughout training, and we bound the population loss (Lemma D.5) within a polynomial number of iterations.

Part 3. **Layer-wise denoising effect.** To demonstrate the denoising effect across all attention layers of the trained deep transformers, we track the evolution of the coefficients $\alpha_{l,i}^{(t)}$, $\beta_{l,i,k}^{(t)}$ across layers using the coefficient dynamics characterized in Part 2 (Appendix E.1). Quantifying the relative amounts of signal and noise captured in successive layer outputs enables a comparison of SNR between layers l and $l+1$, which in turn yields the denoising result stated in Theorem 3.2 (Appendix E.4).

5 EXPERIMENTS

In this section, we present our main experimental results on synthetic and real-world datasets (MNIST, CIFAR-10) to verify the theoretical findings in Section 3. Experiments on these real-world datasets demonstrate that our theoretical results and insights are applicable in broader practical scenarios. Furthermore, to verify that our theoretical findings also hold for more complex architectures, we conduct experiments with GPT-2 model and in multi-class settings. More experimental results and detailed settings are provided in Appendix C.

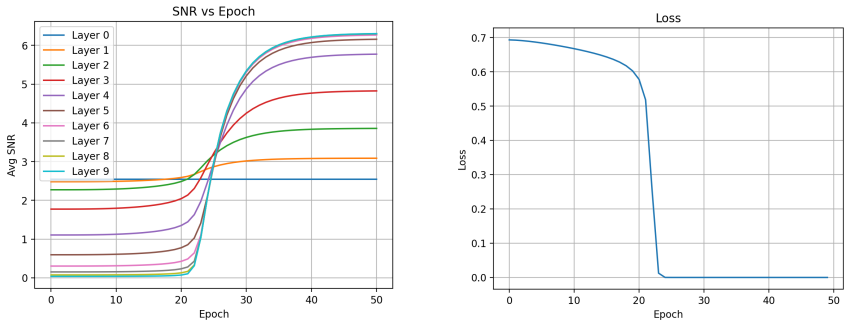
5.1 EXPERIMENTS ON SYNTHETIC DATA

In the synthetic-data experiments, we instantiate the data model $\mathcal{D}$ (Definition 2.1) and train the deep transformer model in (2.1). The detailed settings for the synthetic-data experiments are presented in Appendix C.1.

Layer-wise SNR increases with depth after training. We measure SNR at the outputs of different layers (the input layer, each transformer block output, and final readout) on a held-out set. As shown in Figure 1a, after convergence, the SNR exhibits a clear increasing trend along depth, with features from deeper layers consistently attaining higher SNR. This empirical depth profile matches the monotonic denoising behavior predicted by Theorem 3.2, suggesting that successive self-attention blocks progressively suppress the component orthogonal to $\boldsymbol{\mu}$ while preserving (or amplifying) the signal-aligned component. Figure 1b confirms that the observed SNR improvement across layers is achieved under stable optimization, as the training objective decreases smoothly throughout training and the reported SNR are measured for a model with close-to-zero training loss.

To demonstrate robustness to parameters variations, we evaluate the synthetic setup under multiple configurations by varying dimension d, noise level σ_{noise}, and signal scale $\|\boldsymbol{\mu}\|_2$. Figure 7 and 8 show consistent post-training layer-wise increases in SNR across a wide range of synthetic settings.

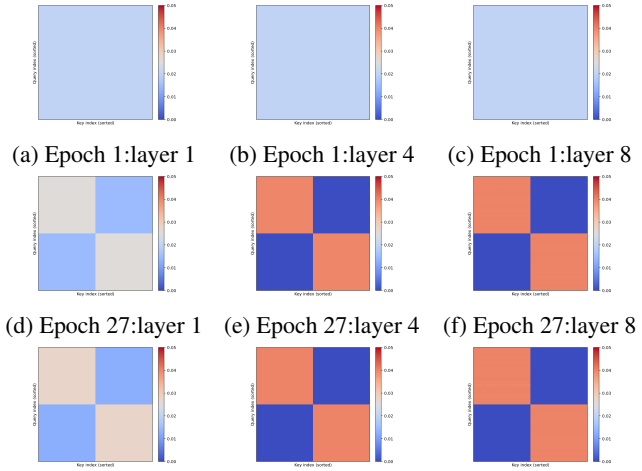
Attention aligns same-class tokens and suggests a denoising mechanism. To probe the underlying mechanism, we visualize the Softmax attention matrices across layers. Before plotting, tokens in each sample are reordered by their ground-truth class, so that ideal class-wise aggregation would manifest as a block-diagonal pattern. As shown in Figure 2, attention evolves from random, cross-class mixing in early training to increasingly within-class concentration at later stages, especially in



(a) Layerwise SNR in training. (b) Training loss curve.

Figure 1: Synthetic data results ($d = 100, \sigma_{\text{noise}} = 0.1, \|\boldsymbol{\mu}\|_2 = 2.5$): Denoising across layers. Training loss decreases steadily to near zero. After convergence, the deeper layers in the transformer achieve larger SNRs.

deeper layers. This shift indicates that the model preferentially pools information from same-class tokens while reducing cross-class interactions. This "within-class averaging" offers a mechanistic explanation for layer-wise SNR increase in Theorem 3.2 and Figure 1a.



(a) Epoch 1:layer 1 (b) Epoch 1:layer 4 (c) Epoch 1:layer 8

(d) Epoch 27:layer 1 (e) Epoch 27:layer 4 (f) Epoch 27:layer 8

(g) Epoch 50:layer 1 (h) Epoch 50:layer 4 (i) Epoch 50:layer 8

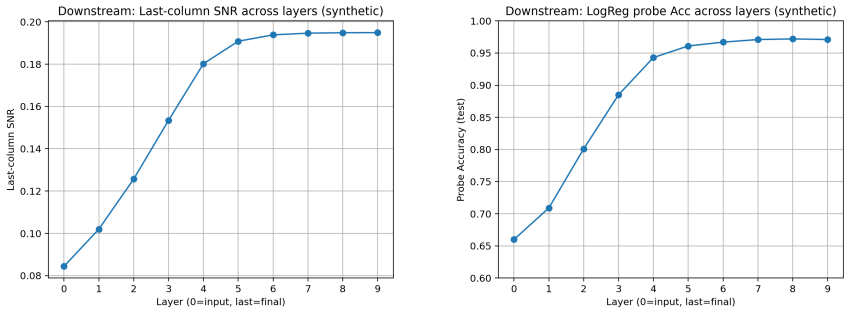
Figure 2: Synthetic data: Token-sorted attention heatmaps during training (representative layers). Attention patterns become progressively more block-diagonal as training proceeds.

Higher SNR corresponds to better downstream performance. To further verify Theorem 3.3, we perform a layer-wise linear probe using ℓ_2-regularized logistic regression on the last-token features and the corresponding label at each layer l. Figure 3b shows that test accuracy of the logistic regression probe improves with depth, mirroring the rise in SNR of the last-token features across layers observed in Figure 3a. Overall, features from layers with higher SNR consistently lead to higher downstream prediction accuracy, supporting SNR as an appropriate proxy for feature quality in our study.

5.2 Experiments on MNIST dataset

In this section, we conduct experiments on the MNIST dataset (LeCun et al., 1998) to evaluate whether the denoising behavior of the deep transformer model (2.1) also emerges on real-world data. Detailed experimental settings are deferred to Appendix C.2.

Problem setup (binary MNIST with noisy tokens). We form a binary classification task on MNIST digits `0` and `1`. Each token is a vectorized image, preprocessed and perturbed by addi-

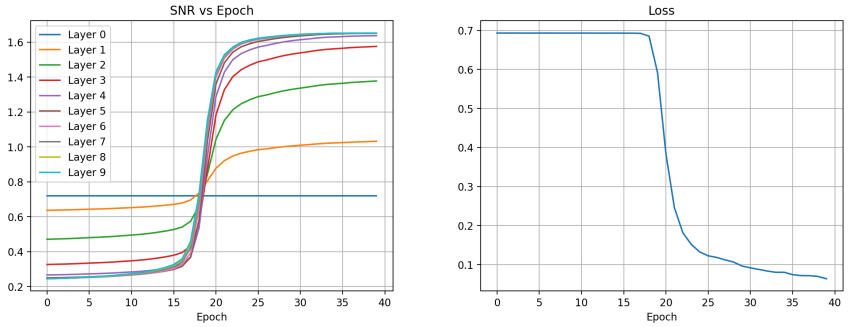


(a) Layer-wise SNR (b) Test accuracy across layers

Figure 3: Synthetic data: Downstream probing across depth. Layer-wise SNR of the last-token features and the corresponding test accuracy of the logistic regression both increase across layers.

tive noise. Each input sample stacks multiple noisy tokens into a matrix that serves as input to the transformer. Figure 9 visualizes representative noisy tokens used as model inputs.

Layer-wise SNR changes in training. We compute layer-wise SNR (the definition is given in (C.3) in the appendix) on a test set for every epoch. The SNR curves in Figure 4a shows that, as training proceeds, deeper layers consistently achieve higher SNRs, indicating a progressive enhancement of within-class signal components and suppression of noise. The training loss depicted in Figure 4b decreases smoothly to near zero over epochs, confirming stable optimization under the same run.

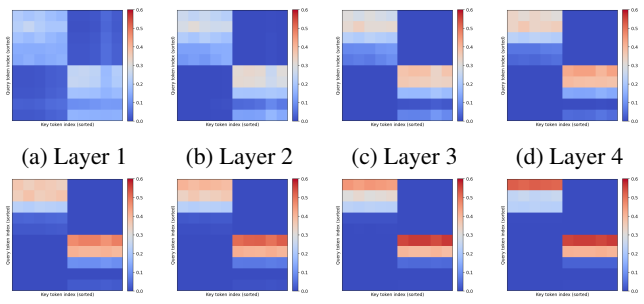


(a) Layerwise SNR in training. (b) Training loss curve.

Figure 4: MNIST: Denoising across layers. The training loss decreases steadily throughout training, and the outputs of deeper layers attain higher SNRs after training.

Attention patterns (sorted heatmaps across layers). We examine the Softmax attention matrices to understand how self-attention achieves denoising. Following the same protocol as in the synthetic experiments, we sort tokens by their ground-truth class within each sample before plotting the attention matrix. Figure 5 shows the sorted heatmaps for layers 1–8. We find that the attention structure evolves across depth: while shallow layers may exhibit relatively diffuse attention, deeper layers tend to place higher weight on a subset of class-consistent tokens, indicating that the model learns to aggregate information from more reliable (signal-dominated) examples and down-weight noisy interactions. This behavior is consistent with the observed SNR improvement in Figure 4a.

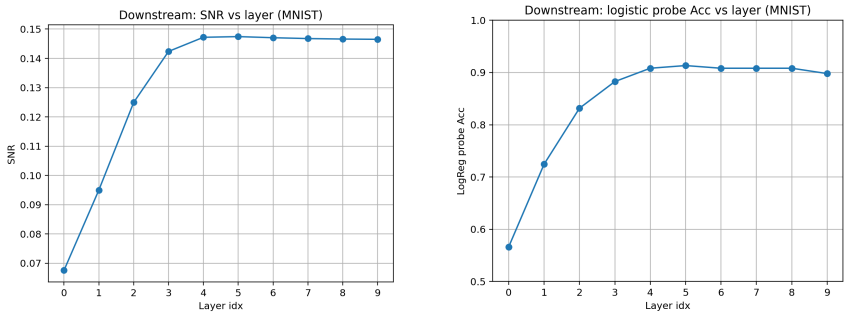
Downstream evaluation. We also perform ℓ_2-regularized logistic regression on frozen features from different layers. For every layer (input, intermediate layers, and the final outputs), we train the probe on a small downstream training set and evaluate it on a substantially larger held-out test set. As shown in Figure 6b, test accuracy increases across layers, indicating that features from deeper layers are more linearly separable. Furthermore, the depth profile of probe accuracy closely follows the layer-wise SNR increase in Figure 6a, supporting a strong positive relationship between feature SNR and downstream performance under the same noisy evaluation protocol.



(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4

(e) Layer 5 (f) Layer 6 (g) Layer 7 (h) Layer 8

Figure 5: MNIST: Token-sorted attention heatmaps across layers (layer 1–8). The evolution of attention across depth suggests increased reliance on within-class token interactions.



(a) Layerwise last-token SNR. (b) Layerwise test accuracy.

Figure 6: MNIST: Downstream probing. Both the SNR of features from deeper layers and the associated logistic regression test accuracy increase progressively across layers.

To evaluate the denoising behavior of the transformer model (2.1) on more complex natural images, we conduct real-world data experiments on the CIFAR-10 dataset (Krizhevsky et al., 2009), which is given in Appendix C.3.

To demonstrate that our theoretical findings also hold for more general architectures and settings, we conduct a series of experiments with GPT-2 model (Radford et al., 2019): The synthetic-data experiments are presented in Appendix C.4. Experiments on MNIST are presented in Appendix C.5. Experiments on CIFAR-10 are in Appendix C.6. Multi-class experiments on MNIST dataset are given in Appendix C.7.

6 Conclusion

This paper provides a theoretical explanation of how depth influences transformer performance through a classification task in unlabeled context. We first present the training convergence of the deep transformer using gradient descent algorithm, and show that the trained transformer achieves layer-wise noise reduction, as evidenced by increasing SNR across successive layer outputs. The population risk bounds for the downstream logistic regression on the transformer outputs demonstrate that deeper layers with higher SNR yield smaller risk bounds and improved prediction accuracy. Our experiments on both synthetic and real-world datasets show that all attention layers can achieve noise reduction effect, resulting in higher test accuracy in downstream tasks. Furthermore, experiments on GPT-2 models confirm that our theoretical findings hold true in broader scenarios. Our work provides insight into the advantage of deep transformer models and demonstrates that leveraging unlabeled context can improve performance in supervised learning tasks. Future work will examine how different layers contribute to performance changes in more complex architectures.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing. We have not used generative AI tools for other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work and check the correctness of it. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Laziz U Abdullaev, Maksim Tkachenko, and Tan M Nguyen. Revisiting transformers with insights from image filtering. *arXiv preprint arXiv:2506.10371*, 2025.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36:57125–57211, 2023.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yuan Cao, Quanquan Gu, and Mikhail Belkin. Risk bounds for over-parameterized maximum margin classification on sub-gaussian mixtures. *Advances in Neural Information Processing Systems*, 34:8407–8418, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Zhiheng Chen, Ruofan Wu, and Guanhua Fang. Transformers as unsupervised learning algorithms: A study on gaussian mixtures. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Yuanbiao Gou, Peng Hu, Jiancheng Lv, Joey Tianyi Zhou, and Xi Peng. Multi-scale adaptive network for single image denoising. *Advances in Neural Information Processing Systems*, 35:14099–14112, 2022.

Akshat Gupta, Jay Yeung, Gopala Anumanchipalli, and Anna Ivanova. How do LLMs use their depth? *arXiv preprint arXiv:2510.18871*, 2025.

Sharut Gupta, Stefanie Jegelka, David Lopez-Paz, and Kartik Ahuja. Context is environment. In *The Twelfth International Conference on Learning Representations*, 2024.

Hangfeng He and Weijie J Su. A law of next-token prediction in large language models. *Physical Review E*, 112(3):035317, 2025.

Yihan He, Yuan Cao, Hong-Yu Chen, Dennis Wu, Jianqing Fan, and Han Liu. Learning spectral methods by transformers. *arXiv preprint arXiv:2501.01312*, 2025.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 19660–19722. PMLR, 2024.

Hidetaka Kamigaito, Ying Zhang, Jingun Kwon, Katsuhiko Hayashi, Manabu Okumura, and Taro Watanabe. Diversity of transformer layers: One aspect of parameter scaling laws. *arXiv preprint arXiv:2505.24009*, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yingcong Li, Xiangyu Chang, Muti Kara, Xiaofeng Liu, Amit Roy-Chowdhury, and Samet Oymak. When and how unlabeled data provably improve in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, October 2021.

Renpu Liu and Jing Yang. Unlabeled data can provably enhance in-context learning of transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Renpu Liu, Ruida Zhou, Cong Shen, and Jing Yang. On the learn-to-optimize capabilities of transformers in in-context sparse recovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Arvind V. Mahankali, Tatsunori Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. In *The Twelfth International Conference on Learning Representations*, 2024.

William Merrill, Ashish Sabharwal, and Noah A Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022.

Aaron Mueller, Robert Frank, Tal Linzen, Luheng Wang, and Sebastian Schuster. Coloring the blank slate: Pre-training imparts a hierarchical inductive bias to sequence-to-sequence models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1352–1368, 2022.

Reese Pathak, Rajat Sen, Weihao Kong, and Abhimanyu Das. Transformers can optimally learn regression mixture models. In *The Twelfth International Conference on Learning Representations*, 2024.

Jackson Petty, Sjoerd Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. The impact of depth on compositional generalization in transformer language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7232–7245, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. *Advances in neural information processing systems*, 36:14228–14246, 2023.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

Hao Shen, Zhong-Qiu Zhao, and Wandi Zhang. Adaptive dynamic filtering network for image denoising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2227–2235, 2023.

Wei Shen, Ruida Zhou, Jing Yang, and Cong Shen. On the training convergence of transformers for in-context classification of gaussian mixtures. In *Forty-second International Conference on Machine Learning*, volume 267, pp. 54732–54771. PMLR, 2025.

Wei Shi and Yuan Cao. Towards understanding transformers in learning random walks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Bingqing Song, Boran Han, Shuai Zhang, Jie Ding, and Mingyi Hong. Unraveling the gradient descent dynamics of transformers. *Advances in Neural Information Processing Systems*, 37:92317–92351, 2024.

Xinyuan Song, Keyu Wang, PengXiang Li, Lu Yin, and Shiwei Liu. Demystifying the roles of llm layers in retrieval, knowledge, and reasoning. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16792–16796. IEEE, 2026.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70): 1–57, 2018.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations*, 2022.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, October 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, volume 202, pp. 35151–35174. PMLR, 2023.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024a.

Mingze Wang and Weinan E. Understanding the expressive power and mechanisms of transformer for sequence modeling. In *Advances in Neural Information Processing Systems*, volume 37, pp. 25781–25856, 2024.

Peng Wang, Yifu Lu, Yaodong Yu, Druv Pai, Qing Qu, and Yi Ma. Attention-only transformers via unrolled subspace denoising. In *Forty-second International Conference on Machine Learning*, volume 267, pp. 63840–63859. PMLR, 2025.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17683–17693, June 2022.

Zixuan Wang, Stanley Wei, Daniel Hsu, and Jason D Lee. Transformers provably learn sparse token selection while fully-connected nets cannot. In *Forty-first International Conference on Machine Learning*, pp. 51854–51912, 2024b.

Yongtao Wu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. On the convergence of encoder-only shallow transformers. *Advances in Neural Information Processing Systems*, 36:52197–52237, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022.

Jiawei Yang, Katie Z Luo, Jiefeng Li, Congyue Deng, Leonidas Guibas, Dilip Krishnan, Kilian Q Weinberger, Yonglong Tian, and Yue Wang. Denoising vision transformers. In *European Conference on Computer Vision*, pp. 453–469, 2024.

Chao Yao, Shuo Jin, Meiqin Liu, and Xiaojuan Ban. Dense residual transformer for image denoising. *Electronics*, 11(3):418, 2022.

Chenyang Zhang, Xuran Meng, and Yuan Cao. Transformer learns optimal variable selection in group-sparse classification. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE transactions on image processing*, 26 (7):3142–3155, 2017.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024a.

Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. Investigating layer importance in large language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 469–479, 2024b.

A NOTATION

$\|\cdot\|_2$ denotes the ℓ^2-norm. $\|\cdot\|_F$ denotes the Frobenius norm. For a vector $\mathbf{a} \in \mathbb{R}^d$ and a positive definite matrix $\mathbf{A}$, we define $\|\mathbf{a}\|_{\mathbf{A}} = \sqrt{\mathbf{a}^\top \mathbf{A}\mathbf{a}}$. For an integer n, denote the set $[n] = \{1, 2, \ldots, n\}$. Denote the product of matrices is with certain order $\prod_{j=1}^{J} \mathbf{A}_j = \mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_J$. Denote $\mathbf{J}_N$ is all-ones matrix of dimension N and $\mathbf{1}_N$ all-ones vector of dimension N. For two sequences $\{a_n\}$ and $\{b_n\}$, denote $a_n = O(b_n)$ if there exists some absolute constant $C > 0$ and $N > 0$ such that $|a_n| \leq C|b_n|$ holds for all $n \geq N$. Denote $a_n = \Omega(b_n)$ if there exist some absolute constant $C > 0$ and $N > 0$ such that $|a_n| \geq C|b_n|$ holds for all $n \geq N$. Denote $a_n = \Theta(b_n)$ if both $a_n = O(b_n)$ and $a_n = \Omega(b_n)$ hold. $\widetilde{O}(\cdot)$, $\widetilde{\Omega}(\cdot)$, $\widetilde{\Theta}(\cdot)$ are used to omit the logarithmic factors in these notations. Denote $a_n = o(b_n)$ if for any $\epsilon > 0$, there exists some integer $N_\epsilon > 0$ such that $|a_n| \leq \epsilon|b_n|$ holds for all $n \geq N_\epsilon$. Denote $a_n = \omega(b_n)$ if $b_n = o(a_n)$. Denote $a_n = \mathrm{poly}(b_n)$ if $a_n = O(b_n^D)$ for some constant $D > 0$.

B RELATED WORK

Theory of In-Context Learning with Transformers. Many theoretical studies have been devoted to understanding in-context learning in transformers. Some works constructed transformers that perform gradient descent for linear regression tasks and non-linear regression tasks (Akyürek et al., 2023; Von Oswald et al., 2023; Huang et al., 2024). Beyond vanilla gradient descent, more recent studies showed that transformers can perform more advanced algorithms across a range of in-context learning tasks (Bai et al., 2023; Ahn et al., 2023; Zhang et al., 2024a; Mahankali et al., 2024; Liu et al., 2025). From statistical perspective, in-context learning can be regarded as an implicit Bayesian inference driven by examples in the prompt, with the pre-training data determining the prior over latent tasks (Xie et al., 2022). Raventós et al. (2023) found that this Bayesian inference framework fails to explain scenarios with high task diversity, and demonstrated that transformers can instead generalize to unseen tasks. Most of the aforementioned papers focused on fully-labeled context settings. Some works investigated unsupervised in-context learning, where the context contains only unlabeled data (Gupta et al., 2024; Agarwal et al., 2024). Several recent studies have attempted to understand how transformers learn from partially unlabeled data in context, known as semi-supervised in-context learning. Li et al. (2025) investigated the influence of semi-supervised in-context learning on the performance of multilayer and looped transformers with linear attention. Liu & Yang (2025) analyzed multi-class linear classification problem and showed that multi-layer transformers with chain-of-thought (CoT) prompting can mimic the behavior of expectation-maximization (EM) algorithm. However, how self-attention mechanism leverages unlabeled context to improve performance of deep transformers remains largely unexplored.

Depth and Transformer Performance. A series of studies has explored how the performance of transformers is influenced by model depth. Empirical findings have demonstrated that deep models generalize more compositionally (Mueller et al., 2022). Merrill et al. (2022) showed that the expressive capacity of transformers increases exponentially with their depth. Wang & E (2024) investigated the roles of critical parameters in the transformer, such as the number of layers and attention heads. Tay et al. (2022) analyzed how scaling choices during pre-training affect downstream fine-tuning performance, and found that scaling by depth is generally more helpful than scaling by width. Petty et al. (2024) observed that, for a fixed total number of parameters, deeper models generalize more compositionally than shallower models. Kamigaito et al. (2025) analyzed transformers using bias-diversity decomposition, and found that performance improvement requires individual layers to make accurate predictions while also remaining mutually different. Some other studies have leveraged the increased depth to improve the performance of transformers (Touvron et al., 2021; Wang et al., 2024a; Saunshi et al., 2025).

LLM Layers Analysis Some existing studies have analyzed the contributions of individual layers in large language models (LLMs) to the overall performance of the models. He & Su (2025) empirically observed that all layers contribute to improved prediction accuracy in a wide range of open-source LLMs, including transformer-based models. Gupta et al. (2025) proposed a "Guess-then-Refine" framework to explain the internal prediction dynamics of LLMs during inference, showing that increased depth allows LLMs to progressively transform raw text into structured, high-level reasoning. Song et al. (2026) emphasized that different LLM layers contribute unevenly across tasks

and that their relative importance depends on the evaluation protocol, task type, and model architecture. Accordingly, claims about layer importance should be regarded as specific to task, metric, and model. For language understanding and reasoning tasks, Zhang et al. (2024b) used Shapley values and layer ablation experiments to measure the importance of individual layers and revealed the existence of cornerstone layers, where certain early layers can exhibit a dominant contribution over others.

Denoising in Transformers. Denoising is an important low-level vision task, which aims to recover high-quality images from the noisy inputs. High quality images are crucial for achieving good performance in a variety of high-level vision tasks, such as image classification and object detection. Previously, many efforts have focused on improving this field using convolutional neural networks (CNN) (Zhang et al., 2017; 2018; 2021; Gou et al., 2022; Shen et al., 2023), which do not show improved ability with an increasing number of layers (Yao et al., 2022). Transformer-based models have gained significant popularity in computer vision domain since the emergence of ViT (Dosovitskiy et al., 2021). Unlike CNN-based networks, transformer-based models tend to exhibit improved performance as their depth increases. Liang et al. (2021) introduced a Swin Transformer-based image restoration model named SwinIR, which uses a shifted window mechanism to achieve better performance with fewer parameters. Wang et al. (2022) proposed Uformer, an effective and efficient U-shaped Transformer network that achieves strong image restoration performance. Yao et al. (2022) proposed a new dense residual skip-connection network for effective image denoising, using both dense residual skip connection and Enhanced Transformer to better fuse local-global features. Yang et al. (2024) proposed a two-stage denoising approach called Denoising Vision Transformers, which leverages neural fields to eliminate feature artifacts from ViTs and has been successfully applied to a series of representative ViTs. Most studies here are from empirical perspective, while only a few recent works attempt to understand the denoising effect of transformers. To understand the feature learning mechanisms behind the components of transformer-based models, Abdullaev et al. (2025) introduced a unified image processing framework that bridges transformer components and image processing algorithms. However, it is still not well understood how transformers can help with this low-level vision task.

C Experimental settings and additional results

This section presents additional experimental results and details of the settings for experiments in Section 5. For experiments with deep transformer (2.1), the experiments on synthetic-data is given in Section C.1, the experiments on MNIST is given in Section C.2, and the experiments on CIFAR-10 is given in Section C.3. For experiments with GPT-2 model, the experiments on synthetic-data is given in Section C.4, the experiments on MNIST is given in Section C.5, the experiments on CIFAR-10 is given in Section C.6, and the multi-class experiments on MNIST is given in Section C.7.

We train our models on NVIDIA RTX 4090 GPUs.

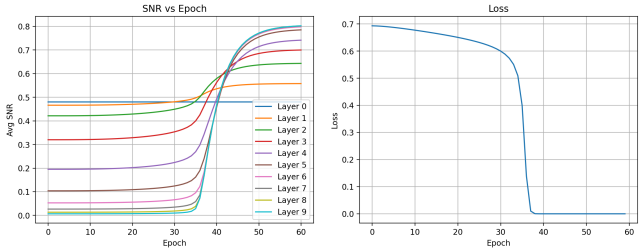
C.1 Experiments on synthetic data

In this section, additional experimental results and detailed settings for the synthetic-data experiments in Section 5.1 are presented.

Robustness to $(d, \sigma_{\text{noise}})$ variations. We further test the synthetic setup under multiple configurations by varying the ambient dimension d and the noise level σ_{noise}. Figure 7 and Figure 8 summarize the resulting layer-wise SNR profiles under different settings. Despite quantitative differences across configurations, the qualitative trend remains similar as Figure 1: The near-zero training loss observed after training indicates stable optimization under different training setups. The increase in SNR with depth after training suggests that stacking self-attention blocks yields a robust denoising effect across a wide range of synthetic regimes.

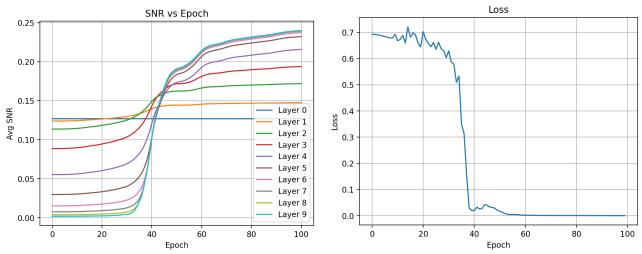
Detailed settings. In this synthetic-data experiment, we adopt the data model $\mathcal{D}$ in Definition 2.1 and the deep transformer model (2.1). The detailed settings are given as follows.

Denote the input matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$ and the label vector $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \mathbb{R}^N$ as the input matrix for data points and labels respectively. In the input matrices, a single token $\mathbf{x}_i$ is



(a) Layer-wise SNR in training. (b) Training loss curve.

Figure 7: Synthetic data experiments with different settings ($d = 200, \sigma_{\text{noise}} = 0.3, \|\boldsymbol{\mu}\|_2 = 2$): Denoising across layers. We vary the parameter settings and rerun the experiments. The training loss decreases steadily to near zero. After convergence, the deeper layers in the transformer achieve larger SNRs. The trend of layerwise SNR increasing across layers persists across configurations.



(a) Layer-wise SNR in training. (b) Training loss curve.

Figure 8: Synthetic data experiments with different settings ($d = 100, \sigma_{\text{noise}} = 2, \|\boldsymbol{\mu}\|_2 = 2.5$): Denoising across layers. We vary the parameter settings and rerun the experiments. The training loss decreases steadily to near zero. After convergence, the deeper layers in the transformer achieve larger SNRs. The trend of layerwise SNR increasing across layers persists across configurations.

generated as

$$\mathbf{x}_i = y_i \cdot \boldsymbol{\mu} + \boldsymbol{\xi}_i \in \mathbb{R}^d,\ i \in [N],$$

where $y_i \in \{\pm 1\}$. The signal vector is generated as $\boldsymbol{\mu} = \mu_{\text{scale}} d^{-1/2} \mathbf{1} \in \mathbb{R}^d$, where $\mu_{\text{scale}} = 2.5$ so that $\|\boldsymbol{\mu}\|_2 = \mu_{\text{scale}}$, and the noise $\boldsymbol{\xi}_i \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{noise}}^2 \cdot (\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2}))$, $i \in [N]$.

In our implementation, we denote the noise level by σ_p (corresponding to σ_{noise} above). The labels $y_1, \ldots, y_N$ are randomly assigned to be either $+1$ or -1 with equal probability, namely $\mathbb{P}(y_i = +1) = \mathbb{P}(y_i = -1) = 1/2$, $i \in [N]$. During implementation, we enforce a balanced split by randomly choosing $N/2$ indices to have label -1 and assigning the rest as $+1$, which reduces sampling variance. We set $d = 100$, $N = 50$, $\sigma_{\text{noise}} = 0.1$, $\mu_{\text{scale}} = 2.5$, and the number of transformer layers $L = 8$. We fix the random seed to be 44 for reproducibility.

In training, we generate $J = 1000$ i.i.d. copies of $(\mathbf{X}, \mathbf{y})$ following the above procedure, denoted by $\{\mathbf{X}_j = [\mathbf{x}_{1,j}, \ldots, \mathbf{x}_{N,j}],\ \mathbf{y}_j = [y_{1,j}, \ldots, y_{N,j}]^\top\}_{j \in [J]}$.

We train the transformer model (2.1) using full-batch SGD with momentum (equivalently, GD with momentum) from zero initialization, by minimizing the following objective:

$$L\big(\mathbf{v}, \{\mathbf{W}_l\}_{l=0}^L\big) = \frac{1}{J} \sum_{j=1}^{J} \left[\frac{1}{N} \sum_{i=1}^{N} \ell\big(y_{i,j} \cdot f(\mathbf{X}_j) \cdot \mathbf{e}_i\big) \right],$$

where $\ell(z) = \log(1 + \exp(-z))$ is the cross-entropy loss, and the constant vector $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)^\top \in \mathbb{R}^N$ where only i-th entry equals 1.

We use full-batch SGD (batch size $= J$) with momentum 0.8 and learning rate 0.0005 for 50 epochs, with zero weight decay, and clip the global gradient norm to 10. In implementation, we map labels

from ± 1 to $0, 1$ and use `BCEWithLogitsLoss`, which is equivalent to optimizing $\ell(z) = \log(1 + \exp(-z))$ up to the $0, 1$ label convention. All parameters ($\mathbf{W}_l$, $\mathbf{W}_{\text{final}}$, and $\mathbf{v}$) are initialized to zero.

