Towards Infinite-Long Prefix in Transformer

Yingyu Liang[†*] **Zhenmei Shi**[‡*] **Zhao Song**[¶*] **Chiwun Yang**[‖*]

[†]The University of Hong Kong, [†,‡]University of Wisconsin-Madison
[¶]University of California, Berkeley, [‖]Sun Yat-sen University

Code: https://github.com/ChristianYang37/NTK-Attention

Contact: [†]yingyul@hku.hk, {[†]yliang,[‡]zmeishi}@cs.wisc.edu,
[¶]magic.linuxkde@gmail.com, [‖]christiannyang37@gmail.com

Abstract

Prompting and context-based fine-tuning methods, which we call Prefix Learning, have been proposed to enhance the performance of language models on various downstream tasks. They are empirically efficient and effective, matching the performance of full parameter fine-tuning, but the theoretical understandings are limited. In this paper, we aim to address this limitation by studying their ability from the perspective of prefix length. In particular, we provide a convergence guarantee for training an ultra-long prefix in a stylized setting using the Neural Tangent Kernel (NTK) framework. Based on this strong theoretical guarantee, we design and implement an algorithm that only needs to introduce and fine-tune a few extra trainable parameters instead of an infinite-long prefix in each layer of a transformer, and can approximate the prefix attention to a guaranteed polynomial-small error. Preliminary experimental results on vision, natural language, and math data show that our method achieves superior or competitive performance compared to existing methods like full parameters fine-tuning, P-Tuning V2, and LoRA. This demonstrates our method is promising for parameter-efficient fine-tuning.

1 Introduction

The advent of Large Language Models (LLMs) and Vision LLMs (vLLMs) has significantly advanced the field of Artificial Intelligence (AI), with prominent examples like ChatGPT (ChatGPT, 2022), Claude (Claude-3, 2024), Gemini (Gemini, 2024). They have exhibited impressive performances across a spectrum of tasks, encompassing chat systems (Maaz et al., 2023; Xu et al., 2023; Zheng et al., 2024), text-to-image conversion (Qiao et al., 2019; Frolov et al., 2021; Zhang et al., 2023a), AI mathematical inference (Hendrycks et al., 2020; Xiong et al., 2022; Yu et al., 2023a; Yao et al., 2023; Xiong et al., 2023b), and many more. However, despite these advancements, pre-existing LLMs often fall short in specialized domains that demand a deeper understanding of professional knowledge (Li et al., 2024b; Wang et al., 2024) . This has led to the development of fine-tuning/adaptation (Shi et al., 2022) methodologies aimed at enhancing the proficiency of these models in executing more specialized tasks (Mangrulkar et al., 2022). Several notable contributions in this area, such as LoRA (Low-Rank Adaptation, Hu et al. (2021)), P-Tuning (Liu et al., 2021b, 2023), and $(IA)^3$ (Liu et al., 2022), have displayed performances rivaling those of full-parameter fine-tuning techniques. This underscores the potential of these fine-tuning strategies to further refine the capabilities of Large Language Models.

Among the methods proposed, most context-based fine-tuning methods, e.g., Prompt-Tuning (Lester et al., 2021; Liu et al., 2021a), Prefix-Tuning (Li and Liang, 2021), P-Tuning (Liu et al., 2023, 2021b), use enhanced input sequences (or virtual prompt, a.k.a soft prompt) to optimize their model outputs. These methods are gaining significant interest due to their ease of implementation across various model architectures, and also prevention of catastrophic forgetting with static pretrained parameters (Wang et al., 2023b; Sohn et al., 2023; Yang et al., 2024). We call the above approaches **Prefix Learning** since they improve the performance by optimizing a prefix matrix added to the input in each attention layer of the LLMs (see detailed formulation in Section 3).

Despite its wide use and strong empirical performance, we still have a limited understanding of why and how prefix learning operates (Wang et al., 2023a; Petrov et al., 2024a,b). One common phenomenon in prior empirical studies is that prefix learning results in better downstream performance when the prefix length increases (Lester et al., 2021; Liu et al., 2023). We call this phenomenon *scaling*

[*]Equal contribution in alphabetical order.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 11127–11191
November 4-9, 2025 ©2025 Association for Computational Linguistics

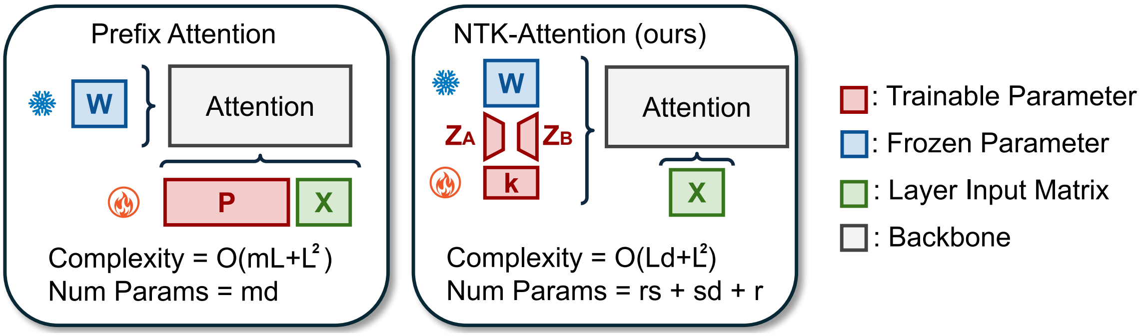


Figure 1: Illustration of existing prefix attention methods (Algorithm 1) and our NTK-Attention (Algorithm 2). Compared to the former, NTK-Attention significantly reduces the number of parameters and the time complexity. Here, $X \in \mathbb{R}^{L\times d}$ is the input of this layer, $W = [W_Q, W_K, W_V]$ is frozen weights of attention, $P \in \mathbb{R}^{m\times d}$ is the trainable prefix matrix and $Z_A \in \mathbb{R}^{r\times s}$, $Z_B \in \mathbb{R}^{s\times d}$, $k \in \mathbb{R}^r$ are the trainable parameters in our method. L is the input length, d the input dimension, m the prefix length, and r a hyperparameter in NTK-attention (i.e., the dimension of the constructed feature mapping; see Section 5). Note that $m \gg L$ and $m \gg d$, and $r = \mathrm{poly}(d)$ (usually be chosen to d or $2d$), $s \le \lfloor d/2 \rfloor$ (low-rank of Z_A, Z_B) are used in our experiments.

law in prefix learning: the longer the prefix, the larger downstream dataset the model can fit, and thus the better performance the model would have. Then intuitively, we would like to ask:

What happens when the prefix length is large or even tends to infinity?