We also generate an additional copy of matrices $\mathbf{X}$ as test data, and calculate the SNR achieved by the l-th layer using this test data. At the t-th iteration of GD, we input the test data $\mathbf{X}$ into the transformer model and perform a forward pass. We analyze the SNR in the outputs of each layer as follows. Denote $\mathbf{h}_{l,i}^{(t)}$ is the i-th column of the output of the l-th layer of the transformer model, trained by t GD iterations. The SNR of $\mathbf{h}_{l,i}^{(t)}$ (the i-th token in the output of the l-th layer) is defined as

$$\mathrm{SNR}_{l,i}^{(t)} = \frac{|\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2}^\top \mathbf{h}_{l,i}^{(t)}|}{\|(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2})\mathbf{h}_{l,i}^{(t)}\|_2}.$$

We denote the average SNR of all tokens produced by layer l as

$$\overline{\mathrm{SNR}}_l^{(t)} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2}^\top \mathbf{h}_{l,i}^{(t)}|}{\|(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2})\mathbf{h}_{l,i}^{(t)}\|_2}.$$

At every training epoch, we compute the layer-wise SNR on the held-out test set and report the average SNR across all tokens. The resulting layer-wise SNR curves are reported in Figure 1a. We also record the loss function value across all training iterations. Moreover, we save label-sorted attention heatmaps at three stages: (i) initialization (epoch 0), (ii) the first epoch when an "SNR flip" is detected (We declare an "SNR flip" at the first epoch when the encoder-layer SNRs become strictly increasing, i.e., $\overline{\mathrm{SNR}}_1^{(t)} < \overline{\mathrm{SNR}}_2^{(t)} < \cdots < \overline{\mathrm{SNR}}_L^{(t)}$.), and (iii) the final epoch.

For downstream evaluation, following Theorem 3.3, we probe each frozen layer representation using an ℓ_2-regularized logistic regression classifier. We construct two i.i.d. datasets: a probe-training set with $M_{\text{tr}} = 20$ samples and a held-out probe-evaluation set with $M_{\text{te}} = 1000$ samples, with matched noise levels $\sigma_{p,\text{probe}} = \sigma_{p,\text{eval}} = 3$. For each sample m, we use the last-token label $y_{N,m} \in \{\pm 1\}$ as the target and extract its last-token representation $\mathbf{h}_{l,N,m} \in \mathbb{R}^d$ from layer l (including layer 0, layer 1 to layer L, and the final outputs). The probe at layer l is trained on $\{\mathbf{h}_{l,N,m}, y_{N,m}\}_{m=1}^{M_{\text{tr}}}$ and evaluated on the held-out set of size M_{te}. The probe is a logistic regression model $\widehat{p}(y = 1 \mid \mathbf{h}) = \sigma(\mathbf{w}^\top \mathbf{h} + b)$ trained with an L-BFGS optimizer (max. 100 iterations) and an ℓ_2 penalty with coefficient $\lambda = 0.001$. In addition, we compute the layer-wise last-token SNR (using the same definition as in the main text) and compare it against the corresponding probe accuracy across layers.

C.2 Experiments on MNIST dataset

In this section, the detailed settings for the experiments on MNIST dataset (LeCun et al., 1998) in Section 5.2 are presented.

In the experiments on the MNIST dataset, each input matrix is $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d\times N}$ with token-wise labels $\mathbf{y} = [y_1, \ldots, y_N]^\top \in \{0, 1\}^N$, constructed according to the following procedure. For each token $\mathbf{x}_i$ and its label y_i, $i \in [N]$, Tokens are sampled from the filtered MNIST subset (digits `0` and `1`); within each group we enforce a balanced composition (5 zeros and 5 ones), and the token order is randomly shuffled. The clean signal $\boldsymbol{\mu}_i \in \mathbb{R}^d$ is taken as a preprocessed MNIST image from class $y_i \in \{0, 1\}$ (after global-mean subtraction and ℓ_2 rescaling described below). The noise $\boldsymbol{\xi}_i \in \mathbb{R}^d$ is sampled as $\boldsymbol{\xi}_i \sim \mathcal{N}\big(\mathbf{0}, \sigma_{\text{noise}}^2(\mathbf{I} - \frac{\boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top}{\|\boldsymbol{\mu}_i\|_2^2})\big)$, $i \in [N]$. Here $\mathbf{I} \in \mathbb{R}^{d\times d}$ denotes the identity matrix. Equivalently, in implementation we sample $z \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{noise}}^2\mathbf{I})$ and construct $\boldsymbol{\xi}_i = z - \langle z, \frac{\boldsymbol{\mu}_i}{\|\boldsymbol{\mu}_i\|_2}\rangle \frac{\boldsymbol{\mu}_i}{\|\boldsymbol{\mu}_i\|_2}$ so that $\langle \boldsymbol{\xi}_i, \boldsymbol{\mu}_i\rangle = 0$. The token $\mathbf{x}_i$ is formulated as $\mathbf{x}_i = \boldsymbol{\mu}_i + \boldsymbol{\xi}_i \in \mathbb{R}^d$, where $\boldsymbol{\mu}_i$ are signals imported from MNIST dataset and $\boldsymbol{\xi}_i$ is the noise added to the clean MNIST image.

We first filter the MNIST dataset and only keep digits `0` and `1`. All images are flattened into vectors in $\mathbb{R}^d$ with $d = 28 \times 28 = 784$. Before adding noise, we subtract the global mean image (computed from the filtered training set) from every image, and then rescale each image to have ℓ_2-norm 20.

We then pack the tokens into groups with $N = 10$ tokens per input matrix, where each group contains 5 samples from digit `0` class and 5 samples from digit `1` class, and we randomly shuffle the token order within each group. Let J denote the number of such balanced groups constructed from the dataset. From the filtered training split, we construct J balanced groups and denote them by $\{\mathbf{X}_j = [\mathbf{x}_{1,j}, \ldots, \mathbf{x}_{N,j}], \mathbf{y}_j = [y_{1,j}, \ldots, y_{N,j}]^\top\}_{j=1}^J$. We construct the held-out test set in the same way, i.e., by forming balanced groups from the filtered test split. This noisy dataset serves as the input data in the transformer model (2.1). We set the number of layers in model (2.1) as $L = 8$, the token number per group as $N = 10$, and the token dimension as $d = 784$. We normalize each clean signal token to have $\|\boldsymbol{\mu}_i\|_2 = 20$ after global mean subtraction. The orthogonal noise level is set as $\sigma_{\text{noise}} = 0.1$. We construct J balanced groups from the filtered MNIST dataset (each group contains 5 zeros and 5 ones), and use these J groups as the training set.

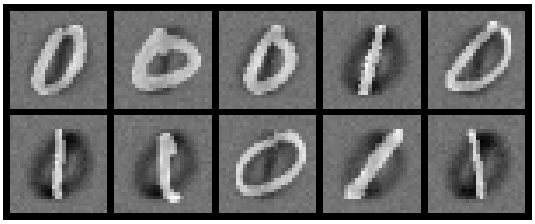


Figure 9: Noisy MNIST tokens used as inputs. Typical tokens from the binary MNIST subset after preprocessing and noise addition.

We follow a similar training procedure as that in the synthetic-data experiments. The transformer model (2.1) is trained using SGD from zero initialization to minimize the following training objective:

$$L\big(\mathbf{v}, \{\mathbf{W}_l\}_{l=0}^L\big) = \frac{1}{J}\sum_{j=1}^{J}\left[\frac{1}{N}\sum_{i=1}^{N}\ell\big(y_{i,j}\cdot f(\mathbf{X}_j)\cdot \mathbf{e}_i\big)\right], \tag{C.1}$$

where $\ell(z) = \log(1+\exp(-z))$ is the cross-entropy loss, and the constant vector $\mathbf{e}_i = (0,\ldots,0,1,0,\ldots,0)^\top \in \mathbb{R}^N$ where only i-th entry equals 1. In implementation, we use `BCEWithLogitsLoss` with labels in $\{0,1\}$ applied to the token-wise logits produced by the model, which is equivalent to the logistic loss up to the standard label mapping between $\{0,1\}$ and $\{\pm 1\}$.

We train the transformer model (2.1) using SGD with learning rate 0.000005, momentum 0.9, and weight decay 0.0001. The batch size is 64 (each batch contains 64 groups), and we train for 40 epochs. We use gradient clipping with maximum norm 5. All parameters are initialized as zeros (including all $\mathbf{W}_l$, $\mathbf{W}_{\text{final}}$, and the classifier vector $\mathbf{v}$).

In the previous synthetic model, the signal vector is a fixed vector $\boldsymbol{\mu}$ shared across samples. For real images, however, token representations within a group are aggregated by self-attention, and the outputs at a given token position may mix information from multiple inputs. Therefore, an SNR defined with respect to the *per-token* clean direction $\boldsymbol{\mu}_i$ would primarily measure instance-level fidelity and can become less informative after deep mixing. To obtain a stable measure of class-discriminative signal that is comparable across tokens and layers, we compute SNR with respect to the *class-mean direction* of the token label.

Concretely, let $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1 \in \mathbb{R}^d$ be the average of preprocessed training images in each class, normalized to unit ℓ_2 norm. For token i with label $y_i \in \{0,1\}$ at layer l, we define its SNR as

$$\mathrm{SNR}_{l,i}^{(t)} = \frac{\left|\boldsymbol{\mu}_{y_i}^\top \mathbf{h}_{l,i}^{(t)}\right|}{\left\|\mathbf{h}_{l,i}^{(t)} - \left(\boldsymbol{\mu}_{y_i}^\top \mathbf{h}_{l,i}^{(t)}\right)\boldsymbol{\mu}_{y_i}\right\|_2}. \tag{C.2}$$

The layer-wise average SNR is then

$$\overline{\mathrm{SNR}}_l^{(t)} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\boldsymbol{\mu}_{y_i}^\top \mathbf{h}_{l,i}^{(t)}\right|}{\left\|\mathbf{h}_{l,i}^{(t)} - \left(\boldsymbol{\mu}_{y_i}^\top \mathbf{h}_{l,i}^{(t)}\right)\boldsymbol{\mu}_{y_i}\right\|_2}. \tag{C.3}$$

This class-mean-based definition does not change the qualitative conclusions of our experiments: our theoretical prediction concerns progressive amplification of task-relevant signal while suppressing orthogonal components, and in the real-data setting the class-mean directions provide a natural proxy for such signal. Empirically, the resulting $\overline{\mathrm{SNR}}_l^{(t)}$ increases with depth and closely tracks downstream linear separability (logistic regression probing), supporting the same denoising interpretation.

In the downstream logistic regression task (Theorem 3.3), we freeze the trained transformer and train an ℓ_2-regularized logistic regression probe on features extracted from different layers. We use MNIST binary classification (digits 0 vs. 1) with input dimension $d = 28 \times 28 = 784$ and group size $N = 10$ tokens per input matrix. For the downstream dataset, we sample $M_{\mathrm{tr}} = 13$ groups from the training split and $M_{\mathrm{te}} = 1000$ groups from the test split, and add orthogonal Gaussian noise with $\sigma_{\mathrm{DS}} = 6$ to both training and test features. For each layer $l \in \{0, 1, \ldots, L, L+1\}$ with $L = 8$, we extract the last-token representation $\mathbf{h}_{l,N,m}$ from model (2.1) and form probe training pairs $\{(\mathbf{h}_{l,N,m}, y_{N,m})\}_{m=1}^{M_{\mathrm{tr}}}$. The downstream model is a single linear classifier $\widehat{p}(y = 1 \mid \mathbf{h}) = \sigma(\mathbf{w}^\top \mathbf{h} + b)$ trained with L-BFGS (step size 1.0, 100 iterations) using binary cross-entropy and ℓ_2 penalty $\lambda = 0.001$; we standardize features per layer using the probe training split statistics and report test accuracy on the held-out set.

C.3 Experiments on CIFAR-10 dataset

In this section, we present the experimental results and detailed settings for the experiments on CIFAR-10 dataset (Krizhevsky et al., 2009).

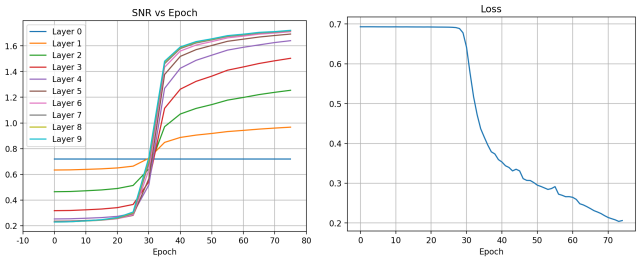
Problem setup (binary CIFAR-10 with noisy tokens). We construct a binary classification task by selecting two visually distinct CIFAR-10 classes, `airplane` and `frog`. Each token is formed by flattening a CIFAR-10 image into a vector in $\mathbb{R}^d$ (with $d = 3 \times 32 \times 32$), after global mean subtraction and per-image ℓ_2 normalization to a fixed scale, so that signal magnitudes are controlled across samples. Then, a single transformer input is created by packing N such tokens into a matrix, using balanced groups (equal number of tokens per class) with a random within-group permutation. Finally, we add Gaussian noise projected onto the subspace orthogonal to each clean image's direction, producing the noisy tokens fed into the model. Figure 10 visualizes typical noisy CIFAR-10 tokens used as model inputs.



Figure 10: Noisy CIFAR-10 tokens used as inputs. Typical images from the CIFAR-10 subset after preprocessing and noise addition.

Layer-wise SNR changes and training stability. During training, we compute the layer-wise SNR (C.3) measured at the outputs of each layer on a test set. The SNR trajectories across epochs for each layer (layer 0–9) in Figure 11a shows that deeper layers consistently attain higher SNRs after convergence, indicating a progressive enhancement of signal components and suppression of noise, in line with our denoising predictions. Meanwhile, the training loss in Figure 11b decreases smoothly over epochs, corroborating stable optimization under the same run.

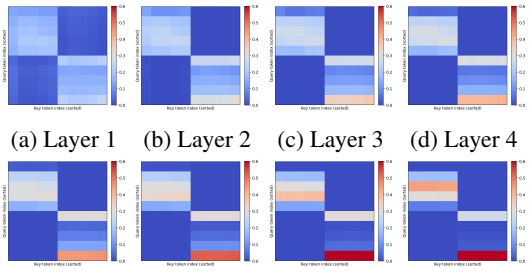
Attention patterns (sorted heatmaps across layers). To gain mechanistic insight, we visualize the Softmax attention matrices across layers. We reorder tokens within each sample by their class before visualization, so that ideal within-class aggregation would appear as a block-diagonal structure. Figure 12 shows the heatmaps for layers 1–8. Compared with MNIST, CIFAR-10 tokens exhibit substantially great intra-class diversity, resulting in attention patterns that are less strictly block-diagonal. Nevertheless, deeper layers still show increased focus on within-class interactions,



(a) Layer-wise SNR in training. (b) Training loss curve.

Figure 11: CIFAR-10: Denoising across layers. The training loss decreases stably during training, and the outputs of deeper layers attain higher SNRs after training.

often concentrating on a subset of more reliable tokens, which aligns with the layer-wise SNR improvements in Figure 11a.

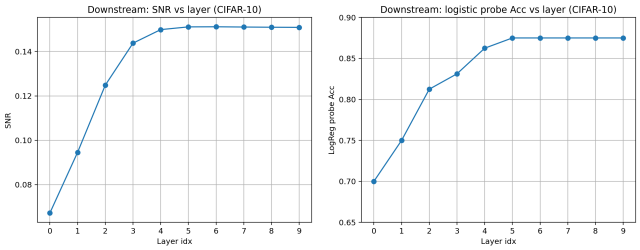


(a) Layer 1 (b) Layer 2 (c) Layer 3 (d) Layer 4

(e) Layer 5 (f) Layer 6 (g) Layer 7 (h) Layer 8

Figure 12: CIFAR-10: Token-sorted attention heatmaps across layers (layer 1–8). Deeper layers still show increased emphasis on within-class and more reliable token interactions.

Downstream evaluation (logistic regression). Finally, we evaluate representation quality using an ℓ_2-regularized logistic regression probe trained on layer-wise features. In the binary classification CIFAR-10 task (`airplane` vs. `frog`), for each layer l we extract the last-token representation as the feature and train a linear classifier on a small downstream training set, reporting test accuracy on a larger held-out set. Figure 13b shows that probe accuracy improves with depth, and the depth profile closely follows the layer-wise SNR trend in Figure 11a. Overall, layers with higher SNR yield more linearly separable representations for this task, consistent with the view that deeper attention blocks progressively suppress orthogonal noise while preserving class-discriminative signal.



(a) Layer-wise last-token SNR. (b) Layer-wise test accuracy.

Figure 13: CIFAR-10: Downstream evaluation. Features from deeper layers exhibit higher SNRs and yield better test accuracy when evaluated with a logistic regression probe.

Detailed settings. We construct noisy token groups from CIFAR-10 (Krizhevsky et al., 2009) following the same real-data pipeline as in Appendix C.2, with class-specific choices detailed below. We retain two CIFAR-10 classes, `airplane` (id=0) and `frog` (id=6), and form a binary classification task. Following the implementation, we map class id 0 (`airplane`) to label 0 and

class id 6 (`frog`) to label 1. The tokens in the input data matrix contain noisy versions of images from CIFAR-10 dataset. Each image is converted to a tensor in $[0, 1]$, flattened into $\mathbb{R}^d$ with $d = 3 \times 32 \times 32 = 3072$, and we form token groups of size $N = 10$. Within each group, we enforce a balanced composition (5 tokens per class) and randomly shuffle token order. After global mean subtraction, we rescale each clean token to have $\|\boldsymbol{\mu}_i\|_2 = 20$, and then add orthogonal Gaussian noise with $\sigma_{\text{noise}} = 0.1$. We adopt the following preprocessing procedure before inputting the data into the transformer model (2.1): (i) we compute a global mean vector from the filtered training set and center both train/test tokens by subtracting this mean; (ii) we normalize each centered token to have $\|\boldsymbol{\mu}_i\|_2 = 20$; (iii) we form each input matrix by grouping $N = 10$ tokens with balanced composition (5 tokens from each class) and shuffle within each group; (iv) we add Gaussian noise projected onto the subspace orthogonal to each token direction with noise level $\sigma_{\text{noise}} = 0.1$. (v) (center filtering) we optionally perform an intra-class center filtering step that keeps the top fraction of samples in each class according to alignment with the class-mean direction (details below).

Intra-class center filtering. After global centering and ℓ_2 rescaling, we compute unit class-mean directions from the training split and score each token by $\langle x, \mu_y \rangle$. For each class, we keep the top fraction `CENTER_KEEP_FRAC` $= 0.8$ of samples and discard the rest; we apply the same filtering procedure to both the training and test splits for consistency with our implementation.

We follow a similar training procedure as that in the synthetic-data experiments. The transformer model (2.1) is trained using SGD from zero initialization to minimize the training objective (C.1). We train the model for 75 epochs using SGD with learning rate 0.000005, momentum 0.9, and weight decay 0.0001. In the filtered CIFAR-10 training split, each class provides 5000 images; with 5 tokens per class in each group. The batch size is 128 groups, and we apply gradient clipping with max norm 5. We also fix the random seed to 4235 for reproducibility across data grouping, filtering, and optimization. The objective is implemented as token-wise `BCEWithLogitsLoss` applied to the per-token logits and averaged over tokens and groups.

During training, we compute layer-wise SNR on the held-out test set using the same real-data definition as in the MNIST experiments (Appendix C.2), where the reference direction is the unit class-mean vector $\boldsymbol{\mu}_y$ for each token label. This class-mean-based SNR serves as a stable proxy of task-relevant signal under token mixing by self-attention, and thus preserves the qualitative conclusions about progressive signal amplification and suppression of orthogonal components. We average SNR over tokens to obtain $\overline{\text{SNR}}_l^{(t)}$ for each layer and evaluate it every 5 epochs; the resulting trajectories are reported in Figure 11a. For attention visualization, we take a representative test group, reorder tokens by ground-truth labels, and plot the Softmax attention matrices across layers (including the final extra attention block), producing Figure 12.

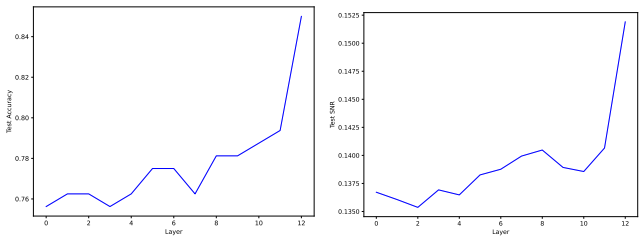
For the downstream logistic regression task, following the settings in Theorem 3.3, we sample $M_{\text{tr}} = 14$ groups for probe training and $M_{\text{te}} = 1000$ groups for probe evaluation, and for each group we extract the last-token representation from every layer as probe features. We use $d = 3072$ and $\|\boldsymbol{\mu}\|_2 = 20$ after preprocessing; for downstream probing we add orthogonal noise at the input-token level with $\sigma_{\text{DS,tr}} = \sigma_{\text{DS,te}} = 3$ and use one noise draw per group (i.e., `DS_COPIES_TR` $=$ `DS_COPIES_TE` $= 1$). For each layer, we fit a logistic regression probe $\widehat{p}(y = 1 \mid \mathbf{h}) = \sigma(\mathbf{w}^\top \mathbf{h} + b)$ on standardized features (mean/std computed on the probe-training split of that layer) using L-BFGS with strong Wolfe line search (learning rate 1.0, max. 100 iterations) and ℓ_2 penalty $\lambda = 0.001$, and report probe test accuracy across layers. In addition, we compute the last-token SNR for each layer (using the same class-mean-based definition) and compare it against the layer-wise probe accuracy, as reported in Figure 13b.

C.4 Experiments with GPT-2 on synthetic data

In this section, experimental results and detailed settings for the experiments with GPT-2 model (Radford et al., 2019) on synthetic data are presented.

We adopt the GPT-2 model (Radford et al., 2019) architecture in this experiment. The GPT-2 model is pre-trained on synthetic-data following the data model $\mathcal{D}$ in Definition 2.1. In the following downstream logistic regression task, we utilize outputs from different GPT-2 layers as training data for the linear model. For these experiments on outputs from different layers of GPT-2 model, we compare their test accuracy and generalized test SNR in Figure 14. It is generally observed in Figure 14 that the outputs from deeper layers in pre-trained GPT-2 have higher generalized test

SNRs and retain higher test accuracy in the corresponding downstream logistic regression task. This indicates that deeper layers of GPT-2 model preserve relatively more signals and achieve better performance on the downstream task. This verifies our theoretical results in Theorem 3.3 in more general scenarios.



(a) Layer-wise test accuracy (b) Layer-wise test SNR

Figure 14: GPT-2 on synthetic data: Both downstream test accuracy and generalized test SNR show increasing trend across layers.

Detailed settings. Following Garg et al. (2022), we select the standard GPT-2 architecture with embedding size 64, the number of attention heads 2, and the number of layers 12. The GPT-2 model is pre-trained on binary classification task on the synthetic-data model $\mathcal{D}$ defined in Definition 2.1 with $d = 200, \|\boldsymbol{\mu}\|_2 = 20, \sigma_{\text{noise}} = 2,\ N = 11$. We pre-train the GPT-2 model using the Adam optimizer, learning rate 0.0001, batch size 64, training steps 500000, number of training data 1000. And we adopt a similar curriculum learning strategy as in Garg et al. (2022) to accelerate the pre-training. We zero out all except the first d_{cur} dimensions of the x and the tokens of size k_{cur} in the pre-training. Starting from $d_{\text{cur}} = 5$, $k_{\text{cur}} = 11$, d_{cur} and k_{cur} increase by 1 and 2 every 2000 steps respectively until $d_{\text{cur}} = d,\ k_{\text{cur}} = 41$. The training loss is presented in Figure 15, and it is shown in Figure 15 that the training loss decreases steadily toward near zero, which indicates that the model is effectively pre-trained.

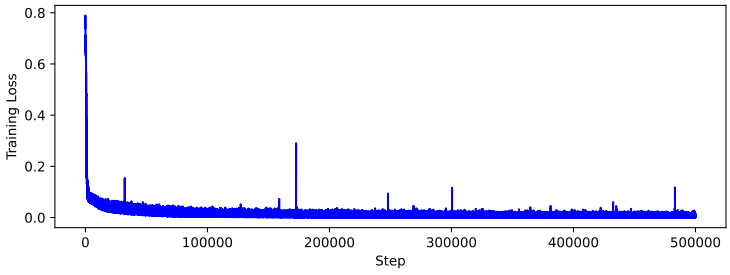


Figure 15: **Experiments with GPT-2 on synthetic data: the training loss curve in the pre-training.**

In the downstream task, we follow the setting of Theorem 3.3, and adopt linear architectures for the downstream logistic regression task. We generate M independent copies of $\mathbf{X}_m$, based on the synthetic-data model $\mathcal{D}$ in Definition 2.1, and feed the matrices $\mathbf{X}_m,\ m \in [M]$ into the pre-trained GPT-2 model. For each layer l, we utilize the first column token in the outputs as the training dataset for logistic regression, namely $\{(\mathbf{h}_{l,1,m}, y_{1,m})\},\ m \in [M]$. We set $d = 200,\ \|\boldsymbol{\mu}\|_2 = 20,\ \sigma_{\text{noise}} = 12.5,\ N = 11$, the training sample size $M = 240$.

The downstream logistic regression model is trained by SGD optimizer with learning rate 0.0035, weight decay 0.009, batch size 32, for 100 epochs. For these experiments using outputs from different layers of GPT-2 model, we compare downstream task test accuracy and generalized SNR as

defined in (C.4),

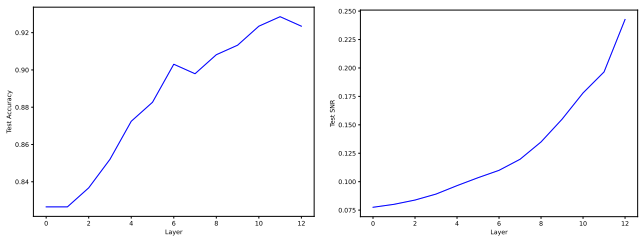
$$\widetilde{\mathrm{SNR}}_l^{(t)} = \frac{1}{M_0}\sum_{m=1}^{M_0} \frac{|\frac{\theta_l^{(t)}}{\|\theta_l^{(t)}\|_2}^\top \mathbf{h}_{l,1,m}^{(t)}|}{\|(\mathbf{I} - \frac{\theta_l^{(t)}\theta_l^{(t)\top}}{\|\theta_l^{(t)}\|_2^2})\mathbf{h}_{l,1,m}^{(t)}\|_2}, \tag{C.4}$$

on a test set of size $M_0 = 160$. The results across layers ($l = 0, 1, \ldots, 12$) are presented in Figure 14.

C.5 Experiments with GPT-2 on MNIST dataset

In this section, the experimental results and detailed settings for the experiments with GPT-2 model (Radford et al., 2019) on MNIST dataset are presented.

We first pre-train GPT-2 model on MNIST, following a similar preprocessing procedure as employed in experiments in Section 5.2. In the following downstream logistic regression task, we use outputs from different GPT-2 layers as training data for the linear model. For these experiments on outputs from different GPT-2 layers, Figure 16 compares their test accuracy and generalized test SNR. It is shown in Figure 16 that the outputs from deeper GPT-2 layers have higher generalized test SNRs and achieve higher test accuracies in the downstream task. This indicates that deeper layers in GPT-2 model can effectively preserve more signals and achieve better downstream performance, verifying our theoretical results in Theorem 3.3 in general scenarios.



(a) Layer-wise test accuracy (b) Layer-wise test SNR

Figure 16: GPT-2 on MNIST: Both the generalized test SNR and the downstream task test accuracy increase across layers.

Detailed settings. The GPT-2 model is pre-trained on binary classification task on MNIST dataset. Similar as experiments in Section C.2, we perform binary classification on two MNIST classes: digit `0` class and `1` class (with their labels set at 0 and 1 respectively). Similar preprocessing procedure are adopted: (i) subtract the global mean from all training and test tokens; (ii) normalize each token to $\|\boldsymbol{\mu}_i\|_2 = 20$; (iii) form input matrices of token size N with a balanced class composition; (iv) add orthogonal Gaussian noise with noise level σ_{noise} Here we set $d = 784,\ \|\boldsymbol{\mu}\|_2 = 20,\ \sigma_{\text{noise}} = 0.5,\ N = 10$.

Following Garg et al. (2022), we select the standard GPT-2 architecture with embedding size 256, the number of attention heads 2, and the number of layers 12. We pre-train the GPT-2 model using the Adam optimizer with learning rate 0.0001, batch size 64, training steps 500000, number of training data 1000. And we adopt a similar curriculum learning strategy as in Garg et al. (2022) to accelerate the pre-training. We zero out all except the first d_{cur} dimensions of the x in the pre-training. Start from $d_{\text{cur}} = 5$, and increase d_{cur} by 1 every 2000 steps until $d_{\text{cur}} = d$. The training loss curve is presented in Figure 17, which shows that the training loss decreases steadily to around zero, which indicates that the model is already well pre-trained.

In the downstream tasks, we follow the setting in Theorem 3.3 and adopt linear architecture for the downstream logistic regression task. We follow a similar preprocessing procedure for MNIST dataset as described above. We generate M independent copies of $\mathbf{X}_m$ and input the matrices $\mathbf{X}_m,\ m \in [M]$ into the pre-trained GPT-2 model. For each layer l, we use the first column token in the outputs as the training dataset for logistic regression, namely $\{(\mathbf{h}_{l,1,m}, y_{1,m})\},\ m \in [M]$. We set the $d = 784,\ \|\boldsymbol{\mu}\|_2 = 20,\ \sigma_{\text{noise}} = 4,\ N = 10$, the training sample size $M = 64$.

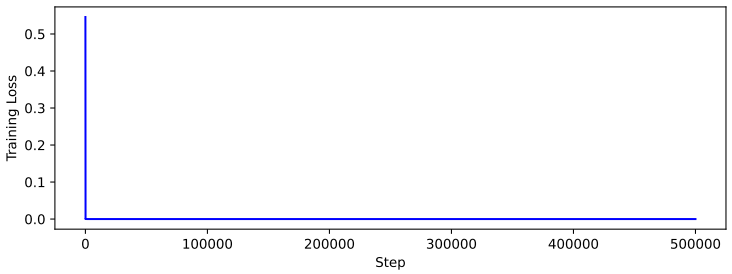


Figure 17: **Experiments with GPT-2 on MNIST dataset: the training loss curve in the pre-training.**

The downstream logistic regression model is trained by Adam optimizer with learning rate 0.0008, weight decay 0.0008, batch size 64, training for 130 epochs. The learning rate adopts a linear warm-up strategy for the first 30 epochs. For these experiments with outputs from different layers of GPT-2 model, we compare test accuracy and generalized SNR (C.4) on a test set of size $M_0 = 196$. The results across layers ($l = 0, 1, \ldots, 12$) are presented in Figure 16.

C.6 Experiments with GPT-2 on CIFAR-10 dataset

In this section, experimental results and detailed settings for the experiments with GPT-2 model (Radford et al., 2019) on CIFAR-10 dataset (Krizhevsky et al., 2009) are presented.

Detailed settings. The GPT-2 model is first pre-trained on binary classification task on CIFAR-10 dataset. Similar as experiments in Section C.3, we perform binary classification on two CIFAR-10 classes: `airplane` (id=0, mapped to label 0 as `CLASS_A`) and `frog` (id=6, mapped to label 1 as `CLASS_B`). Similar preprocessing procedure are adopted: (i) subtract the global mean from all training and test tokens; (ii) normalize each token to $\|\boldsymbol{\mu}_i\|_2 = 20$; (iii) form input matrices of token size N with a balanced class composition; (iv) add orthogonal Gaussian noise with noise level σ_{noise}. Here we have $d = 3072, \|\boldsymbol{\mu}_i\|_2 = 20, \sigma_{\text{noise}} = 0.5, N = 10$.

Following Garg et al. (2022), we select the standard GPT-2 architecture with embedding size 512, the number of attention heads 2, and the number of layers 12. We pre-train the GPT-2 model using the Adam optimizer with learning rate 0.0001, batch size 64, training steps 200000, number of training data 1000. And we adopt a similar curriculum learning strategy as in Garg et al. (2022) to accelerate the pre-training. We zero out all except the first d_{cur} dimensions of the x in the pre-training. Starting from $d_{\text{cur}} = 5$, we increase d_{cur} by 1 every 20 steps until $d_{\text{cur}} = d$. The training loss curve is presented in Figure 18, which shows that the training loss decreases steadily to around 0, which indicates that the model is already well pre-trained.