The answer to this cannot be directly figured out via empirical evaluations, since it is impractical to implement networks with ultra-long or even infinite prefixes in practice. Therefore, we first perform a theoretical analysis of prefix learning. We study the optimization of ultra-long prefix learning via the Neural Tangent Kernel (NTK) technique (Jacot et al., 2018), which has been used for analyzing overparameterized networks and thus is suitable for ultra-long prefix learning. Based on the insights gained from the analysis, we propose our method, NTK-attention, which reparameterizes prefix learning and can approximate infinite-long prefix learning using a finite number of parameters. We also conduct some empirical evaluations of our method on vision, natural language understanding, and math inference datasets to demonstrate its effectiveness. We have the following contributions:

- We first perform a theoretical analysis of optimizing an ultra-long prefix in a stylized attention network; see Section 4. We consider a simplified attention network, and show that when prefix length m is sufficiently large (i.e., prefix learning is sufficiently over-parameterized), the training can be analyzed via NTK, which leads to our theoretical guarantee of convergence to small errors. This also provides theoretical support for scaling law in prefix learning.
- We then propose our NTK-Attention (Algorithm 2), motivated by the above strong theoretical guarantee; see Section 5. Our method approximates existing prefix attention (Algorithm 2) by utilizing three trainable parameters Z_A, Z_B and k, to replace the parameter in prefix attention (the prefix matrix P). This allows scaling the prefix length without large memory usage and computational time that increases with the prefix length. It reduces the computation complexity from $O(mL)$ to $O(L^2)$, where L is the input length and m is the prefix length. See Figure 1 for an illustration.
- We further conduct experiments on vision, language and math datasets to verify our theoretical results; see Section 6 and Section 7. The experiments include (1) a comparison among our NTK-Attention, full parameters fine-tuning, and LoRA on CIFAR-100, Food-101 and Tiny-Imagenet datasets with the same pre-trained ViT backbone; (2) a comparison among our NTK-Attention, P-Tuning V2, and LoRA on SuperGLUE, WikiText-103, Penn TreeBank and LAMBADA datasets with the same pre-trained ChatGLM3-6B and OPT-{125M, 350M, 1.3B, 2.7B, 6.7B} family; (3) a comparison among our NTK-Attention and LoRA on GSM8K and MATH datasets with supervised fine-tune pre-trained models LLAMA-3.2; (4) an ablation study to validate sensitivity of hyper-parameters in NTK-Attention; (5) a comparison of the computational costs between our method and standard prefix learning on random

data. The empirical results show that on average our NTK-Attention method achieves better performance than the competitors. For example, on SuperGLUE datasets, it achieves an average accuracy that is 1.07% higher than LoRA and 12.94% higher than P-Tuning V2. It is also observed that our method maintains low time and memory costs while those of prefix learning scales with prefix length. The experimental results demonstrate that our method is effective and efficient and supports our theoretical analysis.

2 Related Work

Prefix Learning. Prefix Learning (Lester et al., 2021; Ding et al., 2021; Wang et al., 2022b; Zhou et al., 2022; Liu et al., 2021a; Petrov et al., 2024a; Wu et al., 2023; Xiong et al., 2023a), including Prompt-Tuning (Lester et al., 2021), Prefix-Tuning (Li and Liang, 2021), P-Tuning (Liu et al., 2023, 2021b), Reweighted In-Context Learning (RICL) (Chu et al., 2023) and so on, is proposed to enhance the performance of language models on the downstream tasks and to reduce the costs of computational resources of fine-tuning the whole model. Those methods optimize task-specific prompts for downstream task improvement. On the other hand, besides the Parameter-Efficient-Fine-Tuning (PEFT) approaches (Mangrulkar et al., 2022) we mentioned above, Retrieval Augmented Generation (RAG) (Lewis et al., 2020; Jiang et al., 2023; Gao et al., 2023b) and Chain-of-Thought (CoT) prompting (Wei et al., 2022b; Wang et al., 2022a; Fu et al., 2022) can also be considered as prefix learning. We conclude all these works to an optimization problem that improves the prefix based on task-specific measurements.

Neural Tangent Kernel. Neural Tangent Kernel (NTK) (Jacot et al., 2018) studies the gradient flow of neural networks in the training process. They showed neural networks are equivalent to Gaussian processes in the infinite-width limit at initialization. A bunch of works has explained the strong performance and the learning ability of neural networks at over-parameterization, such as (Li and Liang, 2018; Du et al., 2019; Song and Yang, 2019; Allen-Zhu et al., 2019; Wei et al., 2019; Bietti and Mairal, 2019; Lee et al., 2020; Chizat and Bach, 2020; Shi et al., 2021; Zhou et al., 2021; Seleznova and Kutyniok, 2022; Gao et al., 2023a; Li et al., 2024a; Shi et al., 2024b) and many more. Furthermore, Arora et al. (2019) gave the first exact algorithm on computing Convolutional NTK (CNTK), Alemohammad et al. (2020) proposed Recurrent NTK, and Hron et al. (2020) presented infinite attention via NNGP and NTK for attention networks. These works have demonstrated advanced performance by utilizing NTK in different neural network architectures. In particular, Malladi et al. (2023) have studied the training dynamic of fine-tuning LLMs via NTK and confirmed the efficiency of such methods.

Theory of Understanding Large Language Models. Since the complicated transformer-based architecture and stochastic optimization process of LLMs lead the study of their behaviors to be a challenge, analyzing LLMs through some theoretical guarantee helps in providing insights to improve and design the next generation of AI systems. This topic includes efficient LLMs (Alman and Song, 2023, 2024a,b; Han et al., 2024; Kacham et al., 2023; Addanki et al., 2023; Deng et al., 2024b; Shi et al., 2024a), optimization of LLMs (Deng et al., 2023; Li et al., 2024a), white-box transformers (Yu et al., 2023b,c; Ferrando et al., 2024; Pai et al., 2024), analysis of emergent abilities of LLMs (Brown et al., 2020; Wei et al., 2022a; Allen-Zhu and Li, 2023a,c,b, 2024), etc. Especially, (Alman and Song, 2023) proved that the hardness of attention can be achieved within $n^{1+o(1)}$ times executions, one effective way is to construct a high-order polynomial mapping based on Taylor expansion of the exponential function $\exp(\cdot)$, and it inspired the design of our NTK-Attention.

3 Preliminary: General Prefix Learning

In this section, we provide the detailed formulation for prefix learning, which optimizes prefix matrices in the attention layers of transformer-based LLMs.

Transformer Network with Prefix. Let $X \in \mathbb{R}^{L\times d}$ be an input matrix to the transformer network, where L and d are the input length and dimension. An N-layer transformer network with an initial-layer prefix matrix $P_{(0)} \in \mathbb{R}^{m_{(0)}\times d}$ and a positional embedding matrix $E \in \mathbb{R}^{(m_{(0)}+L)\times d}$ first concatenate $S_{(0)} := \left[P_{(0)}^\top, X^\top\right]^\top \in \mathbb{R}^{(m_{(0)}+L)\times d}$, then the output of the whole model is defined as: $f_{\mathcal{T}}(X) := \mathsf{TF}_{(N)} \circ \cdots \circ \mathsf{TF}_{(1)}(S_{(0)} + E)$, where the ℓ-th layer of transformer block for $\ell \in [N]$, is then given by: $\mathsf{TF}_{(\ell)}(X) := \mathsf{FF}_{(\ell)} \circ \mathsf{PAttn}_{(\ell)}(X)$. Denote $L' := m_{(0)} + L$. The feed-forward network: $\mathsf{FF}_{(\ell)}(X) := \mathsf{ReLU}(XW_{(\ell),1} + \mathbf{1}_{L'} b_{(\ell),1}^\top)W_{(\ell),2}^\top$