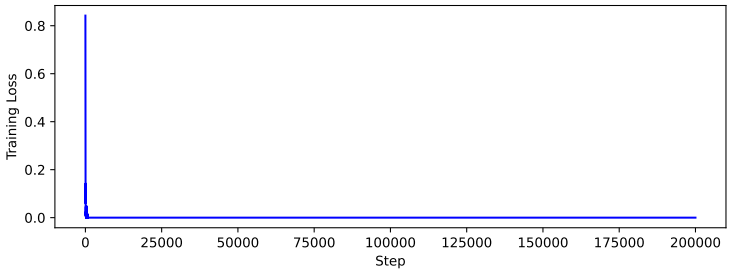
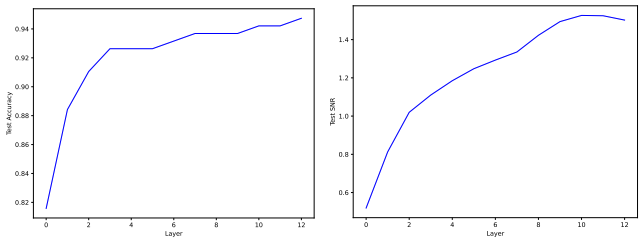


Figure 18: **Experiments with GPT-2 on CIFAR-10 dataset: the training loss in the pre-training.**

In the downstream logistic regression task, we follow the setup of Theorem 3.3 and adopt a linear model for the downstream task. We follow a similar preprocessing procedure for CIFAR-10 as described above, and filter out 5% of the images in each class that are farthest from the class mean.

We generate M independent copies of $\mathbf{X}_m$, and feed the matrices $\mathbf{X}_m,\ m \in [M]$ into the pre-trained GPT-2 model. For each layer l, we utilize the first column token in the outputs as the training dataset for downstream logistic regression, namely $\{(\mathbf{h}_{l,1,m}, y_{1,m})\},\ m \in [M]$. Here we set the $d = 3072,\ \|\boldsymbol{\mu}\|_2 = 20,\ \sigma_{\text{noise}} = 0.2,\ N = 10$, the training sample size $M = 64$.

The downstream logistic regression model is trained by Adam optimizer with learning rate 0.00045, weight decay 0.005. The learning rate adopts the linear warm-up for first 20 epochs and cosine annealing strategy to the minimum learning rate 0.00005. Gradient clipping with the maximum norm of 1.0 is applied. The training batch size is 64 and the model is trained for 60 epochs. We adopt an early stopping strategy if neither the test loss nor the test accuracy improves for 8 consecutive epochs. For these experiments using outputs from different layers of GPT-2 model, we compare test accuracy and generalized SNR (C.4) on a test set of size $M_0 = 190$, and results across layers ($l = 0, 1, \ldots, 12$) are presented in Figure 19. Figure 19 shows that outputs from deeper GPT-2 layers exhibit higher SNRs and achieve higher test accuracies in the downstream logistic regression task. This indicates that deeper layers of GPT-2 model capture more signals and achieve better downstream performance, validating our theoretical results in Theorem 3.3 in more practical scenarios.



(a) Layer-wise test accuracy (b) Layer-wise test SNR

Figure 19: GPT-2 on CIFAR-10: Both downstream task test accuracy and generalized test SNR increase across layers.

C.7 Experiments on multi-class classification problem (GPT-2, MNIST)

In this section, to demonstrate that our theoretical findings hold in general scenarios, we present experimental results and detailed settings for experiments with GPT-2 on multi-class (10-class) classification task with MNIST dataset

The GPT-2 model is pre-trained on multi-class classification task on MNIST dataset. Similar as experiments in Section C.2, we perform 10-class classification on MNIST dataset. The preprocessing procedure adopted in this experiment are as follows: (i) normalize each token to $\|\boldsymbol{\mu}_i\|_2 = 20$; (ii) form input matrices of token size N. Here we set $d = 784,\ \|\boldsymbol{\mu}\|_2 = 20,\ N = 10$.

Following Garg et al. (2022), we select the standard GPT-2 architecture with embedding size 256, the number of attention heads 2, and the number of layers 12. We pre-train the GPT-2 model using the Adam optimizer with learning rate 0.0001, batch size 64, training steps 80000, number of training data 1000. And we adopt a similar curriculum learning strategy as in Garg et al. (2022) to accelerate the pre-training. We zero out all except the first d_{cur} dimensions of the x in the pre-training. Start from $d_{\text{cur}} = 5$, and increase d_{cur} by 1 every 100 steps until $d_{\text{cur}} = d$. The training loss curve is presented in Figure 20, which shows that the training loss decreases steadily to around zero, which indicates that the model is already well pre-trained.

In the downstream tasks, we follow similar setting in Theorem 3.3 and adopt linear architecture for the downstream logistic regression task. We follow a similar preprocessing procedure for MNIST dataset as described above. We generate M independent copies of $\mathbf{X}_m$ and input the matrices $\mathbf{X}_m,\ m \in [M]$ into the pre-trained GPT-2 model. For each layer l, we use the first column token in the outputs as the training dataset for logistic regression, namely $\{(\mathbf{h}_{l,1,m}, y_{1,m})\},\ m \in [M]$. We set the $d = 784,\ \|\boldsymbol{\mu}\|_2 = 20,\ N = 10$, the training sample size $M = 64$.

The downstream logistic regression model is trained by Adam optimizer with learning rate 0.0008, weight decay 0.0008, batch size 64, training for 300 epochs. Gradient clipping with the maximum

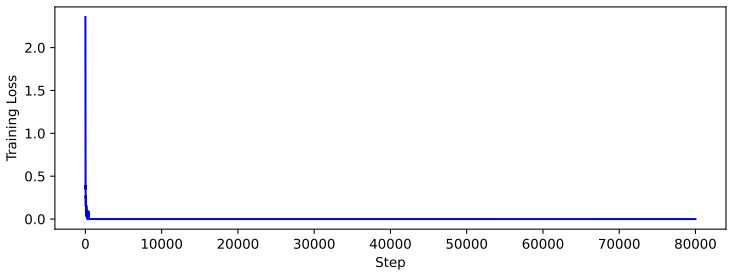


Figure 20: **Experiments with GPT-2 on 10-class full MNIST dataset: the training loss curve in the pre-training.**

norm of 5.0 is applied. The learning rate adopts a linear warm-up strategy for the first 10 epochs. We adopt an early stopping strategy if neither the test loss nor the test accuracy improves for 20 consecutive epochs.

For these experiments with outputs from different layers of GPT-2 model, we compare test accuracy on a test set of size $M_0 = 1000$. The results across layers ($l = 0, 1, \ldots, 12$) are presented in Figure 21. It is shown in Figure 21 that the outputs from deeper GPT-2 layers achieve higher test accuracies in the downstream task. This demonstrates that deeper layers in GPT-2 model more effectively preserve signals and achieve better downstream performance, thereby confirming that our theoretical insights in Theorem 3.3 extends to general multi-class scenarios.

The multi-class experimental results are consistent with the binary classification experiments, further supporting the validity of our theoretical insights in general scenarios.

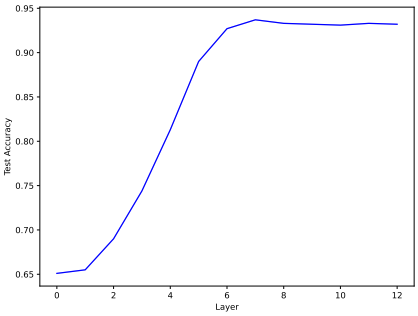


Figure 21: **Experiments with GPT-2 on 10-class MNIST dataset:** Layer-wise test accuracy across layers from the GPT-2 model.

D Proof of the first part of Theorem 3.2: training convergence

In this section, we prove the first point in Theorem 3.2 in Section 3, regarding the training convergence of the deep transformer (2.1).

We first introduce the decomposition of each layer's outputs in the deep transformer (2.1) by defining the following notations. In the outputs $\mathbf{H}_l^{(t)}$ of layer l, $l = 0, 1, \ldots, L$, any token $\mathbf{h}_{l,i}^{(t)}, i \in [N]$ can be decomposed to two parts: the projection of $\mathbf{h}_{l,i}^{(t)}$ onto the direction of $\boldsymbol{\mu}$, and the projection of $\mathbf{h}_{l,i}^{(t)}$ onto the perpendicular space of $\boldsymbol{\mu}$, namely

$$\mathbf{h}_{l,i}^{(t)} = \langle \mathbf{h}_{l,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2} \rangle \cdot \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2} + (\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2}) \mathbf{h}_{l,i}^{(t)}. \tag{D.1}$$

To decompose $\mathbf{h}_{l,i}^{(t)}$, we denote $\alpha_{l,i}^{(t)}$ the coefficient of $\boldsymbol{\mu}$ in $\mathbf{h}_{l,j}^{(t)}$ at t-th iteration, and $\beta_{l,i,k}^{(t)}$ as the coefficient of $\boldsymbol{\xi}_k$ in $\mathbf{h}_{l,i}^{(t)}$ at t-th iteration, $i, k \in [N], l = 0, 1, \ldots, L$, namely

$$\mathbf{h}_{l,i}^{(t)} = \alpha_{l,i}^{(t)}\boldsymbol{\mu} + \sum_{k=1}^{N}\beta_{l,i,k}^{(t)}\boldsymbol{\xi}_k, \tag{D.2}$$

with the input at layer $l = 0$ as

$$\alpha_{0,i}^{(t)} = y_i, \qquad \beta_{0,i,k}^{(t)} = \begin{cases} 1, k = i \\ 0, k \neq i, \end{cases}$$

for all $t \geq 0$. Also, $\mathbf{h}_{l,i}^{(t)}$ can be represented as

$$\langle \mathbf{h}_{l,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2} \rangle = \alpha_{l,i}^{(t)}\|\boldsymbol{\mu}\|_2, \qquad (\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2})\mathbf{h}_{l,i}^{(t)} = \sum_{k=1}^{N}\beta_{l,i,k}^{(t)}\boldsymbol{\xi}_k. \tag{D.3}$$

Following the decomposition in (D.1) and above notations, the general definition of SNR for the outputs of layer l (2.6) can be further expressed as follows.

$$\mathrm{SNR}_l^{(t)} = \frac{|\langle \mathbf{h}_{l,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2}\rangle|}{\|(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2})\mathbf{h}_{l,i}^{(t)}\|_2} = \frac{|\alpha_{l,i}^{(t)}| \cdot \|\boldsymbol{\mu}\|_2}{\|\sum_{k=1}^{N}\beta_{l,i,k}^{(t)}\boldsymbol{\xi}_k\|_2}, \tag{D.4}$$

where the last equality is by the definition of the coefficients $\alpha_{l,i}^{(t)}$ and $\beta_{l,i,k}^{(t)}$ in (D.3). This general definition of SNR allows us to study how the captured signals in the features evolve across layers in the transformer.

For any $i \in [N]$, denote the following sets:

$$\mathcal{I}_{y_i} = \{j | j \in [N], y_j = y_i\},\ \mathcal{I}_{\overline{y_i}} = \{j | j \in [N], y_j = -y_i\}. \tag{D.5}$$

In model (2.1), denote the Softmax matrix at layer l at iteration t as $\mathbf{S}_l^{(t)} := \mathrm{Softmax}({\mathbf{H}_l^{(t)}}^\top \mathbf{W}_l^{(t)}\mathbf{H}_l^{(t)})$, then the entry of the Softmax matrix $\mathbf{S}_l^{(t)}$ is given as

$$[\mathbf{S}_l^{(t)}]_{ji} = \frac{\exp({\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)}\mathbf{h}_{l,i}^{(t)})}{\sum_{j'=1}^{N}\exp({\mathbf{h}_{l,j'}^{(t)}}^\top \mathbf{W}_l^{(t)}\mathbf{h}_{l,i}^{(t)})} = \begin{cases} a_{l,j,i}^{(t)}, \text{ for } j \in \mathcal{I}_{y_i} \\ b_{l,j,i}^{(t)}, \text{ for } j \in \mathcal{I}_{\overline{y_i}}, \end{cases}$$

where j, i refer to j-th row and i-th column entry in the Softmax matrix, and we denote $a_{l,j,i}^{(t)}$ to represent the entries for $j \in \mathcal{I}_{y_i}$ and $b_{l,j,i}^{(t)}$ for $j \in \mathcal{I}_{\overline{y_i}}$ respectively.

Training results analysis. The first point of Theorem 3.2 presents the training convergence of the deep attention-only transformers (2.1) with the gradient descent updates (2.4)-(2.5). We prove the first point of Theorem 3.2 by proving Lemmas D.1, D.2, D.3, D.4, and D.5 below.

Lemma D.1 presents the training results at the second iteration. For the training dynamics after the second iteration, Lemmas D.2 and D.3 demonstrate the error bounds for $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ and $\mathbf{v}$ at iteration t. Lemma D.4 presents the bounds for the changes of parameters $a_{l,j,i}^{(t)}, b_{l,j,i}^{(t)}, \alpha_{l,i}^{(t)}, \beta_{l,i,k}^{(t)}$ at iteration t. Lemma D.5 presents the bound for the population loss function (2.3) after iteration t.

The following Lemma D.1 presents the trained $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ and $\mathbf{v}$ after the second iteration.

Lemma D.1. *Under Condition 3.1, in deep attention-only transformer* (2.1)*, the trained* $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ *and* $\mathbf{v}$ *after the second iteration under gradient descent updates* (2.4) *and* (2.5) *are as follows,*

$$\mathbf{v}^* = (\frac{1}{2} + r_1)\frac{2^{L+1} - 1 + N}{N}\eta\boldsymbol{\mu},$$

and for $l = 0, \ldots, L$*,*

$$\mathbf{W}_l^* = r_1\frac{N-1}{2N^3}(2^{L+1} - 1 + N)\eta^2\left\{\left[\frac{(N-2)(2^L - 1) + N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L + N)\sigma_{\mathrm{noise}}^2\right]\boldsymbol{\mu}\boldsymbol{\mu}^\top + \|\boldsymbol{\mu}\|_2^2\sigma_{\mathrm{noise}}^2\mathbf{I}_d\right\}, \tag{D.6}$$

where $r_1 = 1/[\exp(\frac{2^{L+1}-1+N}{2N}\eta\|\boldsymbol{\mu}\|_2^2) + 1]$.

Denote the error term of $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ and $\mathbf{v}^{(t)}$ at iteration t as $\mathbf{W}_{l,\text{err}}^{(t)} = \mathbf{W}_l^{(t)} - \mathbf{W}_l^*$, $\mathbf{v}_{\text{err}}^{(t)} = \mathbf{v}^{(t)} - \mathbf{v}^*$. The bound of the norm of the error term of $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ and $\mathbf{v}$ are presented in Lemmas D.2 and D.3.

Lemma D.2. *Define $T^* = \text{poly}(\|\boldsymbol{\mu}\|_2, N, 2^L, d) \geq 2$, under Condition 3.1, for all $2 \leq t \leq T^*$, it holds that the bound of the norm for the error term of $\mathbf{v}$ is*

$$\|\mathbf{v}_{\text{err}}^{(t)}\|_2 \leq \exp(-c_3 \cdot 2^L), \tag{D.7}$$

where $c_3 > 0$ is a positive constant.

Lemma D.3. *Define $T^* = \text{poly}(\|\boldsymbol{\mu}\|_2, N, 2^L, d) \geq 2$, under Condition 3.1, for all $2 \leq t \leq T^*$, it holds that the bound of the norm for the error term of $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ are as follows,*

$$\|\mathbf{W}_{l,\text{err}}^{(t)}\|_2 \leq \exp(-c_4 \cdot 2^L),\ l = 0, \ldots, L, \tag{D.8}$$

where $c_4 > 0$ is a positive constant.

Lemma D.4. *Define $T^* = \text{poly}(\|\boldsymbol{\mu}\|_2, N, 2^L, d) \geq 2$, then for all $2 \leq t \leq T^*$, it holds that*

$$|a_{l,j,i}^{(t+1)} - a_{l,j,i}^{(t)}| \leq 2\exp(-c_5 \cdot 2^L) \cdot a_{l,j,i}^{(t)}, \tag{D.9}$$

$$|b_{l,j,i}^{(t+1)} - b_{l,j,i}^{(t)}| \leq 2\exp(-c_5 \cdot 2^L) \cdot b_{l,j,i}^{(t)}, \tag{D.10}$$

$$||\alpha_{l,i}^{(t+1)}| - |\alpha_{l,i}^{(t)}|| \leq (3^{l+1} - 2^{l+1}) \cdot \exp(-c_6' \cdot 2^L) \leq \exp(-c_6 \cdot 2^L),\ \forall i \in [N], \tag{D.11}$$

$$|\beta_{l,i,k}^{(t+1)} - \beta_{l,i,k}^{(t)}| \leq (2^{l+1} + 2l) \cdot \exp(-c_7' \cdot 2^L) \leq \exp(-c_7 \cdot 2^L),\ \forall i, k \in [N],, \tag{D.12}$$

where $c_5, c_6, c_7 > 0$ are positive constants.

Lemma D.5. *Define $T^* = \text{poly}(\|\boldsymbol{\mu}\|_2, N, 2^L, d) \geq 2$, then for all $2 \leq t \leq T^*$, it holds that the population loss function* (2.3) *after iteration t satisfies*

$$L\big(\mathbf{v}^{(t)}, \{\mathbf{W}_l^{(t)}\}_{l=0}^L\big) \leq \exp(-c_8 \cdot 2^L), \tag{D.13}$$

where $c_8 > 0$ is a positive constant.

D.1 Proof of Lemma D.1

Proof of Lemma D.1. We present the training results of the first and second iteration in this proof. Since the partial derivative of loss function (2.3) w.r.t $\mathbf{v}$ in the updating rules is

$$\frac{\partial L}{\partial \mathbf{v}} = \mathbb{E}\bigg[\ell'(y_N f(\mathbf{X})\mathbf{e}_N) \cdot y_N \mathbf{X} \prod_{l=0}^{L} \big[\mathbf{I} + \text{Softmax}(\mathbf{H}_l^\top \mathbf{W}_l \mathbf{H}_l)\big] \cdot \mathbf{e}_N\bigg], \tag{D.14}$$

then in the first iteration, by (2.4), we have

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(0)}, \{\mathbf{W}_l^{(0)}\}_{l=0}^L\big) =& -\frac{1}{2}\mathbb{E}\bigg[y_N \mathbf{X} \prod_{l=0}^{L} \big(\mathbf{I} + \frac{1}{N}\mathbf{J}_N\big) \cdot \mathbf{e}_N\bigg] \\
=& -\frac{1}{2}\mathbb{E}\bigg\{y_N \mathbf{X} \cdot \big[\mathbf{I} + \frac{1}{N}(2^{L+1} - 1)\mathbf{J}_N\big] \cdot \mathbf{e}_N\bigg\} \\
=& -\frac{1}{2}\mathbb{E}[y_N \mathbf{x}_N] - \frac{1}{2}\mathbb{E}\bigg[y_N \sum_{j=1}^{N} \frac{1}{N}(2^{L+1} - 1)\mathbf{x}_j\bigg] \\
=& -\frac{1}{2}\mathbb{E}[y_N \mathbf{x}_N] - \frac{(2^{L+1} - 1)}{2N}\mathbb{E}[y_N \mathbf{x}_N] \\
& - \frac{(2^{L+1} - 1)}{2N}\mathbb{E}\bigg[\sum_{j \neq N} y_N \mathbf{x}_j\bigg] \\
=& -\frac{2^{L+1} - 1 + N}{2N}\boldsymbol{\mu},
\end{aligned} \tag{D.15}$$

where $\mathbf{J}_N$ is all-ones matrix of dimension N. Thus, by the updating rules of $\mathbf{v}$ (2.4),

$$\mathbf{v}^{(1)} = \mathbf{v}^{(0)} - \eta \frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(0)}, \{\mathbf{W}_l^{(0)}\}_{l=0}^L\big) = \frac{2^{L+1}-1+N}{2N}\eta\boldsymbol{\mu}.$$

In the first iteration, all $\{\mathbf{W}_l^{(t)}\}_{l=0}^L$ remain unchanged because by (2.5), $\frac{\partial L}{\partial \mathbf{W}_l}\big(\mathbf{v}^{(0)}, \{\mathbf{W}_l^{(0)}\}_{l=0}^L\big) = \mathbf{0}$ with the initialization $\mathbf{v}^{(0)} = \mathbf{0}_d,\ \mathbf{W}_l^{(0)} = \mathbf{0}_{d\times d}, l = 0, \ldots, L$.

In the second iteration, based on the results of first iteration, namely $\mathbf{v}^{(1)} = \frac{(2^{L+1}-1+N)}{2N}\eta\boldsymbol{\mu}$ and $\mathbf{W}_l^{(1)} = \mathbf{0}_{d\times d}, l = 0, \ldots, L$, the training in the second iteration is given as follows. By the partial derivative in the updating rules of $\mathbf{v}$ (D.14), the training of $\mathbf{v}$ is given as

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(1)}, \{\mathbf{W}_l^{(1)}\}_{l=0}^L\big) &= \mathbb{E}\bigg[\ell'(y_N f(\mathbf{X})\mathbf{e}_N)\cdot y_N \mathbf{X}\prod_{l=0}^{L}\big(\mathbf{I} + \frac{1}{N}\mathbf{J}_N\big)\cdot \mathbf{e}_N\bigg], \\
&\overset{(i)}{=} -r_1 \mathbb{E}\bigg[y_N \mathbf{X}\prod_{l=0}^{L}\big(\mathbf{I} + \frac{1}{N}\mathbf{J}_N\big)\cdot \mathbf{e}_N\bigg] \\
&\overset{(ii)}{=} -\frac{2^{L+1}-1+N}{N} r_1 \boldsymbol{\mu},
\end{aligned}$$

where (i) is by denote $r_1 = 1/[\exp(\frac{2^{L+1}-1+N}{2N}\eta\|\boldsymbol{\mu}\|_2^2) + 1]$, and (ii) is by similar analysis as in (D.15). Therefore, we have that

$$\mathbf{v}^{(2)} = \mathbf{v}^{(1)} - \eta \frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(1)}, \{\mathbf{W}_l^{(1)}\}_{l=0}^L\big) = (\frac{1}{2} + r_1)\cdot \frac{2^{L+1}-1+N}{N}\eta\boldsymbol{\mu}.$$

For the training of $\{\mathbf{W}_l\}_{l=0}^L$, denote $\widetilde{\mathbf{x}}^{(l)} = \frac{1}{N}(2^l - 1)\sum_{j''=1}^N \mathbf{x}_{j''},\ l = 1, \ldots, L$. After the training of first iteration, for $\mathbf{H}_l^{(1)},\ l = 1, \ldots, L$, we have that

$$\begin{aligned}
\mathbf{H}_l^{(1)}|_{\mathbf{W}_k^{(1)}=\mathbf{0}, k=0,\ldots,L} &= \mathbf{X}\prod_{k=0}^{l-1}\big[\mathbf{I} + \text{Softmax}({\mathbf{H}_k^{(1)}}^\top \mathbf{W}_k^{(1)}\mathbf{H}_k^{(1)})\big]|_{\mathbf{W}_k^{(1)}=\mathbf{0}, k=0,\ldots,L} \\
&= \mathbf{X}\big(\mathbf{I} + \frac{1}{N}\mathbf{J}_N\big)^l \\
&= \mathbf{X}[\mathbf{I} + \frac{1}{N}(2^\ell - 1)\mathbf{J}_N] \\
&= \bigg[\mathbf{x}_1 + \frac{1}{N}(2^l - 1)\sum_{j=1}^N \mathbf{x}_j, \ldots, \mathbf{x}_N + \frac{1}{N}(2^l - 1)\sum_{j=1}^N \mathbf{x}_j\bigg] \\
&= \bigg[\mathbf{x}_1 + \widetilde{\mathbf{x}}^{(l)}, \ldots, \mathbf{x}_N + \widetilde{\mathbf{x}}^{(l)}\bigg], \qquad \text{(D.16)}
\end{aligned}$$

where the first equality is by definition of $\mathbf{H}_l^{(1)}$ in (2.1), the second equality is by $\mathbf{W}_k^{(1)} = \mathbf{0}, k = 0, \ldots, L$ then $\text{Softmax}({\mathbf{H}_k^{(1)}}^\top \mathbf{W}_k^{(1)}\mathbf{H}_k^{(1)}) = \frac{1}{N}\mathbf{J}_N$, and the third equality is by the binomial expansion and the property of all-ones matrix $\mathbf{J}_N^q = N^{q-1}\mathbf{J}_N, q = 1, 2, ...$ Therefore, by (D.16) and the definition of $\widetilde{\mathbf{x}}^{(l)}$, we have that $\mathbf{h}_{l,i}^{(1)} = \mathbf{x}_i + \widetilde{\mathbf{x}}^{(l)},\ i \in [N]$ after the first iteration of the training.

For general $\mathbf{W}_{L-l_0},\ l_0 = 1, \ldots, L$, the updating are given as

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{W}_{L-l_0}}\big(\mathbf{v}^{(1)}, \{\mathbf{W}_l^{(1)}\}_{l=0}^{L}\big) \\
&= \frac{1}{N^2}\mathbb{E}\left[\ell'(y_N f(\mathbf{X})\mathbf{e}_N)\cdot y_N \sum_{k=1}^{N}\frac{2^{l_0}-1}{N}\sum_{j=1}^{N}\mathbf{v}^\top\mathbf{x}_j\sum_{j'\neq j}(\mathbf{h}_{L-l_0,j}^{(1)}{\mathbf{h}_{L-l_0,k}^{(1)}}^\top - \mathbf{h}_{L-l_0,j'}^{(1)}{\mathbf{h}_{L-l_0,k}^{(1)}}^\top)\right] \\
&\quad + \frac{1}{N^2}\mathbb{E}\left[\ell'(y_N f(\mathbf{X})\mathbf{e}_N)\cdot y_N \sum_{j=1}^{N}\mathbf{v}^\top\mathbf{x}_j\sum_{j'\neq j}(\mathbf{h}_{L-l_0,j}^{(1)}{\mathbf{h}_{L-l_0,N}^{(1)}}^\top - \mathbf{h}_{L-l_0,j'}^{(1)}{\mathbf{h}_{L-l_0,N}^{(1)}}^\top)\right] \\
&= \frac{1}{N^2}\mathbb{E}\left[\ell'(y_N f(\mathbf{X})\mathbf{e}_N)\cdot y_N \sum_{k\neq N}\frac{2^{l_0}-1}{N}\sum_{j=1}^{N}\mathbf{v}^\top\mathbf{x}_j\sum_{j'\neq j}(\mathbf{h}_{L-l_0,j}^{(1)}{\mathbf{h}_{L-l_0,k}^{(1)}}^\top - \mathbf{h}_{L-l_0,j'}^{(1)}{\mathbf{h}_{L-l_0,k}^{(1)}}^\top)\right] \\
&\quad + \frac{1}{N^2}\mathbb{E}\left[\ell'(y_N f(\mathbf{X})\mathbf{e}_N)\cdot y_N \frac{2^{l_0}-1+N}{N}\sum_{j=1}^{N}\mathbf{v}^\top\mathbf{x}_j\sum_{j'\neq j}(\mathbf{h}_{L-l_0,j}^{(1)}{\mathbf{h}_{L-l_0,N}^{(1)}}^\top - \mathbf{h}_{L-l_0,j'}^{(1)}{\mathbf{h}_{L-l_0,N}^{(1)}}^\top)\right] \\
&\overset{(i)}{=} -\frac{r_1}{N^2}\frac{2^{l_0}-1}{N}\mathbb{E}\left[y_N\sum_{k\neq N}\sum_{j=1}^{N}\mathbf{v}^\top\mathbf{x}_j\sum_{j'\neq j}(\mathbf{x}_j-\mathbf{x}_{j'})(\mathbf{x}_k+\widetilde{\mathbf{x}}^{(L-l_0)})^\top\right] \\
&\quad -\frac{r_1}{N^2}\frac{2^{l_0}-1+N}{N}\mathbb{E}\left[y_N\sum_{j=1}^{N}\mathbf{v}^\top\mathbf{x}_j\sum_{j'\neq j}(\mathbf{x}_j-\mathbf{x}_{j'})(\mathbf{x}_N+\widetilde{\mathbf{x}}^{(L-l_0)})^\top\right] \\
&= -\frac{N-1}{N^2}r_1 q_{l_0}\left\{(-2\mathbf{v}^\top\boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top - \mathbf{v}^\top\boldsymbol{\mu}\sigma_{\text{noise}}^2\mathbf{I}_d - \sigma_{\text{noise}}^2\boldsymbol{\mu}\mathbf{v}^\top) + q_{L-l_0}(N-1)\big[(N-2)\mathbf{v}^\top\boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top + N\sigma_{\text{noise}}^2\mathbf{v}\boldsymbol{\mu}^\top\big]\right\} \\
&\quad -\frac{N-1}{N^2}r_1(q_{l_0}+1)\left\{\big(N\mathbf{v}^\top\boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top + N\sigma_{\text{noise}}^2\mathbf{v}\boldsymbol{\mu}^\top + \sigma_{\text{noise}}^2\boldsymbol{\mu}\mathbf{v}^\top + \mathbf{v}^\top\boldsymbol{\mu}\sigma_{\text{noise}}^2\mathbf{I}_d\big)\right. \\
&\qquad\qquad \left. + q_{L-l_0}\big[(N-2)\mathbf{v}^\top\boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top + N\sigma_{\text{noise}}^2\mathbf{v}\boldsymbol{\mu}^\top\big]\right\} \\
&= -\frac{N-1}{N^2}r_1\left[\frac{N^2+(2^L-1)(N-2)}{N}\mathbf{v}^\top\boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top + (2^L+N-1)\sigma_{\text{noise}}^2\mathbf{v}\boldsymbol{\mu}^\top + \sigma_{\text{noise}}^2\boldsymbol{\mu}\mathbf{v}^\top + \mathbf{v}^\top\boldsymbol{\mu}\sigma_{\text{noise}}^2\mathbf{I}_d\right] \\
&\overset{(ii)}{=} -\frac{N-1}{2N^3}r_1(2^{L+1}-1+N)\eta\left\{\left[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\text{noise}}^2\right]\boldsymbol{\mu}\boldsymbol{\mu}^\top + \|\boldsymbol{\mu}\|_2^2\sigma_{\text{noise}}^2\mathbf{I}_d\right\},
\end{aligned}
$$

where we denote $q_k = (2^k-1)/N, k = 1, \ldots, L$, (i) is by denote $r_1 = 1/[\exp(\frac{2^{L+1}-1+N}{2N}\eta\|\boldsymbol{\mu}\|_2^2)+1]$, and (ii) plugs in $\mathbf{v}^{(1)} = (2^{L+1}-1+N)\eta\boldsymbol{\mu}/(2N)$. Thus, for $l_0 = 1, \ldots, L$,

$$
\begin{aligned}
\mathbf{W}_{L-l_0}^{(2)} &= \mathbf{W}_{L-l_0}^{(1)} - \eta\frac{\partial L}{\partial \mathbf{W}_{L-l_0}}\big(\mathbf{v}^{(1)}, \{\mathbf{W}_l^{(1)}\}_{l=0}^{L}\big) \\
&= r_1\frac{N-1}{2N^3}(2^{L+1}-1+N)\eta^2\left\{\left[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\text{noise}}^2\right]\boldsymbol{\mu}\boldsymbol{\mu}^\top + \|\boldsymbol{\mu}\|_2^2\sigma_{\text{noise}}^2\mathbf{I}_d\right\}.
\end{aligned}
$$

The training of $\mathbf{W}_L$: The above general updating rules for $\mathbf{W}_{L-l_0},\ l_0 = 1, \ldots, L$ do not include

$\mathbf{W}_L$. The training of $\mathbf{W}_L$ is given as follows.

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{W}_L}\big(\mathbf{v}^{(1)}, \{\mathbf{W}_l^{(1)}\}_{l=0}^L\big) \\
&= \mathbb{E}\bigg\{\ell'(y_N f(\mathbf{X})\mathbf{e}_N) y_N \sum_{j=1}^N {\mathbf{v}^{(1)}}^\top \mathbf{h}_{L,j}^{(1)} \sum_{j'=1}^N \frac{\exp({\mathbf{h}_{L,j'}^{(1)}}^\top \mathbf{W}_L^{(1)} \mathbf{h}_{L,N}^{(1)}) \exp({\mathbf{h}_{L,j}^{(1)}}^\top \mathbf{W}_L^{(1)} \mathbf{h}_{L,N}^{(1)})}{[\sum_{j''=1}^N \exp({\mathbf{h}_{L,j''}^{(1)}}^\top \mathbf{W}_L^{(1)} \mathbf{h}_{L,N}^{(1)})]^2} (\mathbf{h}_{L,j}^{(1)} {\mathbf{h}_{L,N}^{(1)}}^\top - \mathbf{h}_{L,j'}^{(1)} {\mathbf{h}_{L,N}^{(1)}}^\top)\bigg\} \\
&\overset{(i)}{=} -\frac{r_1}{N^2}\mathbb{E}\bigg[y_N \sum_{j=1}^N {\mathbf{v}^{(1)}}^\top \mathbf{h}_{L,j}^{(1)} \sum_{j'\neq j} (\mathbf{h}_{L,j}^{(1)} {\mathbf{h}_{L,N}^{(1)}}^\top - \mathbf{h}_{L,j'}^{(1)} {\mathbf{h}_{L,N}^{(1)}}^\top)\bigg] \\
&= -\frac{r_1}{N^2}\mathbb{E}\bigg[y_N \sum_{j=1}^N {\mathbf{v}^{(1)}}^\top (\mathbf{x}_j + \widetilde{\mathbf{x}}^{(L)}) \sum_{j'\neq j} (\mathbf{x}_j - \mathbf{x}_{j'})(\mathbf{x}_N + \widetilde{\mathbf{x}}^{(L)})^\top\bigg] \\
&= -\frac{N-1}{N^2} r_1 \big(N\mathbf{v}^\top \boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top + N\sigma_{\text{noise}}^2 \mathbf{v}\boldsymbol{\mu}^\top + \sigma_{\text{noise}}^2 \boldsymbol{\mu}\mathbf{v}^\top + \mathbf{v}^\top \boldsymbol{\mu}\sigma_{\text{noise}}^2 \mathbf{I}_d\big) \\
&\quad -\frac{N-1}{2N^2} q_L \big[(N-2)\mathbf{v}^\top \boldsymbol{\mu}\boldsymbol{\mu}\boldsymbol{\mu}^\top + N\sigma_{\text{noise}}^2 \mathbf{v}\boldsymbol{\mu}^\top\big] \\
&\overset{(ii)}{=} -\frac{N-1}{2N^3} r_1 (2^{L+1} - 1 + N)\eta \bigg\{\bigg[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\text{noise}}^2\bigg]\boldsymbol{\mu}\boldsymbol{\mu}^\top + \|\boldsymbol{\mu}\|_2^2 \sigma_{\text{noise}}^2 \mathbf{I}_d\bigg\},
\end{aligned}
$$

where (i) is by denote $r_1 = 1/[\exp(\frac{2^{L+1}-1+N}{2N}\eta\|\boldsymbol{\mu}\|_2^2) + 1]$, denote $q_L = \frac{1}{N}(2^L - 1)$, and (ii) is by $\mathbf{v}^{(1)} = (2^{L+1} - 1 + N)\eta\boldsymbol{\mu}/(2N)$. Thus, we have

$$
\begin{aligned}
\mathbf{W}_L^{(2)} &= \mathbf{W}_L^{(1)} - \eta \frac{\partial L}{\partial \mathbf{W}_L}\big(\mathbf{v}^{(1)}, \{\mathbf{W}_l^{(1)}\}_{l=0}^L\big) \\
&= \frac{N-1}{2N^3} r_1 (2^{L+1} - 1 + N)\eta^2 \bigg\{\bigg[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\text{noise}}^2\bigg]\boldsymbol{\mu}\boldsymbol{\mu}^\top + \|\boldsymbol{\mu}\|_2^2 \sigma_{\text{noise}}^2 \mathbf{I}_d\bigg\}.
\end{aligned}
$$

In summary, the trained $\{\mathbf{W}_l\}_{l=0}^L$ and $\mathbf{v}$ after the second iteration are as follows,

$$
\mathbf{v}^{(2)} = (\frac{1}{2} + r_1)\frac{2^{L+1} - 1 + N}{N}\eta\boldsymbol{\mu},
$$

and for $l = 0, \ldots, L$,

$$
\mathbf{W}_l^{(2)} = r_1 \frac{N-1}{2N^3}(2^{L+1} - 1 + N)\eta^2 \bigg\{\bigg[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\text{noise}}^2\bigg]\boldsymbol{\mu}\boldsymbol{\mu}^\top + \|\boldsymbol{\mu}\|_2^2 \sigma_{\text{noise}}^2 \mathbf{I}_d\bigg\},
$$

where $r_1 = 1/[\exp(\frac{2^{L+1}-1+N}{2N}\eta\|\boldsymbol{\mu}\|_2^2) + 1]$. Denote $\mathbf{v}^* = \mathbf{v}^{(2)}$ and $\mathbf{W}_l^* = \mathbf{W}_l^{(2)},\ l = 0, \ldots, L$, this completes the proof of Lemma D.1. □

D.2 Proofs of Lemmas D.2-D.5

Before proving Lemmas D.2-D.5, we first introduce the following Lemma D.6, which characterizes the approximation error of the parameters ($a_{l,j,i}^{(2)},\ b_{l,j,i}^{(2)},\ \alpha_{l,i}^{(2)},\ \beta_{l,i,k}^{(2)}$) at the second iteration when considering the approximation (E.8). The proof of Lemma D.6 is given in Appendix E.3.

Lemma D.6 (The error approximation of the parameters)**.** *Under Condition 3.1, with probability at least $1 - \exp(-c_2 \cdot 2^L)$, we have that the approximation error of the parameters $a_{l,j,i}^{(2)},\ b_{l,j,i}^{(2)},\ \alpha_{l,i}^{(2)},\ \beta_{l,i,k}^{(2)}$ at layer l satisfy*

$$
\begin{aligned}
|a_{l,j,i}^{(2)} - \widetilde{a}_l^{(2)}| \le f(l+1)\cdot \widetilde{a}_l^{(2)}, \qquad |b_{l,j,i}^{(2)} - \widetilde{b}_l^{(2)}| \le f(l+1)\cdot \widetilde{b}_l^{(2)}, \\
|\beta_{l,i,k}^{(2)} - \widetilde{\beta}_{l,i,k}^{(2)}| \le g(l)\cdot \widetilde{\beta}_{l,i,k}^{(2)}, \qquad \big||\alpha_{l,i}^{(2)}| - \widetilde{\alpha}_l^{(2)}\big| \le h(l)\cdot \widetilde{\alpha}_l^{(2)},
\end{aligned}
$$

where $\widetilde{a}_l^{(2)}$, $\widetilde{b}_l^{(2)}$, $\widetilde{\alpha}_l^{(2)}$, $\widetilde{\beta}_{l,i,k}^{(2)}$ *is the approximation of version of these parameters when only the dominant part of* $\mathbf{W}_l^{(2)}$ *are considered, and*

$$f(l+1) = \frac{\Theta(3^{2l}\cdot\sigma_{\text{noise}}^2)}{d^{\frac{3}{8}}\cdot \text{SNR}_0^2}, \quad g(l) = \frac{\Theta(3^{2l}\cdot\sigma_{\text{noise}}^2)}{d^{\frac{3}{8}}\cdot \text{SNR}_0^2}, \quad h(l) = \frac{\Theta(3^{2l}\cdot\sigma_{\text{noise}}^2)}{d^{\frac{3}{8}}\cdot \text{SNR}_0^2}.$$

In the following part, we prove that Lemmas D.2-D.5 hold for all $2 \le t \le T^*$ by induction. We first show (D.7)-(D.13) hold at $t = 2$. Then assume they hold for any iteration $t \le t_0$, and to prove that (D.7)-(D.13) hold for iteration $t_0 + 1$ ($t_0 \le T^*$).

Proof of Lemma D.2. Denote $\mathcal{A}$ the event when the conclusions in Lemma D.6 hold, by the proof of Lemma D.6, we have that $\mathbb{P}(\mathcal{A}) \ge 1 - \exp(-c_2\cdot 2^L)$.

Since $\mathbf{v}_{\text{err}}^{(2)} = \mathbf{0}$, then $\|\mathbf{v}_{\text{err}}^{(2)}\|_2 = 0$ and (D.7) naturally holds for $t = 2$. If we assume (D.7)-(D.13) hold for all $t \le t_0$, we prove (D.7) holds for $t_0 + 1$: For iteration $t_0 + 1$, the gradient descent update gives that

$$\mathbf{v}^{(t_0+1)} = \mathbf{v}^{(t_0)} - \eta\frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(t_0)}, \{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big),$$

where we have

$$\begin{aligned}
&\Big\|-\eta\frac{\partial L}{\partial \mathbf{v}}\big(\mathbf{v}^{(t_0)}, \{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big)\Big\|_2 \\
&= \eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot y_N \mathbf{h}_{L+1,N}^{(t_0)}]\Big\|_2 \\
&\overset{(i)}{\le} \eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\boldsymbol{\mu}]\Big\|_2 + \eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot(y_N\sum_{k=1}^N \beta_{L+1,N,k}^{(t_0)}\boldsymbol{\xi}_k)]\Big\|_2 \\
&\le \underbrace{\eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\boldsymbol{\mu}\,\mathbb{1}_{\{\mathcal{A}\}}]\Big\|_2}_{T_{1,1}} + \underbrace{\eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\boldsymbol{\mu}\,\mathbb{1}_{\{\mathcal{A}^c\}}]\Big\|_2}_{T_{1,2}} + \underbrace{\eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot(y_N\sum_{k=1}^N \beta_{L+1,N,k}^{(t_0)}\boldsymbol{\xi}_k)]\Big\|_2}_{T_2}
\end{aligned}$$

where $\ell'^{(t_0)} = \ell'\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)$, and (i) is by $y_N\mathbf{h}_{L+1,N}^{(t_0)} = y_N\alpha_{L+1,N}^{(t_0)}\boldsymbol{\mu} + y_N\sum_{k=1}^N\beta_{L+1,N,k}^{(t_0)}\boldsymbol{\xi}_k$. For term $T_{1,1}$, we have that

$$\begin{aligned}
\eta T_{1,1} &\le \eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\boldsymbol{\mu}\,\mathbb{1}_{\{\mathcal{A}\}}]\Big\|_2 \\
&\le \eta\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\|\boldsymbol{\mu}\|_2\,\mathbb{1}_{\{\mathcal{A}\}}] \\
&\overset{(i)}{\le} \frac{3}{2}c_1\|\boldsymbol{\mu}\|_2^{-1}\cdot 2^{L+1}\cdot\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^L\big) \\
&\le \frac{1}{2}\exp(-c\cdot 2^L),
\end{aligned}$$

where (i) is by $y_N f(\mathbf{X})\mathbf{e}_N \ge |\alpha_{L+1}^{(t_0)}|\mathbf{v}^{(2)\top}\boldsymbol{\mu} \ge (\frac{1}{2}+r_1)\frac{2^{L+1}-1+N}{N}\eta\|\boldsymbol{\mu}\|_2[1-(t_0-2)\exp(-c_7\cdot 2^L)][1-h(L+1)]\cdot\widetilde{\alpha}_{L+1}^{(2)} \ge \frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)}$ based on (D.11) at time t_0, Lemma D.6, and $\eta\|\boldsymbol{\mu}\|_2^2 = \Theta(1)$ by Condition 3.1, it follows that $|\ell'^{(t_0)}| \le \exp(-\frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)}) \le \exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)$. For term $T_{1,2}$, we have that

$$\begin{aligned}
\eta T_{1,2} &\le \eta\Big\|\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\boldsymbol{\mu}\,\mathbb{1}_{\{\mathcal{A}^c\}}]\Big\|_2 \\
&\le \eta\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\|\boldsymbol{\mu}\|_2\,\mathbb{1}_{\{\mathcal{A}^c\}}] \\
&\overset{(i)}{\le} \eta\sqrt{\mathbb{E}[|\ell'^{(t_0)}|\cdot|\alpha_{L+1,N}^{(t_0)}|\cdot\|\boldsymbol{\mu}\|_2]^2}\cdot\sqrt{\mathbb{P}(\mathcal{A}^c)} \\
&\overset{(ii)}{\le} c_1\|\boldsymbol{\mu}\|_2^{-1}\cdot 2^{L+1}\cdot\exp(-\frac{1}{2}c_2\cdot 2^L) \\
&\le \frac{1}{2}\exp(-c\cdot 2^L),
\end{aligned}$$

where (i) is by Cauchy-Schwarz inequality, (ii) is by $\mathbb{P}(\mathcal{A}) \geq 1 - \exp(-c_2 \cdot 2^L)$, $|\ell'^{(t_0)}| \leq 1$, $|\alpha_{L+1,N}^{(t_0)}| \leq 2^{L+1}$ and $\eta\|\boldsymbol{\mu}\|_2^2 = \Theta(1)$. For term T_2, we have that

$$\eta T_2 \leq \eta \bigg\| \mathbb{E}\big[\sum_{k=1}^N \beta_{L+1,N,k}^{(t_0)} \boldsymbol{\xi}_k \big] \bigg\|_2 = 0,$$

where the first inequality is by $|\ell'^{(t_0)}| \leq 1$, $|y_N| \leq 1$, and the equality is by zero mean and independent noise vector $\boldsymbol{\xi}_k,\ k = 1, \ldots, N$ from the definition. Therefore,

$$\begin{aligned}
\|\mathbf{v}_{\mathrm{err}}^{(t_0+1)}\|_2 &\leq \|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2 + \Big\| -\eta \frac{\partial L}{\partial \mathbf{v}} \big(\mathbf{v}^{(t_0)}, \{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big) \Big\|_2 \\
&\leq t_0 \exp(-c \cdot 2^L) + \eta T_{1,1} + \eta T_{1,2} + \eta T_2 \\
&\leq t_0 \exp(-c \cdot 2^L) + \exp(-c \cdot 2^L) \\
&\leq (t_0 + 1) \exp(-c \cdot 2^L).
\end{aligned}$$

By the definition of T^*, we have that

$$\|\mathbf{v}_{\mathrm{err}}^{(t_0+1)}\|_2 \leq T^* \exp(-c \cdot 2^L) \leq \exp(-c_3 \cdot 2^L),$$

which indicates (D.7) holds for $t_0 + 1$. This finished the proof of Lemma D.2. □

Proof of Lemma D.3. For all $l = 0, \ldots, L$, since $\mathbf{W}_{l,\mathrm{err}}^{(2)} = \mathbf{0}$, then $\|\mathbf{W}_{l,\mathrm{err}}^{(2)}\|_2 = 0$ and (D.8) holds for $t = 2$. If assume (D.7)-(D.13) hold for all $t \leq t_0$, we prove (D.8) holds for $t_0 + 1$: For $\mathbf{W}_{L-l_0}^{(t_0+1)}, l_0 = 0, \ldots, L-1$ at iteration $t_0 + 1$, we have that

$$\begin{aligned}
&\|\mathbf{W}_{L-l_0}^{(t_0+1)} - \mathbf{W}_{L-l_0}^{(t_0)}\|_2 \\
&= \Big\| -\eta \frac{\partial L}{\partial \mathbf{W}_{L-l_0}} \big(\mathbf{v}^{(t_0)}, \{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big) \Big\|_2 \\
&\leq \eta \bigg\| \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} (\mathbf{h}_{L-l_0,j}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top - \mathbf{h}_{L-l_0,j'}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top) \Big] \bigg\|_2 \\
&\leq \eta \bigg\| \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} (\mathbf{h}_{L-l_0,j}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top) \Big] \bigg\|_2 \\
&\quad + \eta \bigg\| \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} (\mathbf{h}_{L-l_0,j'}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top) \Big] \bigg\|_2 \\
&\leq \underbrace{\eta\, \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} \big\| \mathbf{h}_{L-l_0,j}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top \big\|_2 \mathbb{1}_{\{\mathcal{A}\}} \Big]}_{I_{1,1}} \\
&\quad + \underbrace{\eta\, \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} \big\| \mathbf{h}_{L-l_0,j}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top \big\|_2 \mathbb{1}_{\{\mathcal{A}^c\}} \Big]}_{I_{1,2}} \\
&\quad + \underbrace{\eta\, \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} \big\| \mathbf{h}_{L-l_0,j'}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top \big\|_2 \mathbb{1}_{\{\mathcal{A}\}} \Big]}_{I_{2,1}} \\
&\quad + \underbrace{\eta\, \mathbb{E}\Big[|\ell'^{(t_0)}| \cdot y_N \sum_{k=1}^N 2^{l_0} \sum_{j=1}^N {\mathbf{v}^{(t_0)}}^\top \mathbf{h}_{L-l_0,j}^{(t_0)} [\mathbf{S}_{L-l_0}^{(t_0)}]_{jk} \sum_{j' \neq j} [\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k} \big\| \mathbf{h}_{L-l_0,j'}^{(t_0)} {\mathbf{h}_{L-l_0,k}^{(t_0)}}^\top \big\|_2 \mathbb{1}_{\{\mathcal{A}^c\}} \Big]}_{I_{2,2}}.
\end{aligned}$$

For term $I_{1,1}$, we have

$$\begin{aligned}\eta I_{1,1} &\overset{(i)}{\leq} \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\sum_{k=1}^{N}2^{l_0}\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L-l_0,j}^{(t_0)}[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}(1-[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk})\big\|\mathbf{h}_{L-l_0,j}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}\}}\Big]\\ &\overset{(ii)}{\leq} 2c_1(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)N2^{3L}\cdot\max\{\text{SNR}_0^{-2},1\}\cdot\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)\\ &\overset{(iii)}{\leq} \frac{1}{4}\exp(-c\cdot 2^L),\end{aligned}$$

where (i) is by $|y_N|\leq 1$, (ii) is by $|\ell'^{(t_0)}|\leq\exp(-\frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)})\leq\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)$, $\mathbf{v}^{(t_0)\top}\mathbf{h}_{L-l_0,j}^{(t_0)}\leq|\alpha_{L-l_0,j}^{(t_0)}|(\|\mathbf{v}^{(2)}\|_2+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2)\|\boldsymbol{\mu}\|_2\leq 2(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^{L-l_0}$, $\big\|\mathbf{h}_{L-l_0,j}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\leq 2^{2(L-l_0)}\cdot\|\boldsymbol{\mu}\|_2^2\cdot\max\{\text{SNR}_0^{-2},1\}$, and $1-[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}\leq 1$, $\sum_{j=1}^{N}[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}\leq 1$. (iii) is by (D.2) hold at t_0 and Condition 3.1. Similarly, for term $I_{2,1}$, we have

$$\begin{aligned}\eta I_{2,1} &\leq \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\sum_{k=1}^{N}2^{l_0}\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L-l_0,j}^{(t_0)}[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}\sum_{j'\neq j}[\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k}\big\|\mathbf{h}_{L-l_0,j'}^{(t_0)}\big\|_2\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}\}}\Big]\\ &\leq 2c_1(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)N2^{3L}\cdot\max\{\text{SNR}_0^{-2},1\}\cdot\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)\\ &\leq \frac{1}{4}\exp(-c\cdot 2^L),\end{aligned}$$

For $I_{1,2}$, we have

$$\begin{aligned}\eta I_{1,2} &\leq \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\cdot y_N\sum_{k=1}^{N}2^{l_0}\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L-l_0,j}^{(t_0)}[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}(1-[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk})\big\|\mathbf{h}_{L-l_0,j}^{(t_0)}\big\|_2\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\ &\overset{(i)}{\leq} 2\eta(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^L\cdot\mathbb{E}\Big[\sum_{k=1}^{N}\sum_{j=1}^{N}\big\|\mathbf{h}_{L-l_0,j}^{(t_0)}\big\|_2\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\ &\overset{(ii)}{\leq} 2\eta(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^L\cdot\sqrt{\mathbb{E}\Big[\sum_{j=1}^{N}\big\|\mathbf{h}_{L-l_0,j}^{(t_0)}\big\|_2\Big]^2\cdot\mathbb{E}\Big[\sum_{k=1}^{N}\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\Big]^2}\cdot\sqrt{\mathbb{P}(\mathcal{A}^c)}\\ &\overset{(iii)}{\leq} 4c_1(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^{3L}N^2\cdot\max\{\text{SNR}_0^{-2},1\}\cdot\exp(-\frac{1}{2}c_2\cdot 2^L)\\ &\overset{(iv)}{\leq} \frac{1}{4}\exp(-c\cdot 2^L),\end{aligned}$$

where (i) is by $|\ell'^{(t_0)}|\leq 1$, $|y_N|\leq 1$, $\mathbf{v}^{(t_0)\top}\mathbf{h}_{L-l_0,j}^{(t_0)}\leq|\alpha_{L-l_0,j}^{(t_0)}|(\|\mathbf{v}^{(2)}\|_2+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2)\|\boldsymbol{\mu}\|_2\leq 2(3c_1+\|\mathbf{v}_{\text{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^{L-l_0}$ and $[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}\leq 1$, $1-[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}\leq 1$, (ii) is by Cauchy-Schwarz inequality, (iii) is by $\mathbb{P}(\mathcal{A})\geq 1-\exp(-c_2\cdot 2^L)$, and $\mathbb{E}\Big[\sum_{j=1}^{N}\big\|\mathbf{h}_{L-l_0,j}^{(t_0)}\big\|_2\Big]^2\leq 2N^2 4^L\cdot\|\boldsymbol{\mu}\|_2^2\cdot\max\{\text{SNR}_0^{-2},1\}$, $\mathbb{E}\Big[\sum_{k=1}^{N}\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\Big]^2$
$\leq 2N^2 4^L\cdot\|\boldsymbol{\mu}\|_2^2\cdot\max\{\text{SNR}_0^{-2},1\}$. (iv) is by (D.2) hold at t_0 and Condition 3.1. Similarly, for

$I_{2,2}$, we have

$$\begin{aligned}
\eta I_{2,2} &\le \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\cdot y_N\sum_{k=1}^{N}2^{l_0}\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L-l_0,j}^{(t_0)}[\mathbf{S}_{L-l_0}^{(t_0)}]_{jk}\sum_{j'\neq j}[\mathbf{S}_{L-l_0}^{(t_0)}]_{j'k}\big\|\mathbf{h}_{L-l_0,j'}^{(t_0)}\big\|_2\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\
&\le 2\eta(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^L\cdot\mathbb{E}\Big[\sum_{k=1}^{N}\sum_{j'\neq j}\big\|\mathbf{h}_{L-l_0,j'}^{(t_0)}\big\|_2\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\
&\le 2\eta(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^L\cdot\sqrt{\mathbb{E}\Big[\sum_{j'\neq j}\big\|\mathbf{h}_{L-l_0,j'}^{(t_0)}\big\|_2\Big]^2\cdot\mathbb{E}\Big[\sum_{k=1}^{N}\|\mathbf{h}_{L-l_0,k}^{(t_0)}\big\|_2\Big]^2}\cdot\sqrt{\mathbb{P}(\mathcal{A}^c)}\\
&\le 4c_1(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^{3L}N^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot\exp(-\frac{1}{2}c_2\cdot 2^L)\\
&\le \frac{1}{4}\exp(-c\cdot 2^L).
\end{aligned}$$

Therefore, for $l_0=0,\ldots,L-1$, we have that

$$\begin{aligned}
\|\mathbf{W}_{L-l_0,\mathrm{err}}^{(t_0+1)}\|_2 &\le \|\mathbf{W}_{L-l_0,\mathrm{err}}^{(t_0)}\|_2+\Big\|-\eta\frac{\partial L}{\partial\mathbf{W}_{L-l_0}}\big(\mathbf{v}^{(t_0)},\{\mathbf{W}_l^{(t_0)}\}_{l=0}^{L}\big)\Big\|_2\\
&\le \|\mathbf{W}_{L-l_0,\mathrm{err}}^{(t_0)}\|_2+\eta I_{1,1}+\eta I_{1,2}+\eta I_{2,1}+\eta I_{2,2}\\
&\le t_0\exp(-c\cdot 2^L)+\exp(-c\cdot 2^L)\\
&\le (t_0+1)\exp(-c\cdot 2^L).
\end{aligned}\tag{D.17}$$

We can proof similar results for $\mathbf{W}_L^{(t_0+1)}$ at iteration t_0+1,

$$\begin{aligned}
&\frac{\partial L}{\partial\mathbf{W}_L}\big(\mathbf{v}^{(t_0)},\{\mathbf{W}_l^{(t_0)}\}_{l=0}^{L}\big)\\
&=\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}\sum_{j'\neq j}[\mathbf{S}_L^{(t_0)}]_{j'i}(\mathbf{h}_{L,j}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top}-\mathbf{h}_{L,j'}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\Big]\\
&=\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}(1-[\mathbf{S}_L^{(t_0)}]_{ji})(\mathbf{h}_{L,j}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\Big]\\
&\quad-\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}\sum_{j'\neq j}[\mathbf{S}_L^{(t_0)}]_{j'i}(\mathbf{h}_{L,j'}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\Big]\\
&=\underbrace{\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}(1-[\mathbf{S}_L^{(t_0)}]_{ji})(\mathbf{h}_{L,j}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\,\mathbb{1}_{\{\mathcal{A}\}}\Big]}_{\widetilde{I}_{1,1}}\\
&\quad+\underbrace{\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}(1-[\mathbf{S}_L^{(t_0)}]_{ji})(\mathbf{h}_{L,j}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\,\mathbb{1}_{\{\mathcal{A}^c\}}\Big]}_{\widetilde{I}_{1,2}}\\
&\quad-\underbrace{\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}\sum_{j'\neq j}[\mathbf{S}_L^{(t_0)}]_{j'i}(\mathbf{h}_{L,j'}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\,\mathbb{1}_{\{\mathcal{A}\}}\Big]}_{\widetilde{I}_{2,1}}\\
&\quad-\underbrace{\mathbb{E}\Big[\ell'^{(t_0)}\cdot y_N\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}\sum_{j'\neq j}[\mathbf{S}_L^{(t_0)}]_{j'i}(\mathbf{h}_{L,j'}^{(t_0)}\mathbf{h}_{L,i}^{(t_0)\top})\,\mathbb{1}_{\{\mathcal{A}^c\}}\Big]}_{\widetilde{I}_{2,2}}.
\end{aligned}$$

For term $\widetilde{I}_{1,1}$, we have

$$\begin{aligned}\eta\|\widetilde{I}_{1,1}\|_2 &\overset{(i)}{\leq} \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\sum_{j=1}^{N}|\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}|[\mathbf{S}_L^{(t_0)}]_{ji}(1-[\mathbf{S}_L^{(t_0)}]_{ji})\big\|\mathbf{h}_{L,j}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}\}}\Big]\\ &\leq 2c_1(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^{3L}\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)\\ &\leq \frac{1}{4}\exp(-c\cdot 2^L),\end{aligned}$$

where (i) is by $|y_N|\leq 1$, (ii) is by $|\ell'^{(t_0)}|\leq\exp(-\frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)})\leq\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)$, $\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}\leq|\alpha_{L,j}^{(t_0)}|(\|\mathbf{v}^{(2)}\|_2+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2)\|\boldsymbol{\mu}\|_2\leq 2(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^L$, $\big\|\mathbf{h}_{L,j}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L,k}^{(t_0)}\big\|_2\leq 2^{2L}\cdot\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}$, and $1-[\mathbf{S}_L^{(t_0)}]_{jk}\leq 1$, $\sum_{j=1}^{N}[\mathbf{S}_L^{(t_0)}]_{jk}\leq 1$. (iii) is by (D.2) hold at t_0 and Condition 3.1. Similarly, for term $\widetilde{I}_{2,1}$, we have

$$\begin{aligned}\eta\|\widetilde{I}_{2,1}\|_2 &\leq \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\sum_{j=1}^{N}|\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}|[\mathbf{S}_L^{(t_0)}]_{ji}\sum_{j'\neq j}[\mathbf{S}_L^{(t_0)}]_{j'i}\big\|\mathbf{h}_{L,j'}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}\}}\Big]\\ &\leq 2c_1(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^{3L}\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot\exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)\\ &\leq \frac{1}{4}\exp(-c\cdot 2^L).\end{aligned}$$

For term $\widetilde{I}_{1,2}$, we have

$$\begin{aligned}\eta\|\widetilde{I}_{1,2}\|_2 &\leq \eta\mathbb{E}\Big[|\ell'^{(t_0)}|\cdot|y_N|\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}(1-[\mathbf{S}_L^{(t_0)}]_{ji})\big\|\mathbf{h}_{L,j}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\ &\overset{(i)}{\leq} 2\eta(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\cdot\|\boldsymbol{\mu}\|_2)2^L\mathbb{E}\Big[\sum_{j=1}^{N}\big\|\mathbf{h}_{L,j}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\ &\overset{(ii)}{\leq} 2\eta(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\cdot\|\boldsymbol{\mu}\|_2)2^L\sqrt{\mathbb{E}\Big[\sum_{j=1}^{N}\big\|\mathbf{h}_{L,j}^{(t_0)}\big\|_2\Big]^2\cdot\mathbb{E}\Big[\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\Big]^2}\cdot\sqrt{\mathbb{P}(\mathcal{A}^c)}\\ &\overset{(iii)}{\leq} 4c_1(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\cdot\|\boldsymbol{\mu}\|_2)2^{3L}N^{\frac{3}{2}}\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot\exp(-\frac{1}{2}c_2\cdot 2^L)\\ &\overset{(iv)}{\leq} \frac{1}{4}\exp(-c\cdot 2^L),\end{aligned}$$

where (i) is by $|\ell'^{(t_0)}|\leq 1$, $|y_N|\leq 1$, $\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}\leq|\alpha_{L,j}^{(t_0)}|(\|\mathbf{v}^{(2)}\|_2+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2)\|\boldsymbol{\mu}\|_2\leq 2(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\|\boldsymbol{\mu}\|_2)2^L$ and $[\mathbf{S}_L^{(t_0)}]_{ji}\leq 1$, $1-[\mathbf{S}_L^{(t_0)}]_{ji}\leq 1$, (ii) is by Cauchy-Schwarz inequality, (iii) is by $\mathbb{P}(\mathcal{A})\geq 1-\exp(-c_2\cdot 2^L)$, $\eta\|\boldsymbol{\mu}\|_2^2=\Theta(1)$, and $\mathbb{E}\Big[\sum_{j=1}^{N}\big\|\mathbf{h}_{L,j}^{(t_0)}\big\|_2\Big]^2\leq 2N^2 4^L\cdot\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}$, $\mathbb{E}\Big[\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\Big]^2\leq 2N4^L\cdot\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}$. (iv) is by (D.2) hold at t_0

and Condition 3.1. Similarly, for term $\widetilde{I}_{2,2}$, we have

$$\begin{aligned}
\eta\|\widetilde{I}_{2,2}\|_2 &\le \mathbb{E}\Big[|\ell'^{(t_0)}|\cdot|y_N|\sum_{j=1}^{N}\mathbf{v}^{(t_0)\top}\mathbf{h}_{L,j}^{(t_0)}[\mathbf{S}_L^{(t_0)}]_{ji}\sum_{j'\neq j}[\mathbf{S}_L^{(t_0)}]_{j'i}\big\|\mathbf{h}_{L,j'}^{(t_0)}\big\|_2\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\
&\le 2\eta(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\cdot\|\boldsymbol{\mu}\|_2)2^L\cdot\mathbb{E}\Big[\sum_{j'\neq j}\big\|\mathbf{h}_{L,j'}^{(t_0)}\big\|_2\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\mathbb{1}_{\{\mathcal{A}^c\}}\Big]\\
&\le 2\eta(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\cdot\|\boldsymbol{\mu}\|_2)2^L\cdot\sqrt{\mathbb{E}\Big[\sum_{j'\neq j}\big\|\mathbf{h}_{L,j'}^{(t_0)}\big\|_2\Big]^2\cdot\mathbb{E}\Big[\big\|\mathbf{h}_{L,i}^{(t_0)}\big\|_2\Big]^2}\cdot\sqrt{\mathbb{P}(\mathcal{A}^c)}\\
&\le 4c_1(3c_1+\|\mathbf{v}_{\mathrm{err}}^{(t_0)}\|_2\cdot\|\boldsymbol{\mu}\|_2)2^{3L}N^{\frac{3}{2}}\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot\exp(-\frac{1}{2}c_2\cdot 2^L)\\
&\le \frac{1}{4}\exp(-c\cdot 2^L).
\end{aligned}$$

Therefore, for $\mathbf{W}_L^{(t_0+1)}$, we have that

$$\begin{aligned}
\|\mathbf{W}_{L,\mathrm{err}}^{(t_0+1)}\|_2 &\le \|\mathbf{W}_{L,\mathrm{err}}^{(t_0)}\|_2+\Big\|-\eta\frac{\partial L}{\partial\mathbf{W}_L}\big(\mathbf{v}^{(t_0)},\{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big)\Big\|_2\\
&\le \|\mathbf{W}_{L,\mathrm{err}}^{(t_0)}\|_2+\eta\|\widetilde{I}_{1,1}\|_2+\eta\|\widetilde{I}_{1,2}\|_2+\eta\|\widetilde{I}_{2,1}\|_2+\eta\|\widetilde{I}_{2,2}\|_2\\
&\le t_0\exp(-c\cdot 2^L)+\exp(-c\cdot 2^L)\\
&\le (t_0+1)\exp(-c\cdot 2^L). \qquad \text{(D.18)}
\end{aligned}$$