$+ \mathbf{1}_{L'} b_{(\ell),1}^\top + X$, where $W_{(\ell),1}, W_{(\ell),2} \in \mathbb{R}^{d \times d_1}$, $b_1 \in \mathbb{R}^{d_1}, b_2 \in \mathbb{R}^d$, and d_1 represent the hidden dimension of feed-forward. For simplicity, we consider only the single-head attention network with prefix learning:

$$\mathsf{PAttn}_{(\ell)}(X) := \mathsf{Softmax}(\frac{Q_{(\ell)} K_{(\ell),P}^\top}{\sqrt{d}}) V_{(\ell),P} + X,$$

where $Q_{(\ell)} := X W_{(\ell),Q} \in \mathbb{R}^{L' \times d}, K_{(\ell),P} = S_{(\ell)} W_{(\ell),K}, V_{(\ell),V} = S_{(\ell)} W_{(\ell),Q} \in \mathbb{R}^{(m_{(\ell)}+L') \times d}$. Notably, $S_{(\ell)} := \left[P_{(\ell)}^\top, X^\top\right]^\top \in \mathbb{R}^{(m_{(\ell)}+L') \times d}$ be the concatenation of the prefix and the input, and trainable prefix matrix $P_{(\ell)} \in \mathbb{R}^{m_{(\ell)} \times d}$ stands for $m_{(\ell)}$ virtual token vectors (or soft prompt). $W_{(\ell),Q}, W_{(\ell),K}, W_{(\ell),V} \in \mathbb{R}^{d \times d}$ are query, key, and value parameter matrices in ℓ-layer, respectively. Specifically, the length of prefix matrix $m_{(\ell)} \geq 0$ for $\ell \in \{0, \cdots, N\}$ in each layer can be personalized due to some specific requirement.

General Prefix Learning Framework. The concept of Prefix Learning contains P-Tuning (Liu et al., 2023, 2021b), Prefix-Tuning (Li and Liang, 2021). Essentially, they are all searching for the optimal prefix for a specific task based on a strong pre-trained language model. We note the pre-trained weights $\theta = \{W_{(\ell),Q}, W_{(\ell),K}, W_{(\ell),V}, W_{(\ell),1}, W_{(\ell),2}, b_{(\ell),1}, b_{(\ell),2}\}_{\ell=1}^N$ are all frozen during the optimization process of prefix learning. Besides, prefix parameters of the whole model, $\theta_p = \{P_{(\ell)}\}_{\ell=0}^N$ are trainable or adjustable for parameter-efficient-fine-tuning. Denote the dataset as $\mathcal{D}_{\mathrm{pl}} = \{(X_i, Y_i)\}_{i=1}^n$ where n is the dataset size, and $X_i, Y_i \in \mathbb{R}^{L \times d}$. Let $\gamma(\cdot, \cdot)$ denote the loss function for the specific task (e.g., prompting, context-based fine-tuning, etc). The training objective of prefix learning is:

$$\min_{\theta_p} \mathcal{L}_{\mathrm{pl}}(\theta_p) := \sum_{i=1}^n \gamma(f_{\mathcal{T}}(X_i), Y_i). \quad (1)$$

4 Scaling Law in Prefix Learning

In this section, we first introduce an interesting phenomenon, *scaling law in prefix learning*, commonly observed by some prior prefix learning works in Section 4.1. Thus, we give our explanation utilizing the Neural Tangent Kernel(NTK) framework (Jacot et al., 2018) in Section 4.2, where NTK is a popular tool in analyzing overparameterized neural networks, including LLM (Malladi et al., 2023).

4.1 Previous Observation

A rich line of studies (Lester et al., 2021; Liu et al., 2023; Reynolds and McDonell, 2021; Arora et al., 2022; Brown et al., 2020; Dong et al., 2022; Von Oswald et al., 2023; Fu et al., 2022; Agarwal et al., 2024) have reported a common observation that as the prefix length increases, the model's ability to master complex skills also improves. Specifically, the performance of fine-tuned models is enhanced when the prefix length grows within a certain range. A similar trend is observed in prompting methods and in-context learning, where longer and more complex prompts lead to better inference abilities in LLMs, and providing more examples in ICL results in improved performance. We summarize this as the *scaling law in prefix learning*: the longer the prefix length for fine-tuning, the larger dataset the model can fit, thus, the more complicated skill it can master. This motivates investigating prefix learning with long prefixes.

4.2 Theoretical Guarantee via NTK

In our theory, we consider only fine-tuning one certain prefix matrix $P_{(\ell)} \in \theta_p$ for $0 \leq \ell \leq N$, where fine-tuning the whole θ_p is a composition case of the former learning process. On the other hand, we choose $\gamma(X, Y) := \|X - Y\|_F^2$ as the training objective for searching the optimal prefix. Moreover, we give a mild assumption in the NTK framework (also used in (Malladi et al., 2023)):

Assumption 4.1. *We define a matrix $H^* \in \mathbb{R}^{n \times n}$ and its (i,j)-th entry $\forall i, j \in [n]$ is $H^*_{i,j} := \langle \mathrm{vec}(\frac{\mathrm{d}\gamma(f_{\mathcal{T}}(X_i), Y_i)}{\mathrm{d}P_{(\ell)}}), \mathrm{vec}(\frac{\mathrm{d}\gamma(f_{\mathcal{T}}(X_j), Y_j)}{\mathrm{d}P_{(\ell)}})\rangle$, where $\mathrm{vec}(\cdot)$ flattens any matrix row-wise to a vector. We assume H^* is a positive definite (PD) matrix such that its minimum eigenvalue is positive $\lambda := \lambda_{\min}(H^*) > 0$.*

Then the informal version of our main theorem is given as follows:

Theorem 4.2 (Main result, informal version of Theorem K.2)**.** *Let $B := O(\sqrt{\log(nd/\delta)})$, $\epsilon, \delta \in (0, 0.1)$. For any prefix matrix $P_{(\ell)} \in \theta_p$ for $\ell \in [0, N]$, we choose $m_{(\ell)} = \lambda^{-2}\,\mathrm{poly}(n, d, \exp(B))$. If Assumption 4.1 holds, then fine-tuning $P_{(\ell)}$ using gradient descent over $T = \Omega((m_{(\ell)} \eta \lambda)^{-1} \log(nd/\epsilon))$ iterations with learning rate $\eta = \lambda m_{(\ell)}^{-1} / \mathrm{poly}(n, d, \exp(B))$ leads $\mathcal{L}_{\mathrm{pl}}(\theta_p) \leq \epsilon$.*

Discussion. Theorem 4.2 mainly describes the following fact for any dataset with n data points. After initializing the prefix matrix from a normal distribution, assuming the minimum eigenvalue of NTK $\lambda > 0$, setting $m_{(\ell)}$ to be a large enough value so that the network is sufficiently over-parameterized. Then with proper learning rate, the loss can be minimized in finite training time to an arbitrarily small error ϵ. Corresponding to the real-world implementation, it explains that adequately long prefix learning can master downstream tasks when fine-tuning LLMs. Furthermore, it also helps us understand the working mechanism of prefix learning, inspiring us to explore the direction of using ultra-long prefixes.

Now we connect our theory to the *scaling law in prefix learning*. Following (Kaplan et al., 2020), we focus on the relationship between the loss and the computational cost. We provide a theoretical confirmation of the scaling law in prefix learning.

Proposition 4.3 (Scaling Law in Prefix Learning)**.** *We define* $\mathsf{N} := O(\sum_{\ell=0}^{N} m_{(\ell)} d)$ *as the number of parameters,* $\mathsf{D} := O(n)$ *as the size of training dataset,* $\mathsf{C}_{\mathrm{cpt}} := O(\mathsf{N}\mathsf{D}T)$ *as the total compute cost, and* $\alpha := nd$*. We choose* T *as Theorem 4.2, then the loss of training, denotes* L*, satisfies:*

$$\mathsf{L} \approx \frac{\alpha}{\exp(\frac{1}{\alpha}\eta\lambda\mathsf{C}_{\mathrm{cpt}})}.$$

Proposition 4.3, following Theorem 4.2, shows that the training loss of the prefix learning converges exponentially as we increase the computational cost $\mathsf{C}_{\mathrm{cpt}}$, which primarily depends on the number of parameters and the training time in prefix learning, further indicating a possible relationship for formulating scaling law in prefix learning.

Algorithm 1 Prefix Attention

Input: Input matrix $X \in \mathbb{R}^{L \times d}$
Parameters: Frozen query, key and value weights $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$, trainable prefix matrix $P \in \mathbb{R}^{m \times d}$
Output: Exact output $\mathsf{Attn} \in \mathbb{R}^{L \times d}$

procedure PREFIXATTEN(X)
  $S \leftarrow \left[P^\top, X^\top\right]^\top$
  $Q, K_P, V_P \leftarrow XW_Q, SW_K, SW_V$
  $A \leftarrow \exp(QK_P^\top/\sqrt{d})$
  $D \leftarrow \mathrm{diag}(A\mathbf{1}_{m+L})$
  return $D^{-1}AV_P$
end procedure

Algorithm 2 NTK-Attention

Input: Input matrix $X \in \mathbb{R}^{L \times d}$
Parameters: Frozen query, key and value weights $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$, trainable weights $Z_A \in \mathbb{R}^{r \times s}, Z_B \in \mathbb{R}^{s \times d}$ and $k \in \mathbb{R}^r$
Output: Approx output $T \in \mathbb{R}^{L \times d}$

procedure NTK-ATTEN(X)
  $Q, K, V \leftarrow XW_Q, XW_K, XW_V,$
  $\widehat{A} \leftarrow \exp(QK^\top/\sqrt{d})$
  $\widehat{D} \leftarrow \mathrm{diag}(\widehat{A}\mathbf{1}_L + \Phi(Q)k)$
  $T \leftarrow \widehat{D}^{-1}(\widehat{A}V + \Phi(Q)Z_A \cdot Z_B)$
  return T
end procedure

5 NTK-Attention: Approximate Infinite-Long Prefix Attention

The preceding section discussed the convergence guarantee of training sufficiently long prefixes in transformer-based models. This strong theoretical property inspires us to scale up the prefix length m (We omit the notation of layer number (ℓ) in further derivations). However, such prefix learning (Algorithm 1) necessitates a time complexity of $O(mLd + L^2d)$ in each layer of the model, this is impractical due to a large m. This section proposes an approximate algorithm to make long prefix learning practical. Our algorithm, NTK-Attention, is designed to output an approximation of $\mathsf{PAttn}(X)$ in each layer of the model within $O(L^{1+o(1)})$ and without using the long prefix matrix P. We present the derivation and motivation of our algorithm in Section 5.1, formalize the NTK-Attention algorithm in Section 5.2, and provide an approximation guarantee in Section 5.3.

5.1 Derivation: Replacing Prefix P with Trainable Parameters Z, k

There exists a wealth of attention approximation algorithms capable of executing attention computations within $n^{1+o(1)}$ time (Han et al., 2024; Liang et al., 2024a,b). However, our focus lies predominantly with the polynomial method (Tsai et al., 2019; Katharopoulos et al., 2020). Our method has exhibited exceptional performance in terms of time and space complexity through the use of a streaming algorithm.

Polynomial method. In attention networks, the query, key, and value state matrices, denoted as $Q, K, V \in \mathbb{R}^{L \times d}$, are assumed to have all entries bounded (Alman and Song, 2023). Under this condition, the polynomial method first constructs a linear mapping $\phi : \mathbb{R}^d \to \mathbb{R}^r$, where $r = \mathrm{poly}(d)$ (Alman and Song, 2023), and it satisfies the following relation ($i, j \in [L]$, $Q_i, K_j \in \mathbb{R}^d$ represent the i-th row of Q and the j-th row of K respectively):

$$\phi(Q_i)^\top \phi(K_j) \approx \exp(Q_i^\top K_j/\sqrt{d}). \quad (2)$$

Here, the mapping $\phi(\cdot)$ is constructed based on the Taylor expansion of the exponential function, and the larger value of $r \geq d$ would bring the approximation (Eq. (2)) with a smaller error. This is guaranteed by Lemma 3.4 in Alman and Song (2023), refer to a copy in Lemma L.7. The i-th row of the approximate attention (denoted as $\mathsf{PolyAttn}_i \in \mathbb{R}^{1\times d}$) then can be computed as follows: $\mathsf{PolyAttn}_i := \frac{\phi(Q_i)^\top \sum_{j=1}^{L} \phi(K_j) V_j^\top}{\phi(Q_i)^\top \sum_{j=1}^{L} \phi(K_j)} \in \mathbb{R}^{1\times d}, \forall i \in [L]$.

Recall that given an input matrix $X \in \mathbb{R}^{L\times d}$, thus, $Q = XW_Q$, and we have $\begin{bmatrix} K_P, V_P \end{bmatrix} = \begin{bmatrix} P \\ X \end{bmatrix} \cdot \begin{bmatrix} W_K, W_V \end{bmatrix} = \begin{bmatrix} PW_K & PW_V \\ XW_K & XW_V \end{bmatrix}$. Let $K_C := PW_K, V_C := PW_V \in \mathbb{R}^{m\times d}$ and $K := XW_K, V := XW_V \in \mathbb{R}^{L\times d}$. We expand the i-th row of the prefix attention, $\mathsf{PAttn}_i(X) \in \mathbb{R}^{1\times d}$ as:

$$\begin{aligned} &\mathsf{PAttn}_i(X) \\ =& \frac{\exp(Q_i^\top K^\top/\sqrt{d})V + \exp(Q_i^\top K_C^\top/\sqrt{d})V_C}{\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \exp(Q_i^\top K_C^\top/\sqrt{d})\mathbf{1}_m} \\ \approx& \frac{\exp(Q_i^\top K^\top/\sqrt{d})V + \phi(Q_i)^\top Z}{\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \phi(Q_i)^\top k} \end{aligned}$$

where $Z \in \mathbb{R}^{r\times d}, k \in \mathbb{R}^r$, and

$$Z = \sum_{j=1}^{m} \phi(K_{C,j}) V_{C,j}^\top, \quad k = \sum_{j=1}^{m} \phi(K_{C,j}). \quad (3)$$

Here, the first step explicitly computes the softmax function, and the second step holds since replacing $\exp(Q_i^\top K^\top/\sqrt{d})$ by Eq. (2), which is

$$\exp(Q_i^\top K_{C,j}^\top/\sqrt{d}) \approx \phi(Q_i)^\top \phi(K_{C,j}), \forall j \in [m].$$

Therefore, checking the training process of P, we observe that P is updating iff Z and k are updating. Hence, we can replace P by utilizing **trainable parameters** Z and k in Eq. (3) to reparameterize the prefix attention. This is the key to how NTK-Attention approximates prefix attention without a large number of parameters.