Therefore, combine (D.17) and (D.18), by the definition of T^*, we have that for $l=0,\ldots,L$,

$$\|\mathbf{W}_{l,\mathrm{err}}^{(t_0+1)}\|_2 \le T^*\exp(-c\cdot 2^L)\le\exp(-c_4\cdot 2^L), \qquad \text{(D.19)}$$

which indicates (D.8) also holds for t_0+1. This finishes the proof of Lemma D.3. □

Proof of Lemma D.4. We prove that if we assume (D.9)-(D.12) hold for any iteration $2\le t<t_0$ ($t_0\le T^*$), then (D.9)-(D.12) hold for the iteration $t=t_0$: We prove these properties hold at layer $l, l=0,\ldots,L$ by induction among layers. At layer 0, we have

$$\begin{aligned}
\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i &\le \mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i+\|\mathbf{x}_j\|_2\cdot\|\mathbf{W}_{0,\mathrm{err}}^{(t_0+1)}-\mathbf{W}_{0,\mathrm{err}}^{(t_0)}\|_2\cdot\|\mathbf{x}_i\|_2\\
&\le \mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i+\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot 2\exp(-c_4\cdot 2^L),\ \forall i,j\in[N],
\end{aligned}$$

where the last inequality is by (D.19). Similarly, we also have $\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i\ge\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i-\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot 2\exp(-c_4\cdot 2^L)$. It follows that for $j\in\mathcal{I}_{y_i}$,

$$\begin{aligned}
a_{0,j,i}^{(t_0+1)} &= \frac{\exp(\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i)}{\sum_{j'=1}^N\exp(\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i)}\\
&= \frac{\exp(\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)\exp(\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i-\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)}{\sum_{j'=1}^N\exp(\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)\exp(\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i-\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)}\\
&\le a_{0,j,i}^{(t_0)}\cdot\exp(2\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot 2\exp(-c_4\cdot 2^L))\\
&\le a_{0,j,i}^{(t_0)}\cdot\exp(\exp(-c_5\cdot 2^L))\\
&\le a_{0,j,i}^{(t_0)}\cdot(1+2\exp(-c_5\cdot 2^L)),
\end{aligned}$$

where the last inequality is by $\exp(x)\le 1+2x$ for any $0\le x\le 1$, and we also have

$$\begin{aligned}
a_{0,j,i}^{(t_0+1)} &= \frac{\exp(\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)\exp(\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i-\mathbf{x}_j{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)}{\sum_{j'=1}^N\exp(\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)\exp(\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0+1)}\mathbf{x}_i-\mathbf{x}_{j'}{}^\top\mathbf{W}_0^{(t_0)}\mathbf{x}_i)}\\
&\ge a_{0,j,i}^{(t_0)}\cdot\exp(-2\|\boldsymbol{\mu}\|_2^2\cdot\max\{\mathrm{SNR}_0^{-2},1\}\cdot 2\exp(-c_4\cdot 2^L))\\
&\ge a_{0,j,i}^{(t_0)}\cdot\exp(-\exp(-c_5\cdot 2^L))\\
&\ge a_{0,j,i}^{(t_0)}\cdot(1-\exp(-c_5\cdot 2^L)),
\end{aligned}$$

where the last inequality is by $\exp(x) \geq 1 + x$ for any x. Thus, we have $|a_{0,j,i}^{(t_0+1)} - a_{0,j,i}^{(t_0)}| \leq 2\exp(-c_5 \cdot 2^L) \cdot a_{0,j,i}^{(t_0)}$ and (D.9) holds for layer 0 at iteration $t = t_0$. Similarly, we can prove that for $j \in \mathcal{I}_{\overline{y_i}}$,

$$b_{0,j,i}^{(t_0+1)} \leq b_{0,j,i}^{(t_0)} \cdot (1 + 2\exp(-c_5 \cdot 2^L))$$
$$b_{0,j,i}^{(t_0+1)} \geq b_{0,j,i}^{(t_0)} \cdot (1 - 2\exp(-c_5 \cdot 2^L)),$$

then $|b_{0,j,i}^{(t_0+1)} - b_{0,j,i}^{(t_0)}| \leq 2\exp(-c_5 \cdot 2^L) \cdot b_{0,j,i}^{(t_0)}$ and (D.10) holds for layer 0 at iteration $t = t_0$. It is easy to see that for all $i, k \in [N]$, we have $|\alpha_{0,i}^{(t_0+1)}| - |\alpha_{0,i}^{(t_0)}| = 0$, $\beta_{0,i,k}^{(t_0+1)} - \beta_{0,i,k}^{(t_0)} = 0$, then (D.11) and (D.12) hold for layer 0 at iteration $t = t_0$. Therefore, (D.9)-(D.12) hold for layer 0 at iteration $t = t_0$. If we assume (D.9)-(D.12) hold for all layer $l' \leq l$, then we prove (D.9)-(D.12) hold for layer $l + 1$: At layer $l + 1$,

For $\alpha_{l+1,i}^{(t_0+1)}$, we have

$$\begin{aligned}
|\alpha_{l+1,i}^{(t_0+1)}| - |\alpha_{l+1,i}^{(t_0)}| &= (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0+1)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0+1)})|\alpha_{l,i}^{(t_0+1)}| - (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)})|\alpha_{l,i}^{(t_0)}| \\
&= \sum_{j\in\mathcal{I}_{y_i}} (a_{l,j,i}^{(t_0+1)} - a_{l,j,i}^{(t_0)})|\alpha_{l,i}^{(t_0)}| - \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b_{l,j,i}^{(t_0+1)} - b_{l,j,i}^{(t_0)})|\alpha_{l,i}^{(t_0)}| \\
&\quad + (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0+1)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0+1)})(|\alpha_{l,i}^{(t_0+1)}| - |\alpha_{l,i}^{(t_0)}|) \\
&\overset{(i)}{\geq} -2\exp(-c_5 \cdot 2^L) \cdot (\sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}) \cdot 2^l \\
&\quad - [1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)} + 2(t_0+1)\exp(-c_5 \cdot 2^L)] \cdot (3^{l+1} - 2^{l+1})\exp(-c_6' \cdot 2^L) \\
&\geq -(3^{l+2} - 2^{l+2})\exp(-c_6' \cdot 2^L) \\
&\geq -\exp(-c_6 \cdot 2^L),
\end{aligned}$$

where (i) is by (D.9), (D.10), and (D.11) hold at layer l, Meanwhile, we also have

$$\begin{aligned}
|\alpha_{l+1,i}^{(t_0+1)}| - |\alpha_{l+1,i}^{(t_0)}| &\leq 2\exp(-c_5 \cdot 2^L) \cdot (\sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}) \cdot 2^l \\
&\quad - [1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)} + 2(t_0+1)\exp(-c_5 \cdot 2^L)] \cdot (3^{l+1} - 2^{l+1})\exp(-c_6' \cdot 2^L) \\
&\leq (3^{l+2} - 2^{l+2})\exp(-c_6' \cdot 2^L) \\
&\leq \exp(-c_6 \cdot 2^L),
\end{aligned}$$

Thus, we have

$$\big||\alpha_{l+1,i}^{(t_0+1)}| - |\alpha_{l+1,i}^{(t_0)}|\big| \leq (3^{l+2} - 2^{l+2})\exp(-c_6' \cdot 2^L) \leq \exp(-c_6 \cdot 2^L), \tag{D.20}$$

and this indicates (D.11) holds for layer $l + 1$ at iteration $t = t_0$. For $\beta_{l+1,i,k}^{(t_0+1)},\ k \in \mathcal{I}_{y_i}$, we have

$$\begin{aligned}
\beta_{l+1,i,k}^{(t_0+1)} - \beta_{l+1,i,k}^{(t_0)} &= \beta_{l,i,k}^{(t_0+1)} - \beta_{l,i,k}^{(t_0)} + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0+1)}\beta_{l,j,k}^{(t_0+1)} - \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)}\beta_{l,j,k}^{(t_0)} \\
&\quad + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0+1)}\beta_{l,j,k}^{(t_0+1)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}\beta_{l,j,k}^{(t_0)} \\
&\overset{(i)}{\leq} (2^{l+1} + 2l)\exp(-c_7' \cdot 2^L) + 2\exp(-c_5 \cdot 2^L) \cdot (\sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}) \cdot 2^l \\
&\leq [2^{l+2} + 2(l+1)]\exp(-c_7' \cdot 2^L) \\
&\leq \exp(-c_7 \cdot 2^L),
\end{aligned}$$

where (i) is by (D.9), (D.10), and (D.12) hold at layer l and $a_{l,k,i}^{(t_0)} \leq 1,\ \beta_{l,j,k}^{(t_0)} \leq 2^l$. Meanwhile, we also have

$$\begin{aligned}
\beta_{l+1,i,k}^{(t_0+1)} - \beta_{l+1,i,k}^{(t_0)} &\geq -(2^{l+1}+2l)\exp(-c_7' \cdot 2^L) - 2\exp(-c_5 \cdot 2^L) \cdot \big(\sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}\big) \cdot 2^l \\
&\geq -[2^{l+2}+2(l+1)]\exp(-c_7' \cdot 2^L) \\
&\geq -\exp(-c_7 \cdot 2^L),
\end{aligned}$$

Similarly, for $\beta_{l+1,i,k}^{(t_0+1)},\ k \in \mathcal{I}_{\overline{y_i}}$, we have

$$\begin{aligned}
\beta_{l+1,i,k}^{(t_0+1)} - \beta_{l+1,i,k}^{(t_0)} &= \beta_{l,i,k}^{(t_0+1)} - \beta_{l,i,k}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0+1)}\beta_{l,j,k}^{(t_0+1)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}\beta_{l,j,k}^{(t_0)} \\
&\quad + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0+1)}\beta_{l,j,k}^{(t_0+1)} - \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)}\beta_{l,j,k}^{(t_0)} \\
&\overset{(i)}{\leq} (2^{l+1}+2l)\exp(-c_7' \cdot 2^L) + 2\exp(-c_5 \cdot 2^L) \cdot \big(\sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}\big) \cdot 2^l \\
&\leq [2^{l+2}+2(l+1)]\exp(-c_7' \cdot 2^L) \\
&\leq \exp(-c_7 \cdot 2^L),
\end{aligned}$$

Meanwhile, we also have

$$\begin{aligned}
\beta_{l+1,i,k}^{(t_0+1)} - \beta_{l+1,i,k}^{(t_0)} &\geq -(2^{l+1}+2l)\exp(-c_7' \cdot 2^L) - 2\exp(-c_5 \cdot 2^L) \cdot \big(\sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(t_0)} + \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(t_0)}\big) \cdot 2^l \\
&\geq -[2^{l+2}+2(l+1)]\exp(-c_7' \cdot 2^L) \\
&\geq -\exp(-c_7 \cdot 2^L).
\end{aligned}$$

Thus, we have

$$|\beta_{l+1,i,k}^{(t_0+1)} - \beta_{l+1,i,k}^{(t_0)}| \leq [2^{l+2}+2(l+1)]\exp(-c_7' \cdot 2^L) \leq \exp(-c_7 \cdot 2^L), \tag{D.21}$$

and this indicates (D.12) holds for layer $l+1$ at iteration $t = t_0$. For any $i, j \in [N]$, we have that

$$\begin{aligned}
{\mathbf{h}_{l+1,j}^{(t_0+1)}}^\top \mathbf{W}_{l+1}^{(t_0+1)} \mathbf{h}_{l+1,i}^{(t_0+1)} &= {\mathbf{h}_{l+1,j}^{(t_0)}}^\top \mathbf{W}_{l+1}^{(t_0)} \mathbf{h}_{l+1,i}^{(t_0)} + {\mathbf{h}_{l+1,j}^{(t_0)}}^\top \mathbf{W}_{l+1}^{(t_0)} (\mathbf{h}_{l+1,i}^{(t_0+1)} - \mathbf{h}_{l+1,i}^{(t_0)}) \\
&\quad + (\mathbf{h}_{l+1,j}^{(t_0+1)} - \mathbf{h}_{l+1,j}^{(t_0)})^\top \mathbf{W}_{l+1}^{(t_0)} \mathbf{h}_{l+1,i}^{(t_0+1)} + {\mathbf{h}_{l+1,j}^{(t_0+1)}}^\top (\mathbf{W}_{l+1}^{(t_0+1)} - \mathbf{W}_{l+1}^{(t_0)}) \mathbf{h}_{l+1,i}^{(t_0+1)} \\
&\leq {\mathbf{h}_{l+1,j}^{(t_0)}}^\top \mathbf{W}_{l+1}^{(t_0)} \mathbf{h}_{l+1,i}^{(t_0)} + \|\mathbf{h}_{l+1,j}^{(t_0+1)}\|_2 \cdot (\|\mathbf{W}_{l+1}^{(2)}\|_2 + \|\mathbf{W}_{l+1,\mathrm{err}}^{(t_0)}\|_2) \cdot \|\mathbf{h}_{l+1,i}^{(t_0+1)} - \mathbf{h}_{l+1,i}^{(t_0)}\|_2 \\
&\quad + \|\mathbf{h}_{l+1,j}^{(t_0+1)} - \mathbf{h}_{l+1,j}^{(t_0)}\|_2 \cdot \|\mathbf{W}_{l+1}^{(t_0)}\|_2 \cdot \|\mathbf{h}_{l+1,i}^{(t_0+1)}\|_2 \\
&\quad + \|\mathbf{h}_{l+1,j}^{(t_0+1)}\|_2 \cdot \|\mathbf{W}_{l+1,\mathrm{err}}^{(t_0+1)} - \mathbf{W}_{l+1,\mathrm{err}}^{(t_0)}\|_2 \cdot \|\mathbf{h}_{l+1,i}^{(t_0+1)}\|_2 \\
&\overset{(i)}{\leq} {\mathbf{h}_{l+1,j}^{(t_0)}}^\top \mathbf{W}_{l+1}^{(t_0)} \mathbf{h}_{l+1,i}^{(t_0)} + N^2 4^L \cdot \|\boldsymbol{\mu}\|_2^2 \cdot \max\{\mathrm{SNR}_0^{-2}, 1\} \cdot 2\exp(-c_4 \cdot 2^L) \\
&\quad + (2r_1 c_1^2 + \exp(-c_4 \cdot 2^L))2\sqrt{N}2^L \cdot \|\boldsymbol{\mu}\|_2^2 \cdot \max\{\mathrm{SNR}_0^{-2}, 1\} \cdot \exp(-\min\{c_6, c_7\} \cdot 2^L) \\
&\leq {\mathbf{h}_{l+1,j}^{(t_0)}}^\top \mathbf{W}_{l+1}^{(t_0)} \mathbf{h}_{l+1,i}^{(t_0)} + \frac{1}{2}\exp(-c_5 \cdot 2^L),
\end{aligned}$$

where (i) is by (D.19), (D.20), and (D.21). Meanwhile, we also have $\mathbf{h}_{l+1,j}^{(t_0+1)\top}\mathbf{W}_{l+1}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)} \geq \mathbf{h}_{l+1,j}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)} - \frac{1}{2}\exp(-c_5\cdot 2^L)$. It follows that for $a_{l+1,j,i}^{(t_0+1)}$ and $b_{l+1,j,i}^{(t_0+1)}$, we have

$$
\begin{aligned}
a_{l+1,j,i}^{(t_0+1)} &= \frac{\exp(\mathbf{h}_{l+1,j}^{(t_0+1)\top}\mathbf{W}_{l+1}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)})}{\sum_{j'=1}^N \exp(\mathbf{h}_{l+1,j'}^{(t_0+1)\top}\mathbf{W}_{l+1}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)})} \\
&= \frac{\exp(\mathbf{h}_{l+1,j}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})\exp(\mathbf{h}_{l+1,j}^{(t_0+1)\top}\mathbf{W}_{l+1,\mathrm{err}}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)} - \mathbf{h}_{l+1,j}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})}{\sum_{j'=1}^N \exp(\mathbf{h}_{l+1,j'}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})\exp(\mathbf{h}_{l+1,j'}^{(t_0+1)\top}\mathbf{W}_{l+1,\mathrm{err}}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)} - \mathbf{h}_{l+1,j'}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})} \\
&\leq a_{l+1,j,i}^{(t_0)}\cdot\exp(\exp(-c_5\cdot 2^L)) \\
&\leq a_{l+1,j,i}^{(t_0)}\cdot(1+2\exp(-c_5\cdot 2^L)),
\end{aligned}
$$

where the last inequality is by $\exp(x)\leq 1+2x$ for any $0\leq x\leq 1$. Meanwhile, we also have

$$
\begin{aligned}
a_{l+1,j,i}^{(t_0+1)} &= \frac{\exp(\mathbf{h}_{l+1,j}^{(t_0+1)\top}\mathbf{W}_{l+1}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)})}{\sum_{j'=1}^N \exp(\mathbf{h}_{l+1,j'}^{(t_0+1)\top}\mathbf{W}_{l+1}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)})} \\
&= \frac{\exp(\mathbf{h}_{l+1,j}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})\exp(\mathbf{h}_{l+1,j}^{(t_0+1)\top}\mathbf{W}_{l+1,\mathrm{err}}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)} - \mathbf{h}_{l+1,j}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})}{\sum_{j'=1}^N \exp(\mathbf{h}_{l+1,j'}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})\exp(\mathbf{h}_{l+1,j'}^{(t_0+1)\top}\mathbf{W}_{l+1,\mathrm{err}}^{(t_0+1)}\mathbf{h}_{l+1,i}^{(t_0+1)} - \mathbf{h}_{l+1,j'}^{(t_0)\top}\mathbf{W}_{l+1}^{(t_0)}\mathbf{h}_{l+1,i}^{(t_0)})} \\
&\geq a_{l+1,j,i}^{(t_0)}\cdot\exp(-\exp(-c_5\cdot 2^L)) \\
&\geq a_{l+1,j,i}^{(t_0)}\cdot(1-\exp(-c_5\cdot 2^L)),
\end{aligned}
$$

where the last inequality is by $\exp(x)\geq 1+x$ for any x. Thus, we have $|a_{l+1,j,i}^{(t_0+1)} - a_{l+1,j,i}^{(t_0)}| \leq 2\exp(-c_5\cdot 2^L)\cdot a_{l+1,j,i}^{(t_0)}$, and (D.9) holds for layer $l+1$ at iteration $t=t_0$. Similarly, we can prove that $|b_{l+1,j,i}^{(t_0+1)} - b_{l+1,j,i}^{(t_0)}| \leq 2\exp(-c_5\cdot 2^L)\cdot b_{l+1,j,i}^{(t_0)}$, and (D.10) holds for layer $l+1$ at iteration $t=t_0$. Therefore, (D.9)-(D.12) hold for layer $l+1$ at iteration $t=t_0$.

It follows that we can conclude (D.9)-(D.12) holds at iteration $t=t_0$ for all layer $l, l=0,\ldots,L$. It is easy to see that (D.9)-(D.12) holds at iteration $t=2$ for all layer $l, l=0,\ldots,L$ by a similar set of proof. This finishes the proof of Lemma D.4. □

Proof of Lemma D.5. If we assume (D.7)-(D.13) hold for all layer $l, l=0,\ldots,L$ for any iteration $2\leq t<t_0$ ($t_0\leq T^*$), we then prove (D.13) holds for t_0: After the iteration t_0, the population loss function (2.3) gives that

$$
\begin{aligned}
L\big(\mathbf{v}^{(t_0)},\{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big) &= \mathbb{E}\big[\ell\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)\big] \\
&= \underbrace{\mathbb{E}\big[\ell\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)\mathbb{1}_{\{\mathcal{A}\}}\big]}_{\widetilde{T}_1} + \underbrace{\mathbb{E}\big[\ell\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)\mathbb{1}_{\{\mathcal{A}^c\}}\big]}_{\widetilde{T}_2},
\end{aligned}
$$

where $\ell(z)=\log(1+\exp(-z))$. For term $\widetilde{T}_1$, we have that

$$
\begin{aligned}
\widetilde{T}_1 &= \mathbb{E}\big[\ell\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)\mathbb{1}_{\{\mathcal{A}\}}\big] \\
&\overset{(i)}{\leq} \mathbb{E}\big[\exp(-y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N)\mathbb{1}_{\{\mathcal{A}\}}\big] \\
&\overset{(ii)}{\leq} \exp(-\frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)}) \\
&\overset{(iii)}{\leq} \exp(-\frac{1}{4}c_1(\frac{2}{1+p})^L) \\
&\leq \frac{1}{2}\exp(-c_8\cdot 2^L),
\end{aligned}
$$

where (i) is by $\log(1+z) \leq z$ for $z > 0$, and (ii) is by $y_N f(\mathbf{X})\mathbf{e}_N \geq |\alpha_{L+1}^{(t_0)}|\mathbf{v}^{(2)\top}\boldsymbol{\mu} \geq (\frac{1}{2} + r_1)\frac{2^{L+1}-1+N}{N}\eta\|\boldsymbol{\mu}\|_2[1-(t_0-2)\exp(-c_7\cdot 2^L)][1-h(L+1)]\cdot\widetilde{\alpha}_{L+1}^{(2)} \geq \frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)}$ based on (D.11) at time t_0-1, Lemma D.6, and $\eta\|\boldsymbol{\mu}\|_2^2 = \Theta(1)$ by Condition 3.1, (iii) is by $\exp(-\frac{1}{4}c_1\widetilde{\alpha}_{L+1}^{(2)}) \leq \exp\big(-\frac{1}{4}c_1(\frac{2}{1+p})^{L+1}\big)$. For term $\widetilde{T}_2$, we have that

$$
\begin{aligned}
\widetilde{T}_2 &= \mathbb{E}\big[\ell\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)\,\mathbb{1}_{\{\mathcal{A}^c\}}\big] \\
&\overset{(i)}{\leq} \sqrt{\mathbb{P}(\mathcal{A}^c)}\cdot\sqrt{\mathbb{E}[\ell\big(y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N\big)]^2} \\
&\overset{(ii)}{\leq} \sqrt{\mathbb{P}(\mathcal{A}^c)}\cdot\sqrt{\mathbb{E}[\exp(-2y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N)]} \\
&\overset{(iii)}{\leq} \exp(-\frac{1}{2}c_2\cdot 2^L) \\
&\leq \frac{1}{2}\exp(-c_8\cdot 2^L),
\end{aligned}
$$

where (i) is by Cauchy-Schwarz inequality, (ii) is by $\log(1+z) \leq z$ for $z > 0$, and (iii) is by $\mathbb{P}(\mathcal{A}) \geq 1-\exp(-c_2\cdot 2^L)$ and $\exp(-2y_N\cdot f(\mathbf{X})\cdot\mathbf{e}_N) \leq 1$. Therefore,

$$
\begin{aligned}
L\big(\mathbf{v}^{(t_0)}, \{\mathbf{W}_l^{(t_0)}\}_{l=0}^L\big) &\leq \widetilde{T}_1 + \widetilde{T}_2 \\
&\leq \frac{1}{2}\exp(-c_8\cdot 2^L) + \frac{1}{2}\exp(-c_8\cdot 2^L) \\
&\leq \exp(-c_8\cdot 2^L),
\end{aligned}
$$

which indicates (D.13) holds for t_0. It is easy to see that (D.13) holds at iteration $t=2$ by a similar set of proof. This finishes the proof of Lemma D.5. □

E Proof of the second part of Theorem 3.2: analysis of SNR across layers

In this section, we prove the second point in Theorem 3.2, regarding the change of SNR of each layer's outputs in transformer (2.1) with the trained $\mathbf{v}^*,\ \mathbf{W}_l^*,\ l=0,\ldots,L$. By the training results in Lemma D.1, for any layer $l=0,\ldots,L$, we have that the trained $\mathbf{W}_l^*$ (D.6) are given as

$$
\mathbf{W}_l^* = m_1\boldsymbol{\mu}\boldsymbol{\mu}^\top + m_2\mathbf{I}_d, \tag{E.1}
$$

where

$$
\begin{aligned}
m_1 &= r_1\frac{N-1}{2N^3}(2^{L+1}-1+N)\eta^2\bigg[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\text{noise}}^2\bigg], \\
m_2 &= r_1\frac{N-1}{2N^3}(2^{L+1}-1+N)\eta^2\|\boldsymbol{\mu}\|_2^2\sigma_{\text{noise}}^2,
\end{aligned} \tag{E.2}
$$

and $r_1 = 1/[\exp(\frac{2^{L+1}-1+N}{2N}\eta\|\boldsymbol{\mu}\|_2^2)+1]$.

To analyze the change of SNR across layers in deep transformer (2.1), we first introduce the decomposition of the outputs of each layer in Section E.1 and introduced the approximation of $\mathbf{W}_l^*$ in Section E.2. Then, Section E.3 proves the error bound for the approximated parameters in Lemma D.6. Finally, the second part of Theorem 3.2, which analyzes the change of SNR across the outputs of layers, is proved in Section E.4.

E.1 The change of outputs across layers

In this section, we first present the relationship between the outputs of the layers.

The changes of coefficients across layers For the output of l-th layer $\mathbf{H}_l^{(t)},\ l=0,1,\ldots,L$ of the transformer (2.1), denote i-th column of $\mathbf{H}_l^{(t)}$ as $\mathbf{h}_{l,i}^{(t)}$. By the definition of transformer (2.1), the

output of $l+1$ layer can be represented by the output of its last layer l, which is given as

$$\mathbf{h}_{l+1,i}^{(t)} = \mathbf{h}_{l,i}^{(t)} + \sum_{j=1}^{N} \frac{\exp({\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})}{\sum_{j'=1}^{N} \exp({\mathbf{h}_{l,j'}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})} \mathbf{h}_{l,j}^{(t)},\ l = 0, 1, \ldots, L-1,\ i \in [N],$$

then it follows that,

$$\langle \mathbf{h}_{l+1,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2} \rangle = \langle \mathbf{h}_{l,i}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2} \rangle + \sum_{j=1}^{N} \frac{\exp({\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})}{\sum_{j'=1}^{N} \exp({\mathbf{h}_{l,j'}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})} \langle \mathbf{h}_{l,j}^{(t)}, \frac{\boldsymbol{\mu}}{\|\boldsymbol{\mu}\|_2} \rangle,$$

$$(\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2}) \mathbf{h}_{l+1,i}^{(t)} = (\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2}) \mathbf{h}_{l,i}^{(t)} + \sum_{j=1}^{N} \frac{\exp({\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})}{\sum_{j'=1}^{N} \exp({\mathbf{h}_{l,j'}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})} (\mathbf{I} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{\|\boldsymbol{\mu}\|_2^2}) \mathbf{h}_{l,j}^{(t)}.$$

Therefore, by the decomposition (D.3), the change of $\alpha_{l,i}^{(t)}, \beta_{l,i,k}^{(t)}$ across ($l = 0, \ldots, L-1$) are given as

$$\begin{aligned}
\alpha_{l+1,i}^{(t)} &= \alpha_{l,i}^{(t)} + \sum_{j=1}^{N} \frac{\exp({\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})}{\sum_{j'=1}^{N} \exp({\mathbf{h}_{l,j'}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})} \alpha_{l,j}^{(t)},\ i \in [N], \\
\beta_{l+1,i,k}^{(t)} &= \beta_{l,i,k}^{(t)} + \sum_{j=1}^{N} \frac{\exp({\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})}{\sum_{j'=1}^{N} \exp({\mathbf{h}_{l,j'}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)})} \beta_{l,j,k}^{(t)},\ i,k \in [N].
\end{aligned} \tag{E.3}$$

The compact form of the changing rules for $\alpha_{l,i}^{(t)}$ and $\beta_{l,i,j}^{(t)},\ i,j \in [N]$ across layers are as follows, for $l = 0, 1, \ldots, L-1$,

$$[\alpha_{l+1,1}^{(t)}, \ldots, \alpha_{l+1,N}^{(t)}] = [\alpha_{l,1}^{(t)}, \ldots, \alpha_{l,N}^{(t)}] \Big[\mathbf{I} + \text{Softmax}({\mathbf{H}_l^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{H}_l^{(t)})\Big],$$

$$\begin{bmatrix} \beta_{l+1,1,1}^{(t)} & \cdots & \beta_{l+1,j,1}^{(t)} & \cdots & \beta_{l+1,N,1}^{(t)} \\ \vdots & & \vdots & & \vdots \\ \beta_{l+1,1,k}^{(t)} & \cdots & \beta_{l+1,j,k}^{(t)} & \cdots & \beta_{l+1,N,k}^{(t)} \\ \vdots & & \vdots & & \vdots \\ \beta_{l+1,1,N}^{(t)} & \cdots & \beta_{l+1,j,N}^{(t)} & \cdots & \beta_{l+1,N,N}^{(t)} \end{bmatrix} = \begin{bmatrix} \beta_{l,1,1}^{(t)} & \cdots & \beta_{l,j,1}^{(t)} & \cdots & \beta_{l,N,1}^{(t)} \\ \vdots & & \vdots & & \vdots \\ \beta_{l,1,k}^{(t)} & \cdots & \beta_{l,j,k}^{(t)} & \cdots & \beta_{l,N,k}^{(t)} \\ \vdots & & \vdots & & \vdots \\ \beta_{l,1,N}^{(t)} & \cdots & \beta_{l,j,N}^{(t)} & \cdots & \beta_{l,N,N}^{(t)} \end{bmatrix} \Big[\mathbf{I} + \text{Softmax}({\mathbf{H}_l^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{H}_l^{(t)})\Big], \tag{E.4}$$

with the input at layer $l = 0$ as

$$[\alpha_{0,1}^{(t)}, \ldots, \alpha_{0,N}^{(t)}]^\top = [y_1, \ldots, y_N]^\top$$

$$\boldsymbol{\beta}^{(0)} = \begin{bmatrix} \beta_{0,1,1}^{(t)} & \cdots & \beta_{0,j,1}^{(t)} & \cdots & \beta_{0,N,1}^{(t)} \\ \vdots & & \vdots & & \vdots \\ \beta_{0,1,k}^{(t)} & \cdots & \beta_{0,j,k}^{(t)} & \cdots & \beta_{0,N,k}^{(t)} \\ \vdots & & \vdots & & \vdots \\ \beta_{0,1,N}^{(t)} & \cdots & \beta_{0,j,N}^{(t)} & \cdots & \beta_{0,N,N}^{(t)} \end{bmatrix} = \mathbf{I}_N. \tag{E.5}$$

E.2 Approximation of $\mathbf{W}_l^*$

In this section, we consider the approximation of $\mathbf{W}_l^*$ by focusing on its dominant components. Consequently, we analyze the changing rules of the corresponding approximated parameters $\widetilde{\alpha}_{l,i}^{(t)}, \widetilde{\beta}_{l,i,j}^{(t)}$ across layers in transformer (2.1).

By the decomposition of $\mathbf{h}_{l,i}^{(t)}$ (D.2), $\forall i \in [N],\ l = 0, \ldots, L$, we have

$$\begin{aligned}
{\mathbf{h}_{l,j}^{(t)}}^\top \mathbf{W}_l^{(t)} \mathbf{h}_{l,i}^{(t)} &= [\alpha_{l,j}^{(t)} \boldsymbol{\mu} + \sum_{k=1}^{N} \beta_{l,j,k}^{(t)} \boldsymbol{\xi}_k]^\top [m_1 \boldsymbol{\mu}\boldsymbol{\mu}^\top + m_2 \mathbf{I}_d][\alpha_{l,i}^{(t)} \boldsymbol{\mu} + \sum_{k=1}^{N} \beta_{l,i,k}^{(t)} \boldsymbol{\xi}_k] \\
&= m_1 \alpha_{l,j}^{(t)} \alpha_{l,i}^{(t)} \|\boldsymbol{\mu}\|_2^4 + m_2 \alpha_{l,j}^{(t)} \alpha_{l,i}^{(t)} \|\boldsymbol{\mu}\|_2^2 + m_2 \Big\langle \sum_{k=1}^{N} \beta_{l,j,k}^{(t)} \boldsymbol{\xi}_k, \sum_{k'=1}^{N} \beta_{l,i,k'}^{(t)} \boldsymbol{\xi}_{k'} \Big\rangle
\end{aligned} \tag{E.6}$$

where m_1, m_2 are defined in (E.2).