5.2 Algorithm

Our analysis shows that as the prefix length approaches infinity, the model performance converges to an optimal point. Rather than implementing impractically long prefixes, we leverage polynomial approximation of the exponential attention kernel to capture this behavior efficiently. This reformulation allows us to replace a large prefix matrix P with compact parameters Z and k that approximate infinite-length prefix effects while requiring only $O(rd + r)$ parameters instead of $O(md)$.

To present our algorithm, based on ϕ, we define: $\Phi(A) = \begin{bmatrix} \phi(A_{1,*}), \cdots, \phi(A_{L,*}) \end{bmatrix}^\top \in \mathbb{R}^{L\times r}, \forall A \in \mathbb{R}^{L\times d}$. Below we present our NTK-Attention method in Algorithm 2, and for comparison also present the traditional prefix attention for prefix learning in Algorithm 1.

Implementation Detail of ϕ. In order to find a balance between approximation and efficient computation of NTK-Attention, we use the first-order polynomial method. Higher-order polynomial approximations could indeed improve performance at the cost of increased computation. Our first-order approximation balances theoretical guarantees with practical efficiency. In particular, we choose $r = d$, and the function ϕ is given by $\phi(z) := d^{-\frac{1}{4}} \cdot (z \circ \mathbf{1}_{z\geq \mathbf{0}_d} + \exp(z) \circ \mathbf{1}_{z<\mathbf{0}_d}) + \mathbf{1}_d \in \mathbb{R}^d, \forall z \in \mathbb{R}^d$, where $\mathbf{1}_{z\geq \mathbf{0}_d} \in \mathbb{R}^d$ is an indicative vector and its i-th entry for $i \in [d]$ equals 1 only when $z_i \geq 0$, and 0 otherwise.

Initialization, Approximation and Training of Z and k. In Section 4.2, we initialize the prefix matrix P from a standard normal distribution following the NTK framework. Since the pretrained weights $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are known, the initialization of Z and k, denotes $Z(0)$ and $k(0)$, can then be computed by Eq. (3). However, consider that Z caches rd parameters for $r = \mathrm{poly}(d)$, which is insufficiently parameter-efficient. In response to it, we choose $s \leq \lfloor d/2 \rfloor$ as an appropriately small integer, then $Z(0) \approx Z_A(0) \cdot Z_B(0)$ is decomposed into two low-rank matrices $Z_A(0) \in \mathbb{R}^{r\times s}, Z_B(0) \in \mathbb{R}^{s\times d}$. For training, let $g_{Z_A}(t) \in \mathbb{R}^{r\times s}, g_{Z_B}(t) \in \mathbb{R}^{s\times d}$ and $g_k(t) \in \mathbb{R}^r$ denote the gradients of $Z_A(t)$, $Z_B(t)$ and $k(t)$ at time t, and η denote the learning rate. Then the update rule is:

$$\begin{aligned} Z_{A,B}(t+1) :=& Z_{A,B}(t) - \eta \cdot g_{Z_{A,B}}(t), \\ k(t+1) :=& k(t) - \eta \cdot g_k(t). \end{aligned}$$

Method	Num Params	Task					Average
		BoolQ	CB	Copa	MultiRC	RTE	
P-Tuning V2 $m = 1$	0.12M	65.69±0.32	67.06±0.37	52.00±1.00	53.59±0.28	65.97±0.22	60.86±0.44
P-Tuning V2 $m = 10$	1.15M	66.67±0.23	74.07±0.00	54.00±0.00	54.17±0.71	66.55±0.25	63.10±0.24
P-Tuning V2 $m = 100$	11.47M	69.42±0.02	74.54±0.47	64.50±0.50	61.62±2.28	76.77±0.83	69.37±0.82
LoRA $r' = 8$	3.67M	**76.52**±0.10	90.23±0.39	86.50±0.50	65.09±0.41	**87.76**±0.37	81.24±0.35
NTK-Attention (ours), $r = 128, s = 16$	3.78M	75.06±0.12	**96.04**±0.84	**88.00**±2.00	**65.85**±0.33	86.59±0.52	**82.31**±0.76

Table 1: Performance of different fine-tuning methods on the SuperGLUE datasets. The base model is ChatGLM3-6B. The methods include P-Tuning V2, LoRA, and our NTK-Attention method. The metric on these datasets is accuracy (measured in %). The best score on each dataset is **boldfaced**.

Number of Trainable Parameters. Since given r and s as two hyperparameters in NTK-Attention, for each attention layer in the transformer-based architecture, we denote $\beta := \frac{r}{d}$. The number of trainable parameters could be computed by $(\beta s + \beta + s)d$ where integer $\beta \geq 1$ and $s \leq \lfloor d/2 \rfloor$. This is more flexible when adjusting the practical efficiency needs. For LoRA with its hyperparameter $r' \leq \lfloor d/2 \rfloor$, where r' is the rank number used for approximation, its number of trainable parameters is $4r'd$ and for prefix attention with its hyper-parameter $m \geq 1$, its number of trainable parameters is md in each attention layer. By choosing $(\beta s + \beta + s) \leq 4r'$, the higher efficiency of NTK-Attention compared to LoRA will be satisfied.

5.3 Error Bound and Complexity Reduction

Introducing an ultra-long prefix matrix $P \in \mathbb{R}^{m \times d}$ to satisfy the conditions in Theorem K.2 requires md parameters for

$$m \geq \Omega(\lambda^{-2} \operatorname{poly}(n, d, \exp(B))),$$

while it also bring a $O(m(m+L)d)$ time complexity to compute Algorithm 1. Our NTK-Attention relieve this by replacing P with Z and k, where we state our theoretical guarantee as follows:

Theorem 5.1 (Error bound with reduced time complexity, informal version of Theorem L.2)**.** *Let m denote the prefix length. Given an input matrix $X \in \mathbb{R}^{L \times d}$ and prefix matrix $P \in \mathbb{R}^{m \times d}$, we denote $Q = XW_Q$, $K_C = PW_K$ and $V_C = PW_V$. If the condition Eq. (3), $\|Q\|_\infty \leq o(\sqrt{\log m})$, $\|K_C\|_\infty \leq o(\sqrt{\log m})$, $\|V_C\|_\infty \leq o(\sqrt{\log m})$ and $d = O(\log m)$ holds, then Algorithm 2 outputs a matrix $T \in \mathbb{R}^{L \times d}$ within time complexity of $O(L^2 d)$ that satisfies:*

$$\|T - \mathsf{PrefixAttn}(X, P)\|_\infty \leq 1/\operatorname{poly}(m). \quad (4)$$

Furthermore, if we replace the original attention operation (attention computation on input X with $K = XW_K$ and $V = XW_V$) with fast attention algorithms like HyperAttention (Han et al., 2024), then NTK-Attention can be even more efficient, achieving Eq. (4) within complexity $O(L^{1+o(1)}d)$ (see Corollary L.3 for proofs).

6 Empirical Evaluations

In this section, we evaluate our method, NTK-Attention on natural language understanding, math inference, and fine-grained image classification tasks. All our experiments use the Huggingface (Wolf et al., 2019) trainer with AdamW optimizer (Kingma and Ba, 2014), and all optimizer hyperparameters are set to the defaults. "Num Params" in our tables stands for the number of trainable parameters in fine-tuning. We provide more details and hyperparameter choices in Appendix C.

Evaluation on Natural Language Understanding Datasets. In this experiment, we utilize five binary classification datasets in SuperGLUE (Wang et al., 2019) for evaluation: the BoolQ, CB, Copa, MultiRC, and RTE datasets. We use a pre-trained LLM ChatGLM3-6B (Zeng et al., 2022; Du et al., 2022) as the base model. For comparison, we choose P-Tuning V2 (Liu et al., 2023, 2021b) which is a standard prefix learning method, and choose LoRA (Hu et al., 2021) which is a popular parameter-efficient fine-tuning method often achieving state-of-the-art. P-Tuning V2 uses different lengths of virtual prefix $\{1, 10, 100, 200\}$, and LoRA uses rank $r' = 8$. We choose $r = 128$ (the dimension of each head of ChatGLM3-6B) and $s = 16$ for our NTK-Attention.

The results are provided in Table 1. Our NTK-Attention method achieves much higher performance than P-Tuning V2. Interestingly, as m increases, the performance of P-Tuning V2 also improves, which is consistent with our analysis. Our analysis also suggests that NTK-Attention approximates ultra-long prefix learning and thus can perform better than P-Tuning V2. The experimen-

Method	Num Params	Dataset			Average
		CIFAR-100	Food-101	Tiny-Imagenet	
FFT	86.39M	85.15±0.13	84.76±0.07	76.20±0.23	82.04±0.14
LoRA $r' = 16$	7.08M	92.17±0.05	89.38±0.33	88.22±0.09	89.92±0.16
LoRA $r' = 32$	14.16M	92.01±0.20	89.86±0.11	**90.16**±0.12	90.68±0.14
NTK-Attention (ours), $r = 64, s = 32$	7.09M	**92.55**±0.03	**90.57**±0.01	89.46±0.10	**90.86**±0.05

Table 2: Performance of different fine-tuning methods on the CIFAR-100, Food-101 and Tiny-Imagenet datasets. The base model is ViT-Base. The methods include FFT, LoRA, and our method NTK-Attention. The metric is accuracy (measured in %). The best score on each dataset is **boldfaced**.

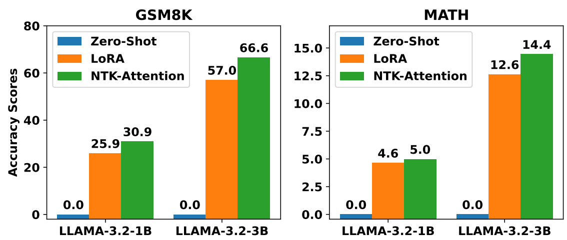


Figure 2: Compare our results with LoRA and Zero-Shot (without using CoT/ICL) on Math inference data. The y-axis is the accuracy.

tal results also show that NTK-Attention achieves better performance than LoRA on CB, Copa, and MultiRC datasets, and achieves better average performance over all the datasets.

Evaluation on Math Inference Datasets. In order to thoroughly verify the effectiveness of NTK-Attention, we conduct experiments on the math inference task, which includes GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) datasets. These are considered fair benchmarks to test the complex capability of LLMs. We follow Yu et al. (2023a) to supervised fine-tune two pretrained models LLAMA-3.2-1B and LLAMA-3.2-3B (Touvron et al., 2023a,b) with dataset MetaMathQA (Yu et al., 2023a). We state our results in Figure 2, and we use accuracy scores for counting the matched answers for evaluation. As we can see, our NTK-attention ($r = d, s = 16$) is better than the two baselines, LoRA and Zero-Shot, where LoRA uses $r' = 16$ for LLAMA-3.2-1B and $r' = 32$ for LLAMA-3.2-3B.

Evaluation on Vision Datasets. We evaluate the method on three image classification datasets: CIFAR-100 (Krizhevsky et al., 2009), Food-101 (Bossard et al., 2014), and Tiny-Imagenet (mnmoustafa, 2017). The base model to be fine-tuned on these datasets is ViT-Base (Dosovitskiy et al., 2020) that is pretrained on the ImageNet-21k (Deng et al., 2009). We compare our method to two baselines: (1) FFT (**F**ull parameters **F**ine-**T**uned) that fine-tunes all parameters; (2) LoRA (Hu et al., 2021) that fine-tunes only query and value weights of the base model with rank $r' = \{16, 32\}$. The results are presented in Table 2. Our method performs much better than FFT: 7.40%, 5.81% and 13.26% higher accuracy on the three datasets, respectively. Note that FFT updates all parameters and has much higher computational costs than LoRA or our method. Our method has a similar performance to LoRA with $r' = 32$, achieving slightly better average accuracy. These results provide positive empirical support for our method.

7 Scalability, Efficiency and Ablation

This basic experimental setup of this section aligns with the one in the previous section. Here, we especially focus on the scalability, efficiency, and how the hyperparameters (r, s) affect the performance.

Evaluation on Language Modeling Tasks. In this experiment, we focus on the scalability of NTK-Attention on a family of language models of different sizes, the OPT family with the model sizes 125M, 350M, 1.3B, 2.7B and 6.7B (Zhang et al., 2022). We introduce three text datasets, which are WikiText-103 (Merity et al., 2016), Penn TreeBank (Marcus et al., 1993), and LAMBADA (Paperno et al., 2016), to compare the scalability of NTK-Attention with LoRA (Hu et al., 2021) and P-Tuning V2 (Liu et al., 2023, 2021b). As we choose $r' = 8$ for LoRA, $m = 32$ for P-Tuning V2, and $r = 2d$ and $s = 10$ for our NTK-Attention, the numbers of trainable parameters are aligned to the same as $32d$ for each attention layer. The results are stated in Table 3, which shows the improvement of NTK-Attention compared to baselines when scaling the model size.