Compare the scale of terms in (E.6), we consider the following approximation for the key terms ${\mathbf{h}_{l,j}^{(2)}}^\top \mathbf{W}_l^{(2)} \mathbf{h}_{l,i}^{(2)}$ in the Softmax matrix, for any $i,\ j \in [N]$,

$$\begin{aligned}
{\mathbf{h}_{l,j}^{(2)}}^\top \mathbf{W}_l^{(2)} \mathbf{h}_{l,i}^{(2)} &= (\alpha_{l,j}^{(2)} \boldsymbol{\mu} + \sum_{k=1}^{N} \beta_{l,j,k}^{(2)} \boldsymbol{\xi}_k)^\top (m_1 \boldsymbol{\mu}\boldsymbol{\mu}^\top + m_2 \mathbf{I}_d)(\alpha_i^{(l)} \boldsymbol{\mu} + \sum_{k=1}^{N} \beta_{i,k}^{(l)} \boldsymbol{\xi}_k) \\
&\approx \alpha_{l,i}^{(2)} \alpha_{l,j}^{(2)} m_1 \|\boldsymbol{\mu}\|_2^4 + \alpha_{l,i}^{(2)} \alpha_{l,j}^{(2)} m_2 \|\boldsymbol{\mu}\|_2^2,
\end{aligned} \tag{E.7}$$

where the coefficients are $m_1 = r_1 \frac{N-1}{2N^3}(2^{L+1} - 1 + N)\eta^2 \left[\frac{(N-2)(2^L-1)+N^2}{N} \|\boldsymbol{\mu}\|_2^2 + (2^L + N)\sigma_{\text{noise}}^2 \right]$.

Consider the approximation $\widetilde{a}_{l,j,i}^{(2)}, \widetilde{b}_{l,j,i}^{(2)}$ for $a_{l,j,i}^{(2)}, b_{l,j,i}^{(2)}$ and $\widetilde{\alpha}_{l,i}^{(2)},\ \widetilde{\beta}_{l,j,k}^{(2)}$ for $\alpha_{l,i}^{(2)},\ \beta_{l,j,k}^{(2)}$ by considering this approximation of the Softmax matrix in (E.7). By the approximation (E.7), we have $\widetilde{a}_{l,j,i}^{(2)} = \widetilde{a}_{l,j',i}^{(2)}, \forall j, j' \in \mathcal{I}_{y_i}$ and $\widetilde{b}_{l,j,i}^{(2)} = \widetilde{b}_{l,j',i}^{(2)}, \forall j, j' \in \mathcal{I}_{\overline{y_i}}$ since $|\alpha_{0,i}^{(2)}| = |\widetilde{\alpha}_{0,i}^{(2)}| = 1$ for the input layer 0. Thus, we can denote $\widetilde{a}_l^{(2)} := \widetilde{a}_{l,j,i}^{(2)}$ and $\widetilde{b}_l^{(2)} := \widetilde{b}_{l,j,i}^{(2)}$. Similarly, by (E.3) and (E.7), since $|\alpha_{0,i}^{(2)}| = |\widetilde{\alpha}_{0,i}^{(2)}| = 1$ for the input layer 0, it follows that $|\widetilde{\alpha}_{l,i}^{(2)}| = |\widetilde{\alpha}_{l,j}^{(2)}|$ for all $i, j \in [N]$. Thus, for layer l, we denote its scale as $\widetilde{\alpha}_l^{(2)} := |\widetilde{\alpha}_{l,i}^{(2)}|$, and (E.7) can be re-written as

$${\mathbf{h}_{l,j}^{(2)}}^\top \mathbf{W}_l^{(2)} \mathbf{h}_{l,i}^{(2)} \approx \alpha_{l,i}^{(2)} \alpha_{l,j}^{(2)} (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2) = \begin{cases} (\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2), \text{for } y_j = y_i \\ -(\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2), \text{for } y_j = -y_i. \end{cases} \tag{E.8}$$

It follows that when consider this approximation (E.8), by the definition of Softmax matrix, we have that the entry in the matrix $\mathbf{S}_l^{(2)}$ can be approximated as,

$$\begin{aligned}
[\mathbf{S}_l^{(2)}]_{ji} &= \frac{\exp({\mathbf{h}_{l,j}^{(2)}}^\top \mathbf{W}_l^{(2)} \mathbf{h}_{l,i}^{(2)})}{\sum_{j'=1}^{N} \exp({\mathbf{h}_{l,j'}^{(2)}}^\top \mathbf{W}_l^{(2)} \mathbf{h}_{l,i}^{(2)})} \\
&\approx \begin{cases} \dfrac{\exp\big((\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2)\big)}{|\mathcal{I}_{y_i}| \exp\big((\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2)\big) + |\mathcal{I}_{\overline{y_i}}| \exp\big(-(\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2)\big)}, \text{for } j \in \mathcal{I}_{y_i} \\ \dfrac{\exp\big(-(\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2)\big)}{|\mathcal{I}_{y_i}| \exp\big((\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2)\big) + |\mathcal{I}_{\overline{y_i}}| \exp\big(-(\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2)\big)}, \text{for } j \in \mathcal{I}_{\overline{y_i}} \end{cases} \\
&= \begin{cases} \dfrac{2}{N} \cdot \dfrac{1}{1 + \exp(-2(\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2))}, \text{for } j \in \mathcal{I}_{y_i} \\ \dfrac{2}{N} \cdot \dfrac{1}{1 + \exp(2(\widetilde{\alpha}_l^{(2)})^2 (m_1 \|\boldsymbol{\mu}\|_2^4 + m_2 \|\boldsymbol{\mu}\|_2^2))}, \text{for } j \in \mathcal{I}_{\overline{y_i}}. \end{cases}
\end{aligned}$$

Without loss of generality, we assume that $y_i = 1,\ i = 1, \ldots, N/2$ and $y_i = -1,\ i = N/2, \ldots, N$ since the order of the token does not affect the outputs of each layer in the transformer. Therefore,

for $l = 0, \ldots, L-1$, we have

$$\widetilde{\mathbf{S}}_l^{(2)} = \begin{bmatrix} \widetilde{a}_l^{(2)}\mathbf{J}_{N/2} & \widetilde{b}_l^{(2)}\mathbf{J}_{N/2} \\ \widetilde{b}_l^{(2)}\mathbf{J}_{N/2} & \widetilde{a}_l^{(2)}\mathbf{J}_{N/2} \end{bmatrix}_{N\times N},$$

where

$$\begin{aligned} \widetilde{a}_l^{(2)} &= \frac{2}{N}\cdot\frac{1}{1+\exp(-2(\widetilde{\alpha}_l^{(2)})^2(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2))}, \\ \widetilde{b}_l^{(2)} &= \frac{2}{N}\cdot\frac{1}{1+\exp(2(\widetilde{\alpha}_l^{(2)})^2(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2))}. \end{aligned} \tag{E.9}$$

Notice that here $\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)} = \frac{2}{N}$. By Lemma G.1, we can rewrite $\widetilde{\mathbf{S}}_l^{(2)}$ as

$$\widetilde{\mathbf{S}}_l^{(2)} = \frac{1}{2}(\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)}) \begin{bmatrix} \mathbf{1}_{N/2} \\ \mathbf{1}_{N/2} \end{bmatrix} [\mathbf{1}_{N/2}^\top, \mathbf{1}_{N/2}^\top] + \frac{1}{2}(\widetilde{a}_l^{(2)} - \widetilde{b}_l^{(2)}) \begin{bmatrix} \mathbf{1}_{N/2} \\ -\mathbf{1}_{N/2} \end{bmatrix} [\mathbf{1}_{N/2}^\top, -\mathbf{1}_{N/2}^\top]. \tag{E.10}$$

In the following part, we analyze changing rules of $\widetilde{\alpha}$ and $\widetilde{\boldsymbol{\beta}}$ across layers.

The changing rules for α. By the changing rules of $\alpha_{l,i}^{(t)}$ across layers (E.4), we have that for $l = 0, 1, \ldots, L-1$,

$$\begin{aligned} [\widetilde{\alpha}_{l+1,1}^{(2)}, \ldots, \widetilde{\alpha}_{l+1,N}^{(2)}] &= [\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}][\mathbf{I} + \widetilde{\mathbf{S}}_l^{(2)}] \\ &= [\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}] + \frac{1}{2}(\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)})[\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}] \begin{bmatrix} \mathbf{1}_{N/2} \\ \mathbf{1}_{N/2} \end{bmatrix} [\mathbf{1}_{N/2}^\top, \mathbf{1}_{N/2}^\top] \\ &\quad + \frac{1}{2}(\widetilde{a}_l^{(2)} - \widetilde{b}_l^{(2)})[\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}] \begin{bmatrix} \mathbf{1}_{N/2} \\ -\mathbf{1}_{N/2} \end{bmatrix} [\mathbf{1}_{N/2}^\top, -\mathbf{1}_{N/2}^\top] \\ &= ..... \\ &= [\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}] + \left[(\widetilde{a}_l^{(2)} \sum_{i\in\mathcal{I}_1} \widetilde{\alpha}_{l,i}^{(2)} + \widetilde{b}_l^{(2)} \sum_{i\in\mathcal{I}_{-1}} \widetilde{\alpha}_{l,i}^{(2)})\mathbf{1}_{N/2}^\top, (\widetilde{a}_l^{(2)} \sum_{i\in\mathcal{I}_{-1}} \widetilde{\alpha}_{l,i}^{(2)} + \widetilde{b}_l^{(2)} \sum_{i\in\mathcal{I}_1} \widetilde{\alpha}_{l,i}^{(2)})\mathbf{1}_{N/2}^\top \right] \\ &= [\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}] + \left[(\frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)})\widetilde{\alpha}_l^{(2)}\mathbf{1}_{N/2}^\top, (\frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)})(-\widetilde{\alpha}_l^{(2)})\mathbf{1}_{N/2}^\top \right] \\ &= [\widetilde{\alpha}_{l,1}^{(2)}, \ldots, \widetilde{\alpha}_{l,N}^{(2)}](1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)}) \end{aligned}$$

Therefore, we have that for $l = 0, \ldots, L-1$,

$$\widetilde{\alpha}_{l+1,i}^{(2)} = (1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)}) \cdot \widetilde{\alpha}_{l+1,i}^{(2)},\ i \in [N]. \tag{E.11}$$

Since $1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)} = 1 + N\widetilde{a}_l^{(2)} - (\frac{N}{2}\widetilde{a}_l^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}) = N\widetilde{a}_l^{(2)} \geq 1$ by definition of $\widetilde{a}_l^{(2)}$ in (E.9), then

$$\widetilde{\alpha}_{l+1}^{(2)} = |\widetilde{\alpha}_{l+1,i}^{(2)}| \geq \widetilde{\alpha}_l^{(2)} = |\widetilde{\alpha}_{l,i}^{(2)}|,\ \forall l = 0, \ldots, L-1. \tag{E.12}$$

The change of β across layers. We first claim that $\widetilde{\boldsymbol{\beta}}_l^{(2)},\ l = 0, 1, \ldots, L$ have the following format,

$$\widetilde{\boldsymbol{\beta}}_l^{(2)} = \mathbf{I}_N + \begin{bmatrix} \widetilde{\beta}_{l,+}^{(2)}\mathbf{J}_{N/2} & \widetilde{\beta}_{l,-}^{(2)}\mathbf{J}_{N/2} \\ \widetilde{\beta}_{l,-}^{(2)}\mathbf{J}_{N/2} & \widetilde{\beta}_{l,+}^{(2)}\mathbf{J}_{N/2} \end{bmatrix}_{N\times N}, \tag{E.13}$$

for some $\widetilde{\beta}_{l,+}^{(2)}, \widetilde{\beta}_{l,-}^{(2)} \in \mathbb{R}$.

We prove that $\widetilde{\boldsymbol{\beta}}_l^{(2)}$ can be written in the format of (E.13) by induction: Firstly, $\boldsymbol{\beta}_0^{(2)} = \mathbf{I}_N$ satisfies (E.13) with $\beta_{0,+}^{(2)} = \beta_{0,-}^{(2)} = 0$. For $\widetilde{\boldsymbol{\beta}}_1^{(2)}$, we have

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_1^{(2)} &= \widetilde{\boldsymbol{\beta}}_0^{(2)}\left[\mathbf{I}_N + \widetilde{\mathbf{S}}_0^{(2)}\right] \\ &= \mathbf{I}_N + \widetilde{\mathbf{S}}_0^{(2)} \\ &= \mathbf{I}_N + \begin{bmatrix} \widetilde{a}_0^{(2)}\mathbf{J}_{N/2} & \widetilde{b}_0^{(2)}\mathbf{J}_{N/2} \\ \widetilde{b}_0^{(2)}\mathbf{J}_{N/2} & \widetilde{a}_0^{(2)}\mathbf{J}_{N/2} \end{bmatrix},\end{aligned}$$

then $\widetilde{\boldsymbol{\beta}}_1^{(2)}$ satisfies (E.13) with $\widetilde{\beta}_{1,+}^{(2)} = a_0^{(2)},\ \widetilde{\beta}_{1,-}^{(2)} = b_0^{(2)}$. Assume that for all $\widetilde{\boldsymbol{\beta}}_{l'}^{(2)},\ l' \le l$ satisfy (E.13), we are going to prove (E.13) also hold for $\widetilde{\boldsymbol{\beta}}_{l+1}^{(2)}$: For $\widetilde{\boldsymbol{\beta}}_{l+1}^{(2)}$, we have that

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_{l+1}^{(2)} &= \widetilde{\boldsymbol{\beta}}_l^{(2)}\left[\mathbf{I} + \widetilde{\mathbf{S}}_l^{(2)}\right] \\ &\overset{(i)}{=} \left[\mathbf{I} + \frac{1}{2}(\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)})\begin{bmatrix}\mathbf{1}_{N/2}\\ \mathbf{1}_{N/2}\end{bmatrix}[\mathbf{1}_{N/2}^\top, \mathbf{1}_{N/2}^\top] + \frac{1}{2}(\widetilde{\beta}_{l,+}^{(2)} - \widetilde{\beta}_{l,-}^{(2)})\begin{bmatrix}\mathbf{1}_{N/2}\\ -\mathbf{1}_{N/2}\end{bmatrix}[\mathbf{1}_{N/2}^\top, -\mathbf{1}_{N/2}^\top]\right] \\ &\quad \cdot \left[\mathbf{I} + \frac{1}{2}(\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)})\begin{bmatrix}\mathbf{1}_{N/2}\\ \mathbf{1}_{N/2}\end{bmatrix}[\mathbf{1}_{N/2}^\top, \mathbf{1}_{N/2}^\top] + \frac{1}{2}(\widetilde{a}_l^{(2)} - \widetilde{b}_l^{(2)})\begin{bmatrix}\mathbf{1}_{N/2}\\ -\mathbf{1}_{N/2}\end{bmatrix}[\mathbf{1}_{N/2}^\top, -\mathbf{1}_{N/2}^\top]\right] \\ &= .... \\ &= \mathbf{I} + [\frac{1}{2}(\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)}) + \frac{1}{2}(\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)}) + \frac{N}{4}(\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)})(\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)})]\begin{bmatrix}\mathbf{1}_{N/2}\\ \mathbf{1}_{N/2}\end{bmatrix}[\mathbf{1}_{N/2}^\top, \mathbf{1}_{N/2}^\top] \\ &\quad + [\frac{1}{2}(\widetilde{a}_l^{(2)} - \widetilde{b}_l^{(2)}) + \frac{1}{2}(\widetilde{\beta}_{l,+}^{(2)} - \widetilde{\beta}_{l,-}^{(2)}) + \frac{N}{4}(\widetilde{a}_l^{(2)} - \widetilde{b}_l^{(2)})(\widetilde{\beta}_{l,+}^{(2)} - \widetilde{\beta}_{l,-}^{(2)})]\begin{bmatrix}\mathbf{1}_{N/2}\\ -\mathbf{1}_{N/2}\end{bmatrix}[\mathbf{1}_{N/2}^\top, -\mathbf{1}_{N/2}^\top] \\ &= \mathbf{I} + \begin{bmatrix} (\widetilde{a}_l^{(2)} + \widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)})\mathbf{J}_{N/2} & (\widetilde{b}^{(l)} + \widetilde{\beta}_{l,-}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)})\mathbf{J}_{N/2} \\ (\widetilde{b}_l^{(2)} + \widetilde{\beta}_{l,-}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)})\mathbf{J}_{N/2} & (\widetilde{a}^{(l)} + \widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)})\mathbf{J}_{N/2} \end{bmatrix},\end{aligned}$$

where (i) is by $\widetilde{\boldsymbol{\beta}}_l^{(2)}$ satisfy decomposition (E.13) and $\widetilde{\mathbf{S}}_l^{(2)}$ in (E.10). Therefore, $\widetilde{\boldsymbol{\beta}}_{l+1}^{(2)}$ also satisfy (E.13) with

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_{l+1}^{(2)} &= \mathbf{I}_N + \begin{bmatrix} \widetilde{\beta}_{l+1,+}^{(2)}\mathbf{J}_{N/2} & \widetilde{\beta}_{l+1,-}^{(2)}\mathbf{J}_{N/2} \\ \widetilde{\beta}_{l+1,-}^{(2)}\mathbf{J}_{N/2} & \widetilde{\beta}_{l+1,+}^{(2)}\mathbf{J}_{N/2} \end{bmatrix}, \\ \text{where } \widetilde{\beta}_{l+1,+}^{(2)} &= \widetilde{a}_l^{(2)} + \widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)}, \\ \widetilde{\beta}_{l+1,-}^{(2)} &= \widetilde{b}_l^{(2)} + \widetilde{\beta}_{l,-}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)}.\end{aligned}$$

By induction, we can conclude that all $\widetilde{\boldsymbol{\beta}}_l^{(2)},\ l = 0, \ldots, L$ satisfy (E.13), and we have $\widetilde{\boldsymbol{\beta}}_l^{(2)} = [\beta_{l,i,k}^{(2)}]_{N\times N}$ where the entry

$$\widetilde{\beta}_{l+1,i,k}^{(2)} = \begin{cases} \widetilde{\beta}_{l+1,+}^{(2)} + 1, \text{ for } k = i, \\ \widetilde{\beta}_{l+1,+}^{(2)}, \text{ for } k \in \mathcal{I}_{y_i} \setminus \{i\}, \\ \widetilde{\beta}_{l+1,-}^{(2)}, \text{ for } k \in \mathcal{I}_{\overline{y_i}}, \end{cases} \tag{E.14}$$

$$\begin{aligned}\text{with } \widetilde{\beta}_{l+1,+}^{(2)} &= \widetilde{a}_l^{(2)} + \widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)}, \\ \widetilde{\beta}_{l+1,-}^{(2)} &= \widetilde{b}_l^{(2)} + \widetilde{\beta}_{l,-}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)}.\end{aligned} \tag{E.15}$$

We can further obtain the following properties for $\widetilde{\boldsymbol{\beta}}_l^{(2)}$: $\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)} = \frac{2}{N}(2^l - 1),\ l = 0, \ldots, L$. It can be proved as follows: By (E.15), we have

$$\begin{aligned}\widetilde{\beta}_{l+1,+}^{(2)} + \widetilde{\beta}_{l+1,-}^{(2)} &= (\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)}) + (\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)}) + \frac{N}{2}(\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)})(\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)}) \\ &= \frac{2}{N} + 2(\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)}),\end{aligned}$$

where the last equality is by $\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)} = \frac{2}{N}$. Given the initialization at the input layer 0 that $\widetilde{\beta}_{0,+}^{(2)} + \widetilde{\beta}_{0,-}^{(2)} = 0$, it follows that $\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)} = \frac{2}{N}(2^l - 1),\ l = 0, \ldots, L$.

E.3 Proof of Lemma D.6

In this section, we prove the error bounds for the approximated parameters in Lemma D.6.

Proof of Lemma D.6. We prove that the following properties hold for the error bound of these parameters across layer $l,\ l = 1, \ldots, L$ by induction:

$$
\begin{aligned}
|a_{l,j,i}^{(2)} - \widetilde{a}_l^{(2)}| \le f(l+1) \cdot \widetilde{a}_l^{(2)}, \qquad |b_{l,j,i}^{(2)} - \widetilde{b}_l^{(2)}| \le f(l+1) \cdot \widetilde{b}_l^{(2)}, \\
|\beta_{l,i,k}^{(2)} - \widetilde{\beta}_{l,+}^{(2)}| \le g(l) \cdot \widetilde{\beta}_{l,+}^{(2)}, \text{ for } k \in \mathcal{I}_{y_i}, \\
|\beta_{l,i,k}^{(2)} - \widetilde{\beta}_{l,-}^{(2)}| \le g(l) \cdot \widetilde{\beta}_{l,-}^{(2)}, \text{ for } k \in \mathcal{I}_{\overline{y_i}}, \\
\big||\alpha_{l,i}^{(2)}| - \widetilde{\alpha}_l^{(2)}\big| \le h(l) \cdot \widetilde{\alpha}_l^{(2)},
\end{aligned}
$$

where

$$
f(l+1) = \frac{6 \times 3^{2l}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) = \frac{\Theta(3^{2l} \cdot \sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^2}, \tag{E.16}
$$

$$
g(l) = \frac{\frac{4}{5} \times 3^{2l}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) = \frac{\Theta(3^{2l} \cdot \sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^2}, \tag{E.17}
$$

$$
h(l) = \frac{\frac{4}{5} \times 3^{2l}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) = \frac{\Theta(3^{2l} \cdot \sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^2}. \tag{E.18}
$$

We will also prove the following upper bounds for them by induction,

$$
f(l+1) \le \frac{1}{2} \tag{E.19}
$$

$$
g(l) \le (\frac{3}{2})^l - 1 \tag{E.20}
$$

$$
h(l) \le (1 + \frac{1+p^2}{4})^l - 1. \tag{E.21}
$$

First, note that by Condition 3.1, and $(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) = \Theta(\|\boldsymbol{\mu}\|_2^2)$, we have

$$
\frac{3^{2L}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) = o(1). \tag{E.22}
$$

By Lemma G.2 and G.3, we have with probability at least $1 - \exp(-c_2 \cdot 2^L)$,

$$
\Big\langle \sum_{k=1}^N \beta_{l,j,k}^{(2)} \boldsymbol{\xi}_k, \sum_{k'=1}^N \beta_{l,i,k'}^{(2)} \boldsymbol{\xi}_{k'} \Big\rangle \le \sigma_{\mathrm{noise}}^2 d^{\frac{5}{8}} N \Big[\frac{N}{2}({\beta_{l,+}^{(2)}}^2 + \beta_{l,+}^{(2)}\beta_{l,-}^{(2)}) + \frac{1}{2}(\beta_{l,+}^{(2)} + \beta_{l,-}^{(2)}) \Big] + N{\beta_{l,+}^{(2)}}^2 \sigma_{\mathrm{noise}}^2 d,
$$

$$
\Big\langle \sum_{k=1}^N \beta_{l,j,k}^{(2)} \boldsymbol{\xi}_k, \sum_{k'=1}^N \beta_{l,i,k'}^{(2)} \boldsymbol{\xi}_{k'} \Big\rangle \ge -\sigma_{\mathrm{noise}}^2 d^{\frac{5}{8}} N \Big[\frac{N}{2}({\beta_{l,+}^{(2)}}^2 + \beta_{l,+}^{(2)}{\beta_{l,-}}^{(2)}) - \frac{1}{2}(\beta_{l,+}^{(2)} + \beta_{l,-}^{(2)}) \Big] + N{\beta_{l,+}^{(2)}}^2 \sigma_{\mathrm{noise}}^2 d,
$$

and by the definition of m_1 and m_2, we have

$$
\begin{aligned}
\frac{m_1\|\boldsymbol{\mu}\|_2^4}{m_2\sigma_{\mathrm{noise}}^2 d^{\frac{5}{8}}} &= \frac{[\frac{(N-2)(2^L-1)+N^2}{N}\|\boldsymbol{\mu}\|_2^2 + (2^L+N)\sigma_{\mathrm{noise}}^2]\|\boldsymbol{\mu}\|_2^4}{\|\boldsymbol{\mu}\|_2^2\sigma_{\mathrm{noise}}^2\sigma_{\mathrm{noise}}^2 d^{\frac{5}{8}}} \\
&\ge [N + \frac{2}{3}(2^L - 1)] d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4,
\end{aligned}
$$

thus,

$$\begin{aligned}
&\frac{{\alpha_l}^2(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)}{m_2N[N({\beta_{l,+}^{(2)}}^2+\beta_{l,+}^{(2)}\beta_{l,-}^{(2)})+(\beta_{l,+}^{(2)}+\beta_{l,-}^{(2)})]\sigma_{\text{noise}}^2\sqrt{d}}\\
&\geq \frac{N+\frac{2}{3}(2^L-1)}{N}d\cdot d^{\frac{3}{8}}\cdot \text{SNR}_0^4\\
&\geq d\cdot d^{\frac{3}{8}}\cdot \text{SNR}_0^4. \qquad \text{(E.23)}
\end{aligned}$$

For layer 1, for $j\in\mathcal{I}_{y_i}$,

$$\begin{aligned}
a_{0,j,i}^{(2)} &= \frac{\exp({\mathbf{h}_{0,j}}^\top\mathbf{W}_0\mathbf{h}_{0,i})}{\sum_{j'=1}^N\exp({\mathbf{h}_{0,j'}}^\top\mathbf{W}_0\mathbf{h}_{0,i})},\\
&\overset{(i)}{=} \frac{\exp(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_j\rangle)}{\sum_{j'\in\mathcal{I}_{y_i}}\exp(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)+\sum_{j'\in\mathcal{I}_{\overline{y_i}}}\exp(-m_1\|\boldsymbol{\mu}\|_2^4-m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)}\\
&= \frac{\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_j\rangle)}{\sum_{j'\in\mathcal{I}_{y_i}}\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)+\sum_{j'\in\mathcal{I}_{\overline{y_i}}}\exp(-2m_1\|\boldsymbol{\mu}\|_2^4-2m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)}\\
&\overset{(ii)}{\leq} \frac{2}{N}\cdot\frac{\exp\big(\frac{1}{d\cdot d^{\frac{3}{8}}\cdot\text{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4-2m_2\|\boldsymbol{\mu}\|_2^2)}\\
&\overset{(iii)}{\leq} \frac{2}{N}\cdot\frac{1+\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\text{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4-2m_2\|\boldsymbol{\mu}\|_2^2)},
\end{aligned}$$

where (i) is by (E.5) and (E.6), and (ii) is by (E.23), (iii) is by $\exp(x)\leq 1+2x$ for $0\leq x\leq 1$. At the same time, we have that

$$\begin{aligned}
a_{0,j,i}^{(2)} &= \frac{\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_j\rangle)}{\sum_{j'\in\mathcal{I}_{y_i}}\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)+\sum_{j'\in\mathcal{I}_{\overline{y_i}}}\exp(-2m_1\|\boldsymbol{\mu}\|_2^4-2m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)}\\
&\geq \frac{2}{N}\cdot\frac{\exp\big(-\frac{1}{d\cdot d^{\frac{3}{8}}\cdot\text{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4-2m_2\|\boldsymbol{\mu}\|_2^2)}\\
&\overset{(i)}{\geq} \frac{2}{N}\cdot\frac{[1-\frac{1}{d\cdot d^{\frac{3}{8}}\cdot\text{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)]}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4-2m_2\|\boldsymbol{\mu}\|_2^2)},
\end{aligned}$$

where (i) is by $\exp(x)\geq 1+x,\forall x$. Similarly, for $j\in\mathcal{I}_{\overline{y_i}}$,

$$\begin{aligned}
b_{0,j,i}^{(2)} &= \frac{\exp({\mathbf{h}_{0,j}}^\top\mathbf{W}_0\mathbf{h}_{i,0})}{\sum_{j'=1}^N\exp({\mathbf{h}_{0,j'}}^\top\mathbf{W}_0\mathbf{h}_{0,i})}\\
&\overset{(i)}{=} \frac{\exp(-m_1\|\boldsymbol{\mu}\|_2^4-m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_j\rangle)}{\sum_{j'\in\mathcal{I}_{y_i}}\exp(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)+\sum_{j'\in\mathcal{I}_{\overline{y_i}}}\exp(-m_1\|\boldsymbol{\mu}\|_2^4-m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)}\\
&= \frac{\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_j\rangle)}{\sum_{j'\in\mathcal{I}_{y_i}}\exp(2m_1\|\boldsymbol{\mu}\|_2^4+2m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)+\sum_{j'\in\mathcal{I}_{\overline{y_i}}}\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)}\\
&\overset{(ii)}{\leq} \frac{2}{N}\cdot\frac{\exp\big(\frac{1}{d\cdot d^{\frac{3}{8}}\cdot\text{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp(2m_1\|\boldsymbol{\mu}\|_2^4+2m_2\|\boldsymbol{\mu}\|_2^2)}\\
&\overset{(iii)}{\leq} \frac{2}{N}\cdot\frac{1+\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\text{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(2m_1\|\boldsymbol{\mu}\|_2^4+2m_2\|\boldsymbol{\mu}\|_2^2)},
\end{aligned}$$

where (i) is by (E.5) and (E.6), and (ii) is by (E.23), (iii) is by $\exp(x) \le 1 + 2x$ for $0 \le x \le 1$. Similarly, we have

$$b_{0,j,i}^{(2)} = \frac{\exp(-m_1\|\boldsymbol{\mu}\|_2^4 - m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_j\rangle)}{\sum_{j'\in\mathcal{I}_{y_i}}\exp(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle) + \sum_{j'\in\mathcal{I}_{\overline{y_i}}}\exp(-m_1\|\boldsymbol{\mu}\|_2^4 - m_2\|\boldsymbol{\mu}\|_2^2)\exp(m_2\langle\boldsymbol{\xi}_i,\boldsymbol{\xi}_{j'}\rangle)}$$
$$\ge \frac{2}{N}\cdot\frac{1 - \frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(2m_1\|\boldsymbol{\mu}\|_2^4 + 2m_2\|\boldsymbol{\mu}\|_2^2)}.$$

Therefore,

$$|a_{0,j,i}^{(2)} - \widetilde{a}_0^{(2)}| \le \frac{2}{N}\cdot\frac{\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4 - 2m_2\|\boldsymbol{\mu}\|_2^2)} = \frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{a}_0^{(2)}$$
$$|b_{0,j,i}^{(2)} - \widetilde{b}_0^{(2)}| \le \frac{2}{N}\cdot\frac{\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(2m_1\|\boldsymbol{\mu}\|_2^4 + 2m_2\|\boldsymbol{\mu}\|_2^2)} = \frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{b}_0^{(2)} \tag{E.24}$$

Obviously, (E.16) holds for $l = 0$. By condition in (E.22), we directly have $f(1) \le \frac{1}{2}$, and (E.19) holds at $l = 0$.

For $\alpha_{1,i}^{(2)}, \beta_{1,i,j}^{(2)}$, we have

$$\begin{aligned}
|\alpha_{1,i}^{(2)}| - \widetilde{\alpha}_1^{(2)} &\overset{(i)}{=} (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{0,j,i}^{(2)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{0,j,i}^{(2)})|\alpha_{0,i}^{(2)}| - (1 + \frac{N}{2}\widetilde{a}_0^{(2)} - \frac{N}{2}\widetilde{b}_0^{(2)})\widetilde{\alpha}_{0,i}^{(2)} \\
&\overset{(ii)}{=} \sum_{j\in\mathcal{I}_{y_i}}(a_{0,j,i}^{(2)} - \widetilde{a}_0^{(2)}) - \sum_{j\in\mathcal{I}_{\overline{y_i}}}(b_{0,j,i}^{(2)} - \widetilde{b}_0^{(2)}) \\
&\overset{(iii)}{\le} \frac{\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4 - 2m_2\|\boldsymbol{\mu}\|_2^2)} - \Big(-\frac{\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(2m_1\|\boldsymbol{\mu}\|_2^4 + 2m_2\|\boldsymbol{\mu}\|_2^2)}\Big) \\
&= \frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2),
\end{aligned}$$

where (i) is by (E.4) and (E.11), (ii) is by $|\alpha_{0,i}^{(2)}| = \widetilde{\alpha}_0^{(2)} = 1$, (iii) is by (E.24). On the other hand,

$$\begin{aligned}
|\alpha_{1,i}^{(2)}| - \widetilde{\alpha}_1^{(2)} &= \sum_{j\in\mathcal{I}_{y_i}}(a_{0,j,i}^{(2)} - \widetilde{a}_0^{(2)}) - \sum_{j\in\mathcal{I}_{\overline{y_i}}}(b_{0,j,i}^{(2)} - \widetilde{b}_0^{(2)}) \\
&\overset{(i)}{\ge} -\frac{\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(-2m_1\|\boldsymbol{\mu}\|_2^4 - 2m_2\|\boldsymbol{\mu}\|_2^2)} - \frac{\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp(2m_1\|\boldsymbol{\mu}\|_2^4 + 2m_2\|\boldsymbol{\mu}\|_2^2)} \\
&= -\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2),
\end{aligned}$$

where (i) is by the error bound of in (E.24).