Remark 7.1. *Our theory demonstrates that as prefix length increases, model performance improves until convergence, following the scaling law we identified. While practical prefix lengths (≤ 100), our theory shows that performance even-*

Model	Method	Num Params	Datasets			Average
			WikiText-103	Penn TreeBank	LAMBADA	
OPT-125M	LoRA, $r' = 8$	0.29M	**30.50**	35.97	46.02	37.50
	P-Tuning V2, $m = 32$		2264.22	963.09	1762.19	1663.17
	NTK-Attention, $r = 2d, s = 10$		31.41	**33.52**	**45.39**	**36.77**
OPT-350M	LoRA, $r' = 8$	0.77M	**24.76**	30.41	38.80	31.32
	P-Tuning V2, $m = 32$		7383.48	1339.43	14020.36	7581.09
	NTK-Attention, $r = 2d, s = 10$		25.67	**28.85**	**36.97**	**30.50**
OPT-1.3B	LoRA, $r' = 8$	1.57M	**16.71**	21.27	24.16	20.71
	P-Tuning V2, $m = 32$		2230.76	540.17	3480.77	2083.9
	NTK-Attention, $r = 2d, s = 10$		17.04	**20.09**	**24.04**	**20.39**
OPT-2.7B	LoRA, $r' = 8$	2.62M	15.06	19.61	22.13	18.93
	P-Tuning V2, $m = 32$		772.48	277.99	3378.18	1476.22
	NTK-Attention, $r = 2d, s = 10$		**14.83**	**18.52**	**21.85**	**18.40**
OPT-6.7B	LoRA, $r' = 8$	4.19M	12.81	17.36	19.38	16.52
	P-Tuning V2, $m = 32$		2051.10	409.37	4709.46	2389.98
	NTK-Attention, $r = 2d, s = 10$		**12.56**	**16.68**	**18.81**	**16.02**

Table 3: Performance of different fine-tuning methods on OPT-{125M, 350M, 1.3B, 2.7B, 6.7B} pre-trained models with WikiText-103, Penn TreeBank, and LAMBADA datasets. The metric is perplexity (PPL), with its smaller value standing for better performance. The best score on each dataset and model is **boldfaced**.

tually plateaus as $m \to \infty$, which is precisely why our NTK-Attention method approximates infinite-length prefixes without requiring them explicitly. On the other hand, as the prefix token becomes longer, the trainable parameter will linearly increase. Thus, training ultra-long prefix prompts in a large model size is impractical empirically.

8 Conclusion

In this study, we illuminated the principles of prefix learning for fine-tuning when the prefix length is large. We conducted an in-depth theoretical analysis, demonstrating that when the prefix length is sufficiently large, the attention network is over-parameterized, and the NTK technique can be leveraged to provide a convergence guarantee of prefix learning. Based on these insights, we proposed a novel efficient fine-tuning method called NTK-Attention, approximating prefix attention using two trainable parameters to replace the large prefix matrix, thus significantly mitigating memory usage issues and reducing computational cost for long prefixes. Our empirical results support our theoretical findings, showing NTK-Attention's superior performance on downstream tasks over baselines across natural language, math, and vision datasets.

Discussion. The insights from our analysis of infinite-length prefixes have broader implications for transformer optimization beyond prefix learning. Our theoretical framework using NTK could be applied to:

- Attention mechanism optimization: The mathematical formulation we developed for approximating attention with ultra-long contexts can inform more efficient attention variants beyond our specific implementation, potentially addressing the quadratic complexity bottleneck in standard attention.

- Parameter-efficient adaptation methods: Our findings could extend to other parameter-efficient methods like adapters and LoRA by providing a theoretical understanding of their convergence properties and optimization dynamics.

- Scaling laws for transformers: Our framework helps explain how overparameterization affects optimization, which could contribute to a better understanding of emergent abilities in larger models and inform more efficient scaling strategies.

- Prompt engineering: The relationship we established between prefix length and model capacity could guide automated prompt optimization techniques with theoretical guarantees.

- Model compression: Our approach to approximating large parameter spaces with smaller ones could inform new model compression techniques with provable error bounds.

Limitation

The work has limited experimental analysis and results. While empirical evaluations have been provided for some datasets and LLM models, the proposed method is widely applicable to different data and models, so comprehensive evaluations on more datasets and more practical methods can provide stronger empirical support. Besides, the computational efficiency of NTK-Attention is insufficiently better than prefix attention when $m < d$, since the design of NTK-Attention is toward the ultra-big value of m, such we only compare to the prefix attention with prefix length $m \gg d$ to meet the over-parameterization setting in our analysis.

Acknowledgement

The authors would like to thank all anonymous reviewers and area chairs for their constructive suggestions. Research is partially supported by the National Science Foundation (NSF) Grants 2023239-DMS, CCF-2046710, and Air Force Grant FA9550-18-1-0166.

References

Raghav Addanki, Chenyang Li, Zhao Song, and Chiwun Yang. 2023. One pass streaming algorithm for super long token attention approximation in sublinear space. *arXiv preprint arXiv:2311.14652*.

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, and 1 others. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

Sina Alemohammad, Zichao Wang, Randall Balestriero, and Richard Baraniuk. 2020. The recurrent neural tangent kernel. *arXiv preprint arXiv:2006.10246*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023a. Physics of language models: Part 1, context-free grammar. *arXiv preprint arXiv:2305.13673*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023b. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023c. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. 2019. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR.

Josh Alman and Zhao Song. 2023. Fast attention requires bounded entries. *Advances in Neural Information Processing Systems*, 36.

Josh Alman and Zhao Song. 2024a. The fine-grained complexity of gradient computation for training large language models. *arXiv preprint arXiv:2402.04497*.

Josh Alman and Zhao Song. 2024b. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *The Twelfth International Conference on Learning Representations*.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. 2019. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. 2022. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

Sergei Bernstein. 1924. On a modification of chebyshev's inequality and of the error formula of laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math*, 1(4):38–49.

Alberto Bietti and Julien Mairal. 2019. On the inductive bias of neural tangent kernels. *Advances in Neural Information Processing Systems*, 32.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

ChatGPT. 2022. Optimizing language models for dialogue. *OpenAI Blog*.

Herman Chernoff. 1952. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507.

Lenaic Chizat and Francis Bach. 2020. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on learning theory*, pages 1305–1338. PMLR.

Timothy Chu, Zhao Song, and Chiwun Yang. 2023. Fine-tune language models to approximate unbiased in-context learning. *arXiv preprint arXiv:2310.03331*.

Timothy Chu, Zhao Song, and Chiwun Yang. 2024. How to protect copyright data in optimization of large language models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17871–17879.

Claude-3. 2024. Introducing the next generation of claude. *Anthropic News*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Yichuan Deng, Zhihang Li, and Zhao Song. 2023. Attention scheme inspired softmax regression. *arXiv preprint arXiv:2304.10411*.

Yichuan Deng, Zhao Song, Jing Xiong, and Chiwun Yang. 2024a. How sparse attention approximates exact attention? your attention is naturally n^C-sparse. *arXiv preprint arXiv:2404.02690*.

Yichuan Deng, Zhao Song, and Chiwun Yang. 2024b. Attention is naturally sparse with gaussian distributed input. *arXiv preprint arXiv:2404.02690*.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. 2019. Gradient descent provably optimizes over-parameterized neural networks. In *ICLR*. arXiv preprint arXiv:1810.02054.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Sergey Foss, Dmitry Korshunov, Stan Zachary, and 1 others. 2011. *An introduction to heavy-tailed and subexponential distributions*, volume 6. Springer.

Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas Dengel. 2021. Adversarial text-to-image synthesis: A review. *Neural Networks*, 144:187–209.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Yeqi Gao, Sridhar Mahadevan, and Zhao Song. 2023a. An over-parameterized exponential regression. *arXiv preprint arXiv:2303.16504*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2023. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv preprint arXiv:2310.01801*.