For the error bound of $\beta_{1,i,k}^{(2)}$, by its changing rules across layers (E.4) (also (E.15)), since the $\beta_{0,i,k}^{(2)} = 0,\ \forall k \in [N]\setminus\{i\}$ (also $\widetilde{\beta}_{0,+}^{(2)} = \widetilde{\beta}_{0,-}^{(2)} = 0$), then we have

$$\beta_{1,i,k}^{(2)} = a_0^{(2)},\ k\in\mathcal{I}_{y_i}\setminus\{i\}, \qquad \beta_{1,i,k}^{(2)} = b_0^{(2)},\ k\in\mathcal{I}_{\overline{y_i}},$$

and $\widetilde{\beta}_{1,+}^{(2)} = \widetilde{a}_0^{(2)},\ \widetilde{\beta}_{1,-}^{(2)} = \widetilde{b}_0^{(2)}$. Thus, by (E.24),

$$|\beta_{1,i,k}^{(2)} - \widetilde{\beta}_{1,+}^{(2)}| = |a^{(0)} - \widetilde{a}^{(0)}| \le \frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\beta}_+^{(0)},\ k\in\mathcal{I}_{y_i}\setminus\{i\},$$
$$|\beta_{1,i,k}^{(2)} - \widetilde{\beta}_{1,-}^{(2)}| = |b^{(0)} - \widetilde{b}^{(0)}| \le \frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\beta}_-^{(0)},\ k\in\mathcal{I}_{\overline{y_i}}.$$

Therefore,

$$\big||\alpha_{1,i}^{(2)}|-\widetilde{\alpha}_1^{(2)}\big|\le\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\overset{(i)}{\le}\frac{\frac{4}{5}\cdot 3^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\alpha}_1^{(2)},$$

$$|\beta_{1,i,k}^{(2)}-\widetilde{\beta}_{1,i,k}^{(2)}|\le\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\beta}_{1,+}^{(2)}\le\frac{\frac{4}{5}\cdot 3^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\beta}_{1,+}^{(2)},\ k\in\mathcal{I}_{y_i},$$

$$|\beta_{1,i,k}^{(2)}-\widetilde{\beta}_{1,i,k}^{(2)}|\le\frac{2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\beta}_{1,-}^{(2)}\le\frac{\frac{4}{5}\cdot 3^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\cdot\widetilde{\beta}_{1,-}^{(2)},\ k\in\mathcal{I}_{\overline{y_i}},$$

where (i) is by $\frac{1}{N\widetilde{a}_0^{(2)}}\le 1$, then (E.17) and (E.18) naturally hold for $l=1$.

Also, by condition in (E.22), we have $g(1)\le 1/2,\ h(1)\le 1/2$, then (E.20), (E.21) hold for $l=1$. This means (E.16)-(E.18) and (E.19)-(E.21) all hold at $l=0$.

Assume that (E.16), (E.17), (E.18) hold for layer l, i.e.,

$$\begin{aligned}f(l+1)&=\frac{6\times 3^{2l}}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2),\\ g(l)&=\frac{\frac{4}{5}\times 3^{2l}}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2),\\ h(l)&=\frac{\frac{4}{5}\times 3^{2l}}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2).\end{aligned}\tag{E.25}$$

We also assume that (E.19), (E.20), (E.21) hold for layer l, i.e.,

$$f(l+1)\le\frac{1}{2},\quad g(l)\le(\frac{3}{2})^l-1,\quad h(l)\le(1+\frac{1+p^2}{4})^l-1.\tag{E.26}$$

We prove that the upper bound of the error (E.16)-(E.18) and (E.19)-(E.21) hold for layer $l+1$. For the output of layer $l+1$, the error bound of $\alpha_{l+1,i}^{(2)}$ and $\beta_{l+1,i,k}^{(2)}$ are proved as follows. For $\alpha_{l+1,i}^{(2)}$, by its changing rules across layers (E.4) and (E.11),

$$\begin{aligned}|\alpha_{l+1,i}^{(2)}|-\widetilde{\alpha}_{l+1}^{(2)}&=(1+\sum_{j\in\mathcal{I}_{y_i}}a_{l,j,i}^{(2)}-\sum_{j\in\mathcal{I}_{\overline{y_i}}}b_{l,j,i}^{(2)})|\alpha_{l,i}^{(2)}|-(1+\frac{N}{2}\widetilde{a}_l^{(2)}-\frac{N}{2}\widetilde{b}_l^{(2)})\widetilde{\alpha}_l^{(2)}\\ &\overset{(i)}{\le}(1+\sum_{j\in\mathcal{I}_{y_i}}a_{l,j,i}^{(2)}-\sum_{j\in\mathcal{I}_{\overline{y_i}}}b_{l,j,i}^{(2)})(\widetilde{\alpha}_l^{(2)}+h(l)\widetilde{\alpha}_l^{(2)})-(1+\frac{N}{2}\widetilde{a}_l^{(2)}-\frac{N}{2}\widetilde{b}_l^{(2)})\widetilde{\alpha}_l^{(2)}\\ &=[\sum_{j\in\mathcal{I}_{y_i}}(a_{l,j,i}^{(2)}-\widetilde{a}_l^{(2)})-\sum_{j\in\mathcal{I}_{\overline{y_i}}}(b_{l,j,i}^{(2)}-\widetilde{b}_l^{(2)})]\widetilde{\alpha}_l^{(2)}+(1+\sum_{j\in\mathcal{I}_{y_i}}a_{l,j,i}^{(2)}-\sum_{j\in\mathcal{I}_{\overline{y_i}}}b_{l,j,i}^{(2)})h(l)\widetilde{\alpha}_l^{(2)}\\ &\overset{(ii)}{\le}\big[\frac{N}{2}f(l+1)\widetilde{a}_l^{(2)}+\frac{N}{2}f(l+1)\widetilde{b}_l^{(2)}\big]\widetilde{\alpha}_l^{(2)}+\big[1+\frac{N}{2}(1+f(l+1))\widetilde{a}_l^{(2)}-\frac{N}{2}(1-f(l+1))\widetilde{b}_l^{(2)}\big]h(l)\widetilde{\alpha}_l^{(2)}\\ &=[f(l+1)+N\widetilde{a}_l^{(2)}h(l)+f(l+1)h(l)]\widetilde{\alpha}_l^{(2)}\\ &\overset{(iii)}{=}\Big[\big(1+\frac{f(l+1)}{N\widetilde{a}_l^{(2)}}\big)\big(1+h(l)\big)-1\Big]\widetilde{\alpha}_{l+1}^{(2)},\end{aligned}$$

where (i) is by the error bound of $\alpha_{l,i}^{(2)}$ at layer l, i.e., $|\alpha_{l,i}^{(2)}|-\widetilde{\alpha}_l^{(2)}\le h(l)\cdot\widetilde{\alpha}_l^{(2)}$, (ii) is by the error bound of $a_l^{(2)},b_l^{(2)}$ at layer l i.e., $|a_l^{(2)}-\widetilde{a}_l^{(2)}|\le f(l+1)\cdot\widetilde{a}_l^{(2)},\ |b_l^{(2)}-\widetilde{b}_l^{(2)}|\le f(l+1)\cdot\widetilde{b}_l^{(2)}$, and

(iii) is by $\widetilde{\alpha}_{l+1}^{(2)} = \frac{\widetilde{\alpha}_l^{(2)}}{N\widetilde{a}_l^{(2)}}$ based on (E.11). Meanwhile, we have

$$
\begin{aligned}
|\alpha_{l+1,i}^{(2)}| - \widetilde{\alpha}_{l+1}^{(2)} &= (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(2)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(2)})|\alpha_{l,i}^{(2)}| - (1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)})\widetilde{\alpha}_l^{(2)} \\
&\overset{(i)}{\geq} (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(2)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(2)})(\widetilde{\alpha}_l^{(2)} - h(l)\widetilde{\alpha}_l^{(2)}) - (1 + \frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)})\widetilde{\alpha}_l^{(2)} \\
&= [\sum_{j\in\mathcal{I}_{y_i}} (a_{l,j,i}^{(2)} - \widetilde{a}_l^{(2)}) - \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b_{l,j,i}^{(2)} - \widetilde{b}_l^{(2)})]\widetilde{\alpha}_l^{(2)} - (1 + \sum_{j\in\mathcal{I}_{y_i}} a_{l,j,i}^{(2)} - \sum_{j\in\mathcal{I}_{\overline{y_i}}} b_{l,j,i}^{(2)})h(l)\widetilde{\alpha}_l^{(2)} \\
&\overset{(ii)}{\geq} [-\frac{N}{2}f(l+1)\widetilde{a}_l^{(2)} - \frac{N}{2}f(l+1)\widetilde{b}_l^{(2)}]\widetilde{\alpha}_l^{(2)} \\
&\quad - [1 + \frac{N}{2}(1 + f(l+1))\widetilde{a}_l^{(2)} - \frac{N}{2}(1 - f(l+1))\widetilde{b}_l^{(2)}]h(l)\widetilde{\alpha}_l^{(2)} \\
&= [-f(l+1) - N\widetilde{a}_l^{(2)}h(l) - f(l+1)h(l)]\widetilde{\alpha}_l^{(2)} \\
&\overset{(iii)}{=} -\Big[\big(1 + \frac{f(l+1)}{N\widetilde{a}_l^{(2)}}\big)\big(1 + h(l)\big) - 1\Big]\widetilde{\alpha}_{l+1}^{(2)},
\end{aligned}
$$

where (i) is by the error bound of $\alpha_{l,i}^{(2)}$ at layer l, (ii) is by the error bound of $a_{l,j,i}^{(2)}, b_{l,j,i}^{(2)}$ at layer l, and (iii) is by $\widetilde{\alpha}_{l+1} = \frac{\widetilde{\alpha}_l}{N\widetilde{a}_l}$ based on (E.11). Then,

$$
\big||\alpha_{l+1,i}^{(2)}| - \widetilde{\alpha}_{l+1}^{(2)}\big| \leq \Big[\big(1 + \frac{f(l+1)}{N\widetilde{a}_l^{(2)}}\big)\big(1 + h(l)\big) - 1\Big] \cdot \widetilde{\alpha}_{l+1}^{(2)}.
$$

It follows that by $h(l) \leq (1 + \frac{1+p^2}{4})^l - 1,\ f(l+1) \leq 1/2$ in (E.26) and $\frac{1}{N\widetilde{a}_l} \leq \frac{1+p^2}{2}$, then we have $\big(1 + \frac{f(l+1)}{N\widetilde{a}_l^{(2)}}\big)\big(1 + h(l)\big) - 1 \leq (1 + \frac{1+p^2}{4})^{l+1} - 1$, namely the property (E.21) holds for layer $l+1$. It follows that

$$
\begin{aligned}
\big(1 + \frac{f(l+1)}{N\widetilde{a}_l^{(2)}}\big)\big(1 + h(l)\big) - 1 &\overset{(i)}{\leq} f(l+1) + \frac{3}{2}h(l) \\
&\overset{(ii)}{\leq} \frac{6 \times 3^{2l}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) + \frac{3}{2} \times \frac{\frac{4}{5} \times 3^{2l}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) \\
&\leq \frac{\frac{4}{5} \times 3^{2l+2}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2),
\end{aligned}
$$

where (i) is by $\frac{1}{N\widetilde{a}_l^{(2)}} \leq \frac{1+p^2}{2} \leq 1$ and $f(l+1) \leq 1/2$, (ii) is by (E.25). Then (E.18) holds for layer $l+1$.

For the error bound of $\beta^{(2)}_{l+1,i,k}$, we have that for $\beta^{(2)}_{l+1,i,k},\ k \in \mathcal{I}_{y_i} \setminus \{i\}$, by its changing rules across layers (E.11) and (E.15), we have that

$$
\begin{aligned}
\beta^{(2)}_{l+1,i,k} - \widetilde{\beta}^{(2)}_{l+1,+} =& -\widetilde{a}^{(2)}_l + (\beta^{(2)}_{l,i,k} - \widetilde{\beta}^{(2)}_{l,+}) + \sum_{j\in\mathcal{I}_{y_i}} (a^{(2)}_{l,j,i}\beta^{(2)}_{l,j,k} - \widetilde{a}^{(2)}_l \widetilde{\beta}^{(2)}_{l,+}) + \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b^{(2)}_{l,j,i}\beta^{(2)}_{l,j,k} - \widetilde{b}^{(2)}_l \widetilde{\beta}^{(2)}_{l,-}) \\
=& -\widetilde{a}^{(2)}_l + (\beta^{(2)}_{l,i,k} - \widetilde{\beta}^{(2)}_{l,+}) + \widetilde{a}^{(2)}_l \sum_{j\in\mathcal{I}_{y_i}} (\beta^{(2)}_{l,j,k} - \widetilde{\beta}^{(2)}_{l,+}) + \sum_{j\in\mathcal{I}_{y_i}} (a^{(2)}_{l,j,i} - \widetilde{a}^{(2)}_l)\beta^{(2)}_{l,j,k} \\
&+ \widetilde{b}^{(2)}_l \sum_{j\in\mathcal{I}_{\overline{y_i}}} (\beta^{(2)}_{l,j,k} - \widetilde{\beta}^{(2)}_{l,-}) + \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b^{(2)}_{l,j,i} - \widetilde{b}^{(2)}_l)\beta^{(2)}_{l,j,k} \\
\overset{(i)}{\leq}& f(l+1)\widetilde{a}^{(2)}_l + g(l)\widetilde{\beta}^{(2)}_{l,+} + \frac{N}{2}\widetilde{a}^{(2)}_l g(l)\widetilde{\beta}^{(2)}_{l,+} + \frac{N}{2} f(l+1)\widetilde{a}^{(2)}_l \big(1+g(l)\big)\widetilde{\beta}^{(2)}_{l,+} \\
&+ \frac{N}{2}\widetilde{b}^{(2)}_l g(l)\widetilde{\beta}^{(2)}_{l,-} + \frac{N}{2} f(l+1)\widetilde{b}^{(2)}_l \big(1+g(l)\big)\widetilde{\beta}^{(2)}_{l,-} \\
\overset{(ii)}{\leq}& f(l+1)\widetilde{\beta}^{(2)}_{l+1,+} + g(l)\widetilde{\beta}^{(2)}_{l+1,+} + f(l+1)g(l)\widetilde{\beta}^{(2)}_{l+1,+} \\
=& [f(l+1) + g(l) + f(l+1)g(l)] \cdot \widetilde{\beta}^{(2)}_{l+1,+},
\end{aligned}
$$

where (i) is by the error bound of $a^{(2)}_{l,j,i}, b^{(2)}_{l,j,i}, \widetilde{\beta}^{(2)}_{l,i,k}$ at layer l, and (ii) is by the changing rules for $\widetilde{\beta}^{(2)}_{l,+}$ across layers (E.15). Meanwhile, for $\beta^{(2)}_{l+1,i,k},\ k \in \mathcal{I}_{y_i} \setminus \{i\}$, we have that

$$
\begin{aligned}
\beta^{(2)}_{l+1,i,k} - \widetilde{\beta}^{(2)}_{l+1,+} \overset{(i)}{\geq}& -f(l+1)\widetilde{a}^{(2)}_l - g(l)\widetilde{\beta}^{(2)}_{l,+} - \frac{N}{2}\widetilde{a}^{(2)}_l g(l)\widetilde{\beta}^{(2)}_{l,+} - \frac{N}{2} f(l+1)\widetilde{a}^{(2)}_l \big(1+g(l)\big)\widetilde{\beta}^{(2)}_{l,+} \\
&- \frac{N}{2}\widetilde{b}^{(2)}_l g(l)\widetilde{\beta}^{(2)}_{l,-} - \frac{N}{2} f(l+1)\widetilde{b}^{(2)}_l \big(1+g(l)\big)\widetilde{\beta}^{(2)}_{l,-} \\
\overset{(ii)}{\geq}& -f(l+1)\widetilde{\beta}^{(2)}_{l+1,+} - g(l)\widetilde{\beta}^{(2)}_{l+1,+} - f(l+1)g(l)\widetilde{\beta}^{(2)}_{l+1,+} \\
=& [-f(l+1) - g(l) - f(l+1)g(l)] \cdot \widetilde{\beta}^{(2)}_{l+1,+},
\end{aligned}
$$

where (i) is by the error bound of $a^{(2)}_{l,j,i}, b^{(2)}_{l,j,i}, \widetilde{\beta}^{(2)}_{l,i,k}$ at layer l, and (ii) is by the changing rules for $\widetilde{\beta}^{(2)}_{l,+}$ across layers (E.15). Similarly, for $\beta^{(2)}_{l+1,i,k},\ k \in \mathcal{I}_{\overline{y_i}}$, we have

$$
\begin{aligned}
\beta^{(2)}_{l+1,i,k} - \widetilde{\beta}^{(2)}_{l+1,i,k} =& -\widetilde{b}^{(2)}_l + (\beta^{(2)}_{l,i,k} - \widetilde{\beta}^{(2)}_{l,-}) + \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b^{(2)}_{l,j,i}\beta^{(2)}_{l,j,k} - \widetilde{b}^{(2)}_l \widetilde{\beta}^{(2)}_{l,+}) + \sum_{j\in\mathcal{I}_{y_i}} (a^{(2)}_{l,j,i}\beta^{(2)}_{l,j,k} - \widetilde{a}^{(2)}_l \widetilde{\beta}^{(2)}_{l,-}) \\
=& -\widetilde{b}^{(2)}_l + (\beta^{(2)}_{l,i,k} - \widetilde{\beta}^{(2)}_{l,-}) + \widetilde{b}^{(2)}_l \sum_{j\in\mathcal{I}_{\overline{y_i}}} (\beta^{(2)}_{l,j,k} - \widetilde{\beta}^{(2)}_{l,+}) + \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b^{(2)}_{l,j,i} - \widetilde{b}^{(2)}_l)\beta^{(2)}_{l,j,k} \\
&+ \widetilde{a}^{(2)}_l \sum_{j\in\mathcal{I}_{y_i}} (\beta^{(2)}_{l,j,k} - \widetilde{\beta}^{(2)}_{l,-}) + \sum_{j\in\mathcal{I}_{y_i}} (a^{(2)}_{l,j,i} - \widetilde{a}^{(2)}_l)\beta^{(2)}_{l,j,k} \\
\overset{(i)}{\leq}& f(l+1)\widetilde{b}^{(2)}_l + g(l)\widetilde{\beta}^{(2)}_{l,-} + \frac{N}{2}\widetilde{b}^{(2)}_l g(l)\widetilde{\beta}^{(2)}_{l,+} + \frac{N}{2} f(l+1)\widetilde{b}^{(2)}_l \big(1+g(l)\big)\widetilde{\beta}^{(2)}_{l,+} \\
&+ \frac{N}{2}\widetilde{a}^{(2)}_l g(l)\widetilde{\beta}^{(2)}_{l,-} + \frac{N}{2} f(l+1)\widetilde{a}^{(2)}_l \big(1+g(l)\big)\widetilde{\beta}^{(2)}_{l,-} \\
\overset{(ii)}{\leq}& f(l+1)\widetilde{\beta}^{(2)}_{l+1,-} + g(l)\widetilde{\beta}^{(2)}_{l+1,-} + f(l+1)g(l)\widetilde{\beta}^{(2)}_{l+1,-} \\
\leq& [f(l+1) + g(l) + f(l+1)g(l)] \cdot \widetilde{\beta}^{(2)}_{l+1,-},
\end{aligned}
$$

where (i) is by the error bound of $a_{l,j,i}^{(2)}, b_{l,j,i}^{(2)}, \beta_{l,j,k}^{(2)}$ at layer l, and (ii) is by the changing rules for $\widetilde{\beta}_{l,-}^{(2)}$ across layers (E.15). Meanwhile, for $\beta_{l+1,i,k}^{(2)},\ k \in \mathcal{I}_{\overline{y_i}}$, we also have

$$\begin{aligned}
\beta_{l+1,i,k}^{(2)} - \widetilde{\beta}_{l+1,i,k}^{(2)} =& -\widetilde{b}_l^{(2)} + (\beta_{l,i,k}^{(2)} - \widetilde{\beta}_{l,-}^{(2)}) + \widetilde{b}_l^{(2)} \sum_{j\in\mathcal{I}_{\overline{y_i}}} (\beta_{l,j,k}^{(2)} - \widetilde{\beta}_{l,+}^{(2)}) + \sum_{j\in\mathcal{I}_{\overline{y_i}}} (b_{l,j,i}^{(2)} - \widetilde{b}_l^{(2)})\beta_{l,j,k}^{(2)} \\
&+ \widetilde{a}_l^{(2)} \sum_{j\in\mathcal{I}_{y_i}} (\beta_{l,j,k}^{(2)} - \widetilde{\beta}_{l,-}^{(2)}) + \sum_{j\in\mathcal{I}_{y_i}} (a_{l,j,i}^{(2)} - \widetilde{a}_l^{(2)})\beta_{l,j,k}^{(2)} \\
\overset{(i)}{\geq}& -f(l+1)\widetilde{b}_l^{(2)} - g(l)\widetilde{\beta}_{l,-}^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)} g(l)\widetilde{\beta}_{l,+}^{(2)} - \frac{N}{2} f(l+1)\widetilde{b}_l^{(2)}\big(1+g(l)\big)\widetilde{\beta}_{l,+}^{(2)} \\
&- \frac{N}{2}\widetilde{a}_l^{(2)} g(l)\widetilde{\beta}_{l,-}^{(2)} - \frac{N}{2} f(l+1)\widetilde{a}_l^{(2)}\big(1+g(l)\big)\widetilde{\beta}_{l,-}^{(2)} \\
\overset{(ii)}{\geq}& -f(l+1)\widetilde{\beta}_{l+1,-}^{(2)} - g(l)\widetilde{\beta}_{l+1,-}^{(2)} - f(l+1)g(l)\widetilde{\beta}_{l+1,-}^{(2)} \\
\geq& [-f(l+1) - g(l) - f(l+1)g(l)] \cdot \widetilde{\beta}_{l+1,-}^{(2)},
\end{aligned}$$

where (i) is by the error bound of $a_l^{(2)}, b_l^{(2)}, \beta_{l,j,k}^{(2)}$ at layer l, and (ii) is by the changing rules for $\widetilde{\beta}_{l,-}^{(2)}$ across layers (E.15). Therefore,

$$\begin{aligned}
|\beta_{l+1,i,k}^{(2)} - \widetilde{\beta}_{l+1,i,k}^{(2)}| &\leq [f(l+1) + g(l) + f(l+1)g(l)] \cdot \widetilde{\beta}_{l+1,+}^{(2)},\ k \in \mathcal{I}_{y_i}, \\
|\beta_{l+1,i,k}^{(2)} - \widetilde{\beta}_{l+1,i,k}^{(2)}| &\leq [f(l+1) + g(l) + f(l+1)g(l)] \cdot \widetilde{\beta}_{l+1,-}^{(2)},\ k \in \mathcal{I}_{\overline{y_i}}.
\end{aligned}$$

It follows that by $g(l) \leq (\frac{3}{2})^l - 1,\ f(l+1) \leq \frac{1}{2}$ in (E.26), we have $f(l+1) + g(l) + f(l+1)g(l) \leq (\frac{3}{2})^{l+1} - 1$, then (E.20) holds for layer $l+1$. Also,

$$\begin{aligned}
f(l+1) + g(l) + f(l+1)g(l) &\leq \frac{3}{2} g(l) + f(l+1) \\
&\leq \frac{3}{2} \times \frac{\frac{4}{5}\times 3^{2l}}{d \cdot d^{\frac{3}{8}} \cdot \mathrm{SNR}_0^4} (m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) + \frac{6\times 3^{2l}}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2) \\
&\leq \frac{\frac{4}{5}\times 3^{2(l+1)}}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2),
\end{aligned}$$

where the first inequality is by $f(l+1) \leq \frac{1}{2}$ in (E.26), the second inequality is by (E.25). Thus, we have that (E.17) holds for layer $l+1$.

For the error bound for $a_{l+1,j,i}^{(2)}, b_{l+1,j,i}^{(2)}$ in the Softmax matrix: For $j \in \mathcal{I}_{y_i}$,

$$\begin{aligned}
a_{l+1,j,i}^{(2)} =& \frac{\exp({\mathbf{h}_{l+1,j}^{(2)}}^\top \mathbf{W}_{l+1}^{(2)} \mathbf{h}_{l+1,i}^{(2)})}{\sum_{j'=1}^N \exp({\mathbf{h}_{l+1,j'}^{(2)}}^\top \mathbf{W}_{l+1}^{(2)} \mathbf{h}_{l+1,i}^{(2)})} \\
\overset{(i)}{\leq}& \frac{2}{N} \cdot \frac{\exp\big(\frac{4h(l+1)(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\big)\exp\big(\frac{[1+g(l+1)]^2(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1 + \exp\big(-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4 - 2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)\exp\big(-2(h(l+1))^2(\widetilde{\alpha}_{l+1}^{(2)})^2\big)} \\
\leq& \frac{2}{N} \cdot \frac{\exp\big(\frac{[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1 + \exp\big(-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4 - 2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)} \\
\overset{(ii)}{\leq}& \frac{2}{N} \cdot \frac{1 + \frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)}{1 + \exp\big(-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4 - 2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)} \\
=& \Big\{1 + \frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)\Big\} \cdot \widetilde{a}_{l+1}^{(2)},
\end{aligned}$$

where (i) is by the error bound of $\alpha_{l+1,i}^{(2)}, \beta_{l+1,i,k}^{(2)}$ proved above and (E.23), (ii) is by $\exp(x) \leq 1+2x$ for $0 \leq x \leq 1$ and the fact that $\{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2\}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)/(d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4) \leq 1/2$ that is proved as follows: By $h(l+1) \leq (1+\frac{1+p^2}{4})^{l+1}-1$ and $g(l+1) \leq (\frac{3}{2})^{l+1}-1$ proved above, we have that $4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2 \leq 3\times(\frac{3}{2})^{2(l+1)}$, and by $\widetilde{\alpha}_{l+1}^{(2)} \leq 2^{l+1}$ and condition in (E.22), we have

$$\begin{aligned}
&\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}^{(l+1)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\\
&\leq \frac{6\times 3^{2(l+1)}}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\\
&\leq \frac{1}{2}.
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
a_{l+1,j,i}^{(2)} =& \frac{\exp({\mathbf{h}_{l+1,j}^{(2)}}^\top \mathbf{W}_{l+1}^{(2)}\mathbf{h}_{l+1,i}^{(2)})}{\sum_{j'=1}^N \exp({\mathbf{h}_{l+1,j'}^{(2)}}^\top \mathbf{W}_{l+1}^{(2)}\mathbf{h}_{l+1,i}^{(2)})}\\
\overset{(i)}{\geq}& \frac{2}{N}\cdot \frac{\exp\big(\frac{-4h(l+1)(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)\exp\big(-\frac{[1+g(l+1)]^2(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp\big(-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)\exp\big(-2(h(l+1))^2(\widetilde{\alpha}_{l+1}^{(2)})^2\big)}\\
\geq& \frac{2}{N}\cdot \frac{\exp\big(-\frac{[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp\big(-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)}\\
\overset{(ii)}{\geq}& \frac{2}{N}\cdot \frac{1-\frac{[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp\big(-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4-2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)}\\
\geq& \Big\{1-\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\Big\}\cdot \widetilde{a}_{l+1}^{(2)},
\end{aligned}$$

where (i) is by the error bound of $\alpha_{l+1,i}^{(2)}, \beta_{l+1,i,k}^{(2)}$ proved above and (E.23), (ii) is by $\exp(x) \geq 1+x,\ \forall x$.

Similarly, for $j \in \mathcal{I}_{\overline{y_i}}$, we have

$$\begin{aligned}
b_{l+1,j,i}^{(2)} =& \frac{\exp({\mathbf{h}_{l+1,j}^{(2)}}^\top \mathbf{W}_{l+1}^{(2)}\mathbf{h}_{l+1,i}^{(2)})}{\sum_{j'=1}^N \exp({\mathbf{h}_{l+1,j'}^{(2)}}^\top \mathbf{W}_{l+1}^{(2)}\mathbf{h}_{l+1,i}^{(2)})}\\
\overset{(i)}{\leq}& \frac{2}{N}\cdot \frac{\exp\big(\frac{4h(l+1)(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)\exp\big(\frac{[1+g(l+1)]^2(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp\big(2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4+2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)\exp\big(2(h(l+1))^2(\widetilde{\alpha}_{l+1}^{(2)})^2\big)}\\
\leq& \frac{2}{N}\cdot \frac{\exp\big(\frac{[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp\big(2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4+2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)}\\
\overset{(ii)}{\leq}& \frac{2}{N}\cdot \frac{1+\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp\big(2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_1\|\boldsymbol{\mu}\|_2^4+2(\widetilde{\alpha}_{l+1}^{(2)})^2 m_2\|\boldsymbol{\mu}\|_2^2\big)}\\
\leq& \Big\{1+\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\Big\}\cdot \widetilde{b}_{l+1}^{(2)},
\end{aligned}$$

where (i) is by the error bound of $\alpha_{l+1,i}^{(2)}, \beta_{l+1,i,k}^{(2)}$ proved above and (E.23), (ii) is by $\exp(x) \leq 1+2x$ for $0 \leq x \leq 1$ and the fact that $\{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2\}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)/(d\cdot d^{\frac{3}{8}}\cdot \mathrm{SNR}_0^4) \leq 1/2$ that is proved above. On the other hand,

$$
\begin{aligned}
b_{l+1,j,i}^{(2)} &= \frac{\exp(\mathbf{h}_{l+1,j}^{(2)\top}\mathbf{W}_{l+1}^{(2)}\mathbf{h}_{l+1,i}^{(2)})}{\sum_{j'=1}^{N}\exp(\mathbf{h}_{l+1,j'}^{(2)\top}\mathbf{W}_{l+1}^{(2)}\mathbf{h}_{l+1,i}^{(2)})} \\
&\overset{(i)}{\geq} \frac{2}{N}\cdot\frac{\exp\big(\frac{-4h(l+1)(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)\exp\big(-\frac{[1+g(l+1)]^2(\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp\big(2(\widetilde{\alpha}_{l+1}^{(2)})^2m_1\|\boldsymbol{\mu}\|_2^4+2(\widetilde{\alpha}_{l+1}^{(2)})^2m_2\|\boldsymbol{\mu}\|_2^2\big)\exp\big(2(h(l+1))^2(\widetilde{\alpha}_{l+1}^{(2)})^2\big)} \\
&\geq \frac{2}{N}\cdot\frac{\exp\big(-\frac{[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\big)}{1+\exp\big(2(\widetilde{\alpha}_{l+1}^{(2)})^2m_1\|\boldsymbol{\mu}\|_2^4+2(\widetilde{\alpha}_{l+1}^{(2)})^2m_2\|\boldsymbol{\mu}\|_2^2\big)} \\
&\overset{(ii)}{\geq} \frac{2}{N}\cdot\frac{1-\frac{[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)}{1+\exp\big(2(\widetilde{\alpha}_{l+1}^{(2)})^2m_1\|\boldsymbol{\mu}\|_2^4+2(\widetilde{\alpha}_{l+1}^{(2)})^2m_2\|\boldsymbol{\mu}\|_2^2\big)} \\
&\geq \Big\{1-\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\Big\}\cdot\widetilde{b}_{l+1}^{(2)},
\end{aligned}
$$

where (i) is by the error bound of $\alpha_{l+1}^{(2)}, \beta_{l+1,i,k}^{(2)}$ proved above and (E.23), (ii) is by $\exp(x) \geq 1+x,\ \forall x$. Thus,

$$
\begin{aligned}
|a_{l+1,j,i}^{(2)}-\widetilde{a}_{l+1}^{(2)}| &\leq \left[\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\right]\cdot\widetilde{a}_{l+1}^{(2)}, \\
|b_{l+1,j,i}^{(2)}-\widetilde{b}_{l+1}^{(2)}| &\leq \left[\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2)\right]\cdot\widetilde{b}_{l+1}^{(2)}.
\end{aligned}
$$

As proved above,

$$
\begin{aligned}
&\frac{2[4h(l+1)+2(h(l+1))^2+(1+g(l+1))^2](\widetilde{\alpha}_{l+1}^{(2)})^2}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2) \\
&\quad\leq \frac{6\times 3^{2(l+1)}}{d\cdot d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^4}(m_1\|\boldsymbol{\mu}\|_2^4+m_2\|\boldsymbol{\mu}\|_2^2),
\end{aligned}
$$

which indicates (E.16) holds for layer $l+1$. It follows that by condition in (E.22), $f(l+2) \leq 1/2$ then (E.19) holds for layer $l+1$.

Therefore, by induction, (E.16)-(E.18) and (E.19)-(E.21) hold for all layer $l,\ l \in [L]$. Since (E.16)-(E.18) hold for all $l \in [L]$, by condition in (E.22), we have that

$$
f(l+1) = o(1), \quad g(l) = o(1), \quad h(l) = o(1),\ l \in [L].
$$

This concludes the proof of this Lemma D.6. □

E.4 Proof of the second part of Theorem 3.2: SNR change across layers

Given the above results, in this section, we prove the second point of Theorem 3.2.

Proof of Theorem 3.2. To compare the SNR between layer l and layer $l+1$, by the definition of SNR (D.4), we have that the lower bound for the ratio of SNRs between layer l and layer $l+1$ is

$$\begin{aligned}\frac{\mathrm{SNR}_{l+1}^{(t)}}{\mathrm{SNR}_{l}^{(t)}} &= \frac{\frac{|\alpha_{l+1,i}^{(t)}|\cdot\|\boldsymbol{\mu}\|_2}{\|\sum_{k=1}^N \beta_{l+1,i,k}^{(t)}\boldsymbol{\xi}_k\|_2}}{\frac{|\alpha_{l,i}^{(t)}|\cdot\|\boldsymbol{\mu}\|_2}{\|\sum_{k=1}^N \beta_{l,i,k}^{(t)}\boldsymbol{\xi}_k\|_2}} \\ &\overset{(i)}{\geq} \frac{\frac{|\alpha_{l+1,i}^{(t)}|}{\sqrt{\sum_{k=1}^N {\beta_{l+1,i,k}^{(t)}}^2}\sqrt{1+c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}}}\cdot\frac{\|\boldsymbol{\mu}\|_2}{\sigma_{\mathrm{noise}}\sqrt{d}}}{\frac{|\alpha_{l,i}^{(t)}|}{\sqrt{\sum_{k=1}^N {\beta_{l,i,k}^{(t)}}^2}\sqrt{1-c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}}}\cdot\frac{\|\boldsymbol{\mu}\|_2}{\sigma_{\mathrm{noise}}\sqrt{d}}} \\ &\overset{(ii)}{\geq} \sqrt{1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}}\cdot\frac{|\alpha_{l+1,i}^{(t)}|}{|\alpha_{l,i}^{(t)}|}\cdot\frac{\sqrt{\sum_{k=1}^N {\beta_{l,i,k}^{(t)}}^2}}{\sqrt{\sum_{k=1}^N {\beta_{l+1,i,k}^{(t)}}^2}},\end{aligned} \tag{E.27}$$

where (i) is by Lemma G.4, and (ii) is by $\frac{1+c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}}{1+c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}} \geq 1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}$.

By Lemma D.4 and the definition of T^*, we have that $|\alpha_{l+1,i}^{(t)}| \geq |\alpha_{l+1,i}^{(2)}| - (t-2)\exp(-c_6\cdot 2^L) \geq |\alpha_{l+1,i}^{(2)}| - \exp(-c\cdot 2^L)$. Similarly, $|\alpha_{l,i}^{(t)}| \leq |\alpha_{l,i}^{(2)}| + (t-2)\exp(-c_6\cdot 2^L) \leq |\alpha_{l,i}^{(2)}| + \exp(-c\cdot 2^L)$. It follows that

$$\frac{|\alpha_{l+1,i}^{(t)}|}{|\alpha_{l,i}^{(t)}|} \geq \frac{|\alpha_{l+1,i}^{(2)}| - \exp(-c\cdot 2^L)}{|\alpha_{l,i}^{(2)}| + \exp(-c\cdot 2^L)} \geq \frac{|\alpha_{l+1,i}^{(2)}|}{|\alpha_{l,i}^{(2)}|}\big[1-2\exp(-c\cdot 2^L)\big]. \tag{E.28}$$

Meanwhile, by Lemma D.4, we have that $\beta_{l,i,k}^{(t)} \geq \beta_{l,i,k}^{(2)} - (t-2)\exp(-c_7\cdot 2^L) \geq \beta_{l,i,k}^{(2)} - \exp(-c'\cdot 2^L)$, and then $\sum_{k=1}^N {\beta_{l,i,k}^{(t)}}^2 \geq \sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2 - 3\cdot 2^l\exp(-c'\cdot 2^L) \geq \sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2 - \exp(-c\cdot 2^L)$. Also, we have that $\beta_{l+1,i,k}^{(t)} \leq \beta_{l+1,i,k}^{(2)} - \exp(-c'\cdot 2^L)$ and $\sum_{k=1}^N {\beta_{l+1,i,k}^{(t)}}^2 \leq \sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2 + \exp(-c\cdot 2^L)$. Thus, we have

$$\frac{\sum_{k=1}^N {\beta_{l,i,k}^{(t)}}^2}{\sum_{k=1}^N {\beta_{l+1,i,k}^{(t)}}^2} \geq \frac{\sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2 - \exp(-c\cdot 2^L)}{\sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2 + \exp(-c\cdot 2^L)} \geq \frac{\sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2}{\sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2}\big[1-2\exp(-c\cdot 2^L)\big]. \tag{E.29}$$

Moreover, by Lemma D.6, we have that $|\alpha_{l+1,i}^{(2)}| \geq \big[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]\cdot\widetilde{\alpha}_{l+1}^{(2)}$ and $|\alpha_{l,i}^{(2)}| \leq \big[1+\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]\cdot\widetilde{\alpha}_{l}^{(2)}$, then we have $\frac{|\alpha_{l+1,i}^{(2)}|}{|\alpha_{l,i}^{(2)}|} \geq \big[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]\cdot\frac{\widetilde{\alpha}_{l+1}^{(2)}}{\widetilde{\alpha}^{(l)}}$. Similarly, by Lemma D.6, $\sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2 \geq \big[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]\cdot\sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2$, and $\sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2 \leq \big[1+\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]\cdot\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2$, then we have

$$\frac{|\alpha_{l+1,i}^{(2)}|^2}{|\alpha_{l,i}^{(2)}|^2}\cdot\frac{\sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2}{\sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2} \geq \big[1-2\exp(-c\cdot 2^L)\big]^3\cdot\big[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]^3\cdot\frac{(\widetilde{\alpha}_{l+1}^{(2)})^2}{(\widetilde{\alpha}_{l}^{(2)})^2}\cdot\frac{\sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2}{\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2}. \tag{E.30}$$

Therefore, combined with (E.27), we have

$$\begin{aligned}\Big(\frac{\mathrm{SNR}_{l+1}^{(t)}}{\mathrm{SNR}_{l}^{(t)}}\Big)^2 &\overset{(i)}{\geq} \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot\frac{|\alpha_{l+1,i}^{(2)}|^2}{|\alpha_{l,i}^{(2)}|^2}\cdot\frac{\sum_{k=1}^N {\beta_{l,i,k}^{(2)}}^2}{\sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2}\\ &\overset{(ii)}{\geq} \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot\big[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]^3\cdot\frac{(\widetilde{\alpha}_{l+1}^{(2)})^2}{(\widetilde{\alpha}_{l}^{(2)})^2}\cdot\frac{\sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2}{\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2}\\ &\overset{(iii)}{=} \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot\big[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}\big]^3\cdot\frac{(N\widetilde{a}_{l}^{(2)})^2\sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2}{\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2},\end{aligned}\tag{E.31}$$

where (i) is by (E.28) and (E.29), and (ii) is by (E.30), (iii) is by $|\widetilde{\alpha}_{l+1,i}^{(2)}|/|\widetilde{\alpha}_{l,i}^{(2)}| = 1+\frac{N}{2}\widetilde{a}_l^{(2)} - \frac{N}{2}\widetilde{b}_l^{(2)} = 1+N\widetilde{a}_l^{(2)} - (\frac{N}{2}\widetilde{a}_l^{(2)}+\frac{N}{2}\widetilde{b}_l^{(2)}) = N\widetilde{a}_l^{(2)}$ based on (E.11).

For the last term in (E.31), by the change of $\widetilde{\beta}_{l,i,k}^{(2)}$ across layers (E.15), we have

$$\sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2 = \frac{N}{2}(\widetilde{\beta}_{l,+}^{(2)})^2 + 2\widetilde{\beta}_{l,+}^{(2)} + 1 + \frac{N}{2}(\widetilde{\beta}_{l,-}^{(2)})^2,\tag{E.32}$$

and

$$\begin{aligned}\frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2 &= \frac{1}{(N\widetilde{a}_l^{(2)})^2}(\frac{N}{2}(\widetilde{\beta}_{l+1,+}^{(2)})^2 + 2\widetilde{\beta}_{l+1,+}^{(2)} + 1 + \frac{N}{2}(\widetilde{\beta}_{l+1,-}^{(2)})^2)\\ &= \frac{1}{(N\widetilde{a}_l^{(2)})^2}\frac{N}{2}[N\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \widetilde{a}_l^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}(\widetilde{\beta}_{l,+}^{(2)}+\widetilde{\beta}_{l,-}^{(2)})]^2\\ &\quad + \frac{2}{(N\widetilde{a}_l^{(2)})^2}[N\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \widetilde{a}_l^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}(\widetilde{\beta}_{l,+}^{(2)}+\widetilde{\beta}_{l,-}^{(2)})] + \frac{1}{(N\widetilde{a}_l^{(2)})^2}\\ &\quad + \frac{1}{(N\widetilde{a}_l^{(2)})^2}\frac{N}{2}[N\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)} + \widetilde{b}_l^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}(\widetilde{\beta}_{l,+}^{(2)}+\widetilde{\beta}_{l,-}^{(2)})]^2,\end{aligned}\tag{E.33}$$

where the first equality is by (E.14), the last equality is by $\widetilde{\beta}_{l+1,+}^{(2)} = \widetilde{a}_l^{(2)} + \widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)} = N\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \widetilde{a}_l^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}(\widetilde{\beta}_{l,+}^{(2)}+\widetilde{\beta}_{l,-}^{(2)})$ and $\widetilde{\beta}_{l+1,-}^{(2)} = \widetilde{b}_l^{(2)} + \widetilde{\beta}_{l,-}^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}\widetilde{\beta}_{l,+}^{(2)} + \frac{N}{2}\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)} = N\widetilde{a}_l^{(2)}\widetilde{\beta}_{l,-}^{(2)} + \widetilde{b}_l^{(2)} + \frac{N}{2}\widetilde{b}_l^{(2)}(\widetilde{\beta}_{l,+}^{(2)}+\widetilde{\beta}_{l,-}^{(2)})$ by (E.15).

Denote $\Delta^{(l)} := \sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2 - \frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2$, then the lower bound of the ratio of SNRs between l-th and $(l+1)$-th layer (E.31) can be further written as

$$\begin{aligned}(\frac{\mathrm{SNR}_{l+1}^{(t)}}{\mathrm{SNR}_{l}^{(t)}})^2 &\geq \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}]^3\cdot\frac{\sum_{k=1}^N(\widetilde{\beta}_{l,i,k}^{(2)})^2}{\frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^N {\beta_{l+1,i,k}^{(2)}}^2}\\ &= \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}]^3\cdot[1+\frac{\Delta^{(l)}}{\frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2}].\end{aligned}\tag{E.34}$$

By (E.32) and (E.33), the numerator is

$$
\begin{aligned}
\Delta^{(l)} &= \sum_{k=1}^{N}(\widetilde{\beta}_{l,i,k}^{(2)})^2 - \frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^{N}(\widetilde{\beta}_{l+1,i,k}^{(2)})^2 \\
&\overset{(i)}{=} 2\widetilde{\beta}_{l,+}^{(2)} + 1 - \frac{1}{N\widetilde{a}_l^{(2)}}\left\{2\widetilde{\beta}_{l,+}^{(2)} + N\widetilde{\beta}_{l,+}^{(2)}\big[\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)}(2^l-1)\big] + N\widetilde{\beta}_{l,-}^{(2)}\big[\widetilde{b}_l^{(2)} + \widetilde{b}_l^{(2)}(2^l-1)\big]\right\} \\
&\quad - \frac{1}{(N\widetilde{a}_l^{(2)})^2}\left\{\frac{N}{2}\big[\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)}(2^l-1)\big]^2 + 2\big[\widetilde{a}_l^{(2)} + \widetilde{b}_l^{(2)}(2^l-1)\big] + 1 + \frac{N}{2}\big[\widetilde{b}_l^{(2)} + \widetilde{b}_l^{(2)}(2^l-1)\big]^2\right\} \\
&= 1 - \frac{2}{N}(2^l-1)\big[\frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}} + \frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}}(2^l-1)\big] - \frac{1}{2N}\big[1 + \frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}}(2^l-1)\big]^2 - \frac{2}{N^2}\big[\frac{1}{\widetilde{a}_l^{(2)}} + \frac{\widetilde{b}_l^{(2)}}{(\widetilde{a}_l^{(2)})^2}(2^l-1)\big] \\
&\quad - \frac{1}{(N\widetilde{a}_l^{(2)})^2} - \frac{1}{2N}\big[\frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}} + \frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}}(2^l-1)\big]^2 \\
&\overset{(ii)}{\geq} 1 - \frac{2}{N}(2^l-1)^2p^2 - \frac{2}{N}(2^l-1)p^2 - \frac{1}{2N}[1+(2^l-1)p^2]^2 - \frac{1}{N}(1+p^2)[1+(2^l-1)p^2] \\
&\quad - \frac{1}{2N}p^4(2^l)^2 - \frac{1}{4}(1+p^2)^2 \\
&= 1 - \frac{1}{N}\left\{\frac{3}{2} + [1 + 4(2^l-1) + 2(2^l-1)^2]p^2 + [\frac{1}{2} + 2(2^l-1) + (2^l-1)^2]p^4\right\} - \frac{1}{4}(1+p^2)^2 \\
&\overset{(ii)}{\geq} 1 - \frac{1}{N}\left\{\frac{3}{2} + 4^l[2p^2 + p^4]\right\} - \frac{1}{4}(1+p^2)^2, \qquad \text{(E.35)}
\end{aligned}
$$

where (i) is by $\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)} = \frac{2}{N}(2^l-1)$. For (ii), denote $p := \exp(-m_1\|\boldsymbol{\mu}\|_2^4 - m_2\|\boldsymbol{\mu}\|_2^2)$ with m_1, m_2 defined in (E.2), by (E.12), we have $\widetilde{\alpha}_l^{(2)} \geq 1$, then $\frac{1}{\widetilde{a}_l^{(2)}} = \frac{N}{2}[1 + \exp(-2(\widetilde{\alpha}_l^{(2)})^2(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2))] \leq \frac{N}{2}[1 + \exp(-2m_1\|\boldsymbol{\mu}\|_2^4 - 2m_2\|\boldsymbol{\mu}\|_2^2)] = \frac{N}{2}(1+p^2)$, $\frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}} = \exp(-2(\widetilde{\alpha}_l^{(2)})^2(m_1\|\boldsymbol{\mu}\|_2^4 + m_2\|\boldsymbol{\mu}\|_2^2)) \leq \exp(-2m_1\|\boldsymbol{\mu}\|_2^4 - 2m_2\|\boldsymbol{\mu}\|_2^2) = p^2$. (iii) is by $\frac{1}{2} + 2(2^l-1) + (2^l-1)^2 \leq (2^l)^2$. Meanwhile, the denominator is given as

$$
\begin{aligned}
\frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^{N}(\widetilde{\beta}_{l+1,i,k}^{(2)})^2 &= \sum_{k=1}^{N}(\widetilde{\beta}_{l,i,k}^{(2)})^2 - \Delta^{(l)} \\
&= \frac{N}{2}(\widetilde{\beta}_{l,+}^{(2)})^2 + 2\widetilde{\beta}_{l,+}^{(2)} + 1 + \frac{N}{2}(\widetilde{\beta}_{l,-}^{(2)})^2 - 1 - \Big\{ - \frac{2}{N}(2^l-1)\big[\frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}} + \frac{\widetilde{b}_l^{(2)}}{\widetilde{a}^{(l)}}(2^l-1)\big] \\
&\quad - \frac{1}{2N}\big[1 + \frac{\widetilde{b}_l^{(2)}}{\widetilde{a}_l^{(2)}}(2^l-1)\big]^2 - \frac{2}{N^2}\big[\frac{1}{\widetilde{a}_l^{(2)}} + \frac{\widetilde{b}_l^{(2)}}{(\widetilde{a}_l^{(2)})^2}(2^l-1)\big] - \frac{1}{4}(1+p^2)^2\Big\} \\
&\overset{(i)}{\leq} \frac{1}{N}\left\{4(2^l-1) + 2(2^l-1)^2 + \frac{3}{2} + [1 + 4(2^l-1) + 2(2^l-1)^2]p^2\right. \\
&\qquad \left. + [\frac{1}{2} + 2(2^l-1) + (2^l-1)^2]p^4\right\} + \frac{1}{4}(1+p^2)^2 \\
&\overset{(ii)}{\leq} \frac{1}{N}\left\{\frac{3}{2} + 4^l[2 + 2p^2 + p^4]\right\} + \frac{1}{4}(1+p^2)^2, \qquad \text{(E.36)}
\end{aligned}
$$

where (i) is by (E.35), $\frac{N}{2}(\widetilde{\beta}_{l,+}^{(2)})^2 + \frac{N}{2}(\widetilde{\beta}_{l,-}^{(2)})^2 \leq \frac{N}{2}(\widetilde{\beta}_{l,+}^{(2)} + \widetilde{\beta}_{l,-}^{(2)})^2 = \frac{2}{N}(2^l-1)^2$, and $\widetilde{\beta}_{l,+}^{(2)} \leq \frac{2}{N}(2^l-1)$; (ii) is by $\frac{1}{2} + 2(2^l-1) + (2^l-1)^2 \leq (2^l)^2$. Thus, plug (E.35) and (E.36) into (E.34),

we have

$$
\begin{aligned}
(\frac{\mathrm{SNR}_{l+1}^{(t)}}{\mathrm{SNR}_{l}^{(t)}})^2 &\geq \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}]^3\cdot[1+\frac{\Delta^{(l)}}{\frac{1}{(N\widetilde{a}_l^{(2)})^2}\sum_{k=1}^N(\widetilde{\beta}_{l+1,i,k}^{(2)})^2}] \\
&\geq \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}]^3\cdot\frac{1+2\frac{4^l}{N}}{\frac{1}{N}\big\{\frac{3}{2}+4^l[2+2p^2+p^4]\big\}+\frac{1}{4}(1+p^2)^2} \\
&\geq \big(1-2c_0\frac{N^{\frac{1}{4}}}{\sqrt{d}}\big)\cdot\big[1-2\exp(-c\cdot 2^L)\big]^3\cdot[1-\frac{\Theta(3^{2L}\cdot\sigma_{\mathrm{noise}}^2)}{d^{\frac{3}{8}}\cdot\mathrm{SNR}_0^2}]^3\cdot\frac{4}{\frac{8}{\sqrt{N}}(1+p^2)^2+(1+p^2)^2},
\end{aligned}
$$

where the last inequality is by Condition 3.1 and $p = \exp(-m_1\|\boldsymbol{\mu}\|_2^4 - m_2\|\boldsymbol{\mu}\|_2^2) = \exp(-\Theta(\|\boldsymbol{\mu}\|_2^2)) \in (0,1)$. Therefore, we have

$$
\frac{\mathrm{SNR}_{l+1}^{(t)}}{\mathrm{SNR}_{l}^{(t)}} \geq \frac{2-\epsilon(N,d,L)}{1+p^2},
$$

where $\epsilon(N,d,L) = O\big(\max\{\frac{N^{\frac{1}{4}}}{\sqrt{d}}, \frac{1}{\sqrt{N}}, \exp(-2^L), d^{-\frac{3}{8}}\}\big)$.

This finish the proof of Theorem 3.2. □

F Proof of Theorem 3.3

In this section, we provide the proof of Theorem 3.3.

Proof of Theorem 3.3. We prove it by applying Theorem 3.1 in Cao et al. (2021) on our dataset $\mathcal{S}_l$, $l = 0,\ldots,L$. For the data input $\mathbf{X}_m$ in the deep transformer (2.1), the last column output at layer l is given as $\mathbf{h}_{l,N,m}^{(2)} = \alpha_{l,N,m}^{(2)}\boldsymbol{\mu} + \sum_{k=1}^N \beta_{l,k,N,m}^{(2)}\boldsymbol{\xi}_{k,m}$. Our data $\mathbf{h}_{l,N,m}^{(2)}, m = 1...M$ also follows a sub-Gaussian mixture. This is because $\mathbf{h}_{l,N,m}^{(2)} = \alpha_{l,N,m}^{(2)}\boldsymbol{\mu} + \sum_{k=1}^N \beta_{l,k,N,m}^{(2)}\boldsymbol{\xi}_{k,m}$, which can be decomposed as $\mathbf{h}_{l,N,m}^{(2)} = y_N\mathbb{E}\big[|\alpha_{l,N,m}^{(2)}|\big]\boldsymbol{\mu} + (\alpha_{l,N,m}^{(2)} - y_N\mathbb{E}[|\alpha_{l,N,m}^{(2)}|])\boldsymbol{\mu} + \sum_{k=1}^N \beta_{l,k,N,m}^{(2)}\boldsymbol{\xi}_{k,m}$. Here the first term is the signal part, and the second and third terms together form the noise part $\widetilde{\mathbf{h}}_{l,N,m}^{(2)} = (\alpha_{l,N,m}^{(2)} - y_N\mathbb{E}\big[|\alpha_{l,N,m}^{(2)}|\big])\boldsymbol{\mu} + \sum_{k=1}^N \beta_{l,k,N,m}^{(2)}\boldsymbol{\xi}_{k,m}$, which has zero mean and covariance matrix $\boldsymbol{\Sigma}_l = \mathrm{Cov}\big(\widetilde{\mathbf{h}}_{l,N,m}^{(2)}, \widetilde{\mathbf{h}}_{l,N,m}^{(2)}\big)$. Given $\alpha_{l,N,m}^{(2)}, \beta_{l,k,N,m}^{(2)}$ are all bounded and $\boldsymbol{\xi}_{k,m}$ are i.i.d Gaussian random variables for all m, it follows that the noise term is sub-Gaussian even if they are not independent, and hence our data $\mathbf{h}_{l,N,m}^{(2)}$ is also sub-Gaussian random variable. Then, by Theorem 3.1 in Cao et al. (2021), we have that for some absolute constant $c_{10} > 0$,

$$
\begin{aligned}
&R_l(\widehat{\boldsymbol{\theta}}_l) \\
&\leq \exp\left(\frac{-c_{10}\cdot M\cdot\|\boldsymbol{\mu}_l\|_2^4}{M\|\boldsymbol{\mu}_l\|_{\boldsymbol{\Sigma}_l}^2 + \|\boldsymbol{\Sigma}_l\|_F^2 + M\|\boldsymbol{\Sigma}_l\|_2^2}\right) \\
&\overset{(i)}{\leq} \exp\left(\frac{-c_{10}\cdot M\cdot\frac{2}{3}(\widetilde{\alpha}_l^{(2)})^4\cdot\|\boldsymbol{\mu}\|_2^4}{\frac{4}{3}M(\widetilde{\alpha}_l^{(2)})^2\|\boldsymbol{\mu}\|_2^4 + \frac{4}{3}\|\boldsymbol{\mu}\|_2^4 + 3\sigma_{\mathrm{noise}}^4 d^2(\sum_{k=1}^N(\widetilde{\beta}_{l,k,N}^{(2)})^2)^2 + M\big[\frac{4}{3}\|\boldsymbol{\mu}\|_2^4 + 3\sigma_{\mathrm{noise}}^4(\sum_{k=1}^N(\widetilde{\beta}_{l,k,N}^{(2)})^2)^2\big]}\right) \\
&\overset{(ii)}{\leq} \exp\left(\frac{-c_{10}\cdot M\cdot(\frac{2}{1+p^2})^{2l}}{2M+2+18\mathrm{SNR}_0^{-4} + M(2+18d^{-2}\mathrm{SNR}_0^{-4})}\right) \\
&\leq \exp\left(\frac{-c_{10}\cdot(\frac{2}{1+p^2})^{2l}}{1+M^{-1}(1+\mathrm{SNR}_0^{-4}) + d^{-2}\mathrm{SNR}_0^{-4}}\right),
\end{aligned}
$$

where (i) is by the results of Lemma D.6, (ii) is by the definition of $\mathrm{SNR}_0 = \|\boldsymbol{\mu}\|_2/(\sigma_{\mathrm{noise}}\sqrt{d})$, $(\widetilde{\alpha}_l^{(2)})^2 \geq 1$, $\widetilde{\alpha}_l^{(2)} \geq (\frac{2}{1+p^2})^l$ and $(\widetilde{\beta}_{l,k,N}^{(2)})^2 = \frac{N}{2}(\widetilde{\beta}_{l,+}^{(2)})^2 + 2\widetilde{\beta}_{l,+}^{(2)} + 1 + \frac{N}{2}(\widetilde{\beta}_{l,-}^{(2)})^2 \leq \frac{2(4^l-1)}{N} + 1 \leq 2$ by Condition 3.1.

This finishes the proof of Theorem 3.3. □

G AUXILIARY LEMMAS

This section presents some auxiliary lemmas that are used in the above analysis of noise reduction performance of the deep transformer model.

Lemma G.1. *For any constant $a, b \in \mathbb{R}$, we have the following matrix can be decomposed in the following way.*

$$\begin{bmatrix} a & \dots & a & b & \dots & b \\ \vdots & & \vdots & \vdots & & \vdots \\ a & \dots & a & b & \dots & b \\ b & \dots & b & a & \dots & a \\ \vdots & & \vdots & \vdots & & \vdots \\ b & \dots & b & a & \dots & a \end{bmatrix}_{N\times N} = \begin{bmatrix} a\mathbf{J}_{N/2} & b\mathbf{J}_{N/2} \\ b\mathbf{J}_{N/2} & a\mathbf{J}_{N/2} \end{bmatrix}_{N\times N}$$

$$= \frac{1}{2}(a+b)\begin{pmatrix} \mathbf{1}_{N/2} \\ \mathbf{1}_{N/2} \end{pmatrix}(\mathbf{1}_{N/2}^\top, \mathbf{1}_{N/2}^\top) + \frac{1}{2}(a-b)\begin{pmatrix} \mathbf{1}_{N/2} \\ -\mathbf{1}_{N/2} \end{pmatrix}(\mathbf{1}_{N/2}^\top, -\mathbf{1}_{N/2}^\top).$$

Lemma G.1 can be easily verified by the characteristic decomposition of the matrix.

Lemma G.2. *Under Condition 3.1, with probability at least $1 - \frac{1}{N^2}\exp(-C_1 \cdot 2^L)$, it holds that for any $\left\|\boldsymbol{\xi}_i\right\|_2^2,\ i \in [N]$ is*

$$\sigma_{\text{noise}}^2 d - C_1^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}} \leq \left\|\boldsymbol{\xi}_i\right\|_2^2 \leq \sigma_{\text{noise}}^2 d + C_1^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}},$$

where $C_1^{'}$ is an absolute constant that does not depend on other variables.

Proof of Lemma G.2. For any $\boldsymbol{\xi}_i, i \in [N]$, since $\boldsymbol{\xi}_i$ i.i.d follows $N(\mathbf{0}, \sigma_{\text{noise}}^2\mathbf{I})$, then by Bernstein inequality, with the probability of at least $1 - \frac{1}{N^2}\exp(-C_1 \cdot 2^L)$, we have that

$$\sigma_{\text{noise}}^2 d - \widetilde{C}_1\sigma_{\text{noise}}^2\sqrt{d}N^{\frac{1}{4}} \leq \|\boldsymbol{\xi}_i\|_2^2 \leq \sigma_{\text{noise}}^2 d + \widetilde{C}_1\sigma_{\text{noise}}^2\sqrt{d}N^{\frac{1}{4}},$$

where $\widetilde{C_1}$ is an absolute constant that does not depend on other variables. By the assumption that $d \geq \Omega(N^2)$ in Condition 3.1, it follows that for any $\left\|\boldsymbol{\xi}_i\right\|_2^2,\ i \in [N]$,

$$\left\|\boldsymbol{\xi}_i\right\|_2^2 \geq \sigma_{\text{noise}}^2 d - \widetilde{C}_1\sigma_{\text{noise}}^2\sqrt{d}N^{\frac{1}{4}} \geq \sigma_{\text{noise}}^2 d - C_1^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}}$$
$$\left\|\boldsymbol{\xi}_i\right\|_2^2 \leq \sigma_{\text{noise}}^2 d + \widetilde{C}_1\sigma_{\text{noise}}^2\sqrt{d}N^{\frac{1}{4}} \leq \sigma_{\text{noise}}^2 d + C_1^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}},$$

where $C_1^{'}$ is an absolute constant that does not depend on other variables. This finishes the proof of this lemma. □

Lemma G.3. *Under Condition 3.1, with probability at least $1 - \frac{1}{N^2}\exp(-C_2 \cdot 2^L)$, it holds that for any $i, j \in [N],\ i \neq j$,*

$$-C_2^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}} \leq \langle\boldsymbol{\xi}_i, \boldsymbol{\xi}_j\rangle \leq C_2^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}},$$

where $C_2^{'}$ is an absolute constant that does not depend on other variables.

Proof of Lemma G.3. For any $i, j \in [N],\ i \neq j$, $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ are i.i.d Gaussian random variables, thus, by Bernstein inequality, with probability at least $1 - \frac{1}{N^2}\exp(-C_2 \cdot 2^L)$, we have that

$$|\langle\boldsymbol{\xi}_i, \boldsymbol{\xi}_j\rangle| \leq \widetilde{C}_2\sigma_{\text{noise}}^2\sqrt{d}N^{\frac{1}{4}},$$

where $\widetilde{C}_2$ is an absolute constant that does not depend on other variables. By the assumption that $d \geq \Omega(N^2)$ in Condition 3.1, it follows that

$$|\langle\boldsymbol{\xi}_i, \boldsymbol{\xi}_j\rangle| \leq \widetilde{C}_2\sigma_{\text{noise}}^2\sqrt{d}N^{\frac{1}{4}} \leq C_2^{'}\sigma_{\text{noise}}^2 d^{\frac{5}{8}},$$

where $C_2^{'}$ is an absolute constant that does not depend on other variables. This finishes the proof of this lemma. □

Lemma G.4. *For any $\overline{\boldsymbol{\xi}}_i = \sum_{k=1}^{N} \beta_{i,k}\boldsymbol{\xi}_k$ with constants $\beta_{i,k} \in \mathbb{R}$, with probability at least $1 - 2\exp(-\sqrt{N})$, it holds that*

$$\overline{\sigma_i}\sigma_{\text{noise}}^2 d \cdot (1 - C_3\frac{N^{\frac{1}{4}}}{\sqrt{d}}) \leq \left\|\overline{\boldsymbol{\xi}}_i\right\|_2^2 \leq \overline{\sigma_i}\sigma_{\text{noise}}^2 d \cdot (1 + C_3\frac{N^{\frac{1}{4}}}{\sqrt{d}}),$$

where C_3 is an absolute constant that does not depend on other variables, and $\overline{\sigma_i} = \sum_{k=1}^{N} \beta_{i,k}^2$.

Proof of Lemma G.4. Since $\boldsymbol{\xi}_k, k \in [N]$ i.i.d follows $N(\mathbf{0}, \sigma_{\text{noise}}^2\mathbf{I})$, then we have that $\overline{\boldsymbol{\xi}_i} = \sum_{k=1}^{N} \beta_{i,k}\boldsymbol{\xi}_k \sim N(\mathbf{0}, \overline{\sigma_i}\sigma_{\text{noise}}^2\mathbf{I})$, where $\overline{\sigma_i} = \sum_{k=1}^{N} \beta_{i,k}^2$. Then by Bernstein inequality, with the probability of at least $1 - 2\exp(-\sqrt{N})$, we have

$$\left|\frac{1}{\overline{\sigma_i}\sigma_{\text{noise}}^2}\left\|\overline{\boldsymbol{\xi}}_i\right\|_2^2 - d\right| \leq C_3\sqrt{d}N^{\frac{1}{4}},$$

where C_3 is an absolute constant, which does not depend on other variables. Thus,

$$\overline{\sigma_i}\sigma_{\text{noise}}^2 d - \overline{\sigma_i}\sigma_{\text{noise}}^2 C_3\sqrt{d}N^{\frac{1}{4}} \leq \left\|\overline{\boldsymbol{\xi}}_i\right\|_2^2 \leq \overline{\sigma_i}\sigma_{\text{noise}}^2 d + \overline{\sigma_i}\sigma_{\text{noise}}^2 C_3\sqrt{d}N^{\frac{1}{4}}.$$

This finishes the proof of this lemma. □