Gemini. 2024. Welcome to the gemini era. *Google Deepmind Technologies*.

Uffe Haagerup. 1981. The best constants in the khintchine inequality. *Studia Mathematica*, 70(3):231–283.

Insu Han, Rajesh Jayaram, Amin Karbasi, Vahab Mirrokni, David Woodruff, and Amir Zandieh. 2024. Hyperattention: Long-context attention in near-linear time. In *The Twelfth International Conference on Learning Representations*.

David Lee Hanson and Farroll Tim Wright. 1971. A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426.

Connor Holmes, Masahiro Tanaka, Michael Wyatt, Ammar Ahmad Awan, Jeff Rasley, Samyam Rajbhandari, Reza Yazdani Aminabadi, Heyang Qin, Arash Bakhtiari, Lev Kurilenko, and 1 others. 2024. Deepspeed-fastgen: High-throughput text generation for llms via mii and deepspeed-inference. *arXiv preprint arXiv:2401.08671*.

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. 2020. Infinite attention: Nngp and ntk for deep attention networks. In *International Conference on Machine Learning*, pages 4376–4386. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jerry Yao-Chieh Hu, Maojiang Su, En-Jui Kuo, Zhao Song, and Han Liu. 2024. Computational limits of low-rank adaptation (lora) for transformer-based models. *arXiv preprint arXiv:2406.03136*.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H Abdi, Dongsheng Li, Chin-Yew Lin, and 1 others. 2024. Minference 1.0: Accelerating prefilling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems*, 37:52481–52515.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Praneeth Kacham, Vahab Mirrokni, and Peilin Zhong. 2023. Polysketchformer: Fast transformers via sketches for polynomial kernels. *arXiv preprint arXiv:2310.01655*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Aleksandr Khintchine. 1923. Über dyadische brüche. *Mathematische Zeitschrift*, 18(1):109–116.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alex Krizhevsky, Geoffrey Hinton, and 1 others. 2009. Learning multiple layers of features from tiny images.

Beatrice Laurent and Pascal Massart. 2000. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics*, pages 1302–1338.

Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. 2020. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chenyang Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. 2024a. Exploring the frontiers of softmax: Provable optimization, applications in diffusion model, and beyond. *arXiv preprint arXiv:2405.03251*.

Chenyang Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Tianyi Zhou. 2024b. Fourier circuits in neural networks: Unlocking the potential of large language models in mathematical reasoning and modular arithmetic. *arXiv preprint arXiv:2402.09469*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Yuanzhi Li and Yingyu Liang. 2018. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024c. Chain of thought empowers transformers to solve inherently serial problems. *arXiv preprint arXiv:2402.12875*.

Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, and Junze Yin. 2024a. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. *arXiv preprint arXiv:2405.05219*.

Yingyu Liang, Zhenmei Shi, Zhao Song, and Yufa Zhou. 2024b. Tensor attention training: Provably efficient learning of higher-order transformers. *arXiv preprint arXiv:2405.16411*.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, and 1 others. 2024. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *arXiv preprint arXiv:2409.10516*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pretrain, prompt, and predict: a systematic survey of prompting methods in natural language processing. arxiv. *arXiv preprint arXiv:2107.13586*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. 2013. Faster ridge regression via the subsampled randomized hadamard transform. *Advances in neural information processing systems*, 26.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. 2023. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pages 23610–23641. PMLR.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Mohammed Ali mnmoustafa. 2017. Tiny imagenet.

Alexander Munteanu, Simon Omlor, Zhao Song, and David Woodruff. 2022. Bounding the width of neural networks via coupled initialization a worst case analysis. In *International Conference on Machine Learning*, pages 16083–16122. PMLR.

Druv Pai, Sam Buchanan, Ziyang Wu, Yaodong Yu, and Yi Ma. 2024. Masked completion via structured diffusion with white-box transformers. In *The Twelfth International Conference on Learning Representations*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Aleksandar Petrov, Philip Torr, and Adel Bibi. 2024a. When do prompting and prefix-tuning work? a theory of capabilities and limitations. In *The Twelfth International Conference on Learning Representations*.

Aleksandar Petrov, Philip HS Torr, and Adel Bibi. 2024b. Prompting a pretrained transformer can be a universal approximator. *arXiv preprint arXiv:2402.14753*.

Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. 2019. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1505–1514.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Mark Rudelson and Roman Vershynin. 2013. Hanson-wright inequality and sub-gaussian concentration.

Mariia Seleznova and Gitta Kutyniok. 2022. Neural tangent kernel beyond the infinite-width limit: Effects of depth and initialization. In *International Conference on Machine Learning*, pages 19522–19560. PMLR.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. 2024. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608*.

Zhenmei Shi, Jiefeng Chen, Kunyang Li, Jayaram Raghuram, Xi Wu, Yingyu Liang, and Somesh Jha. 2022. The trade-off between universality and label efficiency of representations from contrastive learning. In *The Eleventh International Conference on Learning Representations*.

Zhenmei Shi, Yifei Ming, Xuan-Phi Nguyen, Yingyu Liang, and Shafiq Joty. 2024a. Discovering the gems in early layers: Accelerating long-context llms with 1000x input token reduction. *arXiv preprint arXiv:2409.17422*.

Zhenmei Shi, Junyi Wei, and Yingyu Liang. 2021. A theoretical analysis on feature learning in neural networks: Emergence from inputs and advantage over fixed features. In *International Conference on Learning Representations*.

Zhenmei Shi, Junyi Wei, and Yingyu Liang. 2024b. Provable guarantees for neural networks via gradient feature learning. *Advances in Neural Information Processing Systems*, 36.

Kihyuk Sohn, Huiwen Chang, José Lezama, Luisa Polania, Han Zhang, Yuan Hao, Irfan Essa, and Lu Jiang. 2023. Visual prompt tuning for generative transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19840–19851.

Zhao Song and Xin Yang. 2019. Quadratic suffices for over-parametrization via matrix chernoff bound. *arXiv preprint arXiv:1906.03593*.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Joel A Tropp. 2011. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, and Neel Joshi. 2024. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *arXiv preprint arXiv:2406.14852*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yihan Wang, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. 2023a. Universality and limitations of prompt tuning. *Advances in Neural Information Processing Systems*, 36.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. 2023b. Multitask prompt tuning enables parameter-efficient transfer learning. *arXiv preprint arXiv:2303.02861*.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022b. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. 2019. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Junda Wu, Tong Yu, Rui Wang, Zhao Song, Ruiyi Zhang, Handong Zhao, Chaochao Lu, Shuai Li, and Ricardo Henao. 2023. Infoprompt: Information-theoretic soft prompt tuning for natural language understanding. *Advances in Neural Information Processing Systems*, 36.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2024. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Jing Xiong, Chengming Li, Min Yang, Xiping Hu, and Bin Hu. 2022. Expression syntax information bottleneck for math word problems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2166–2171.

Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng Yang, Qingxing Cao, Haiming Wang, Xiongwei Han, and 1 others. 2023a. Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. *arXiv preprint arXiv:2310.02954*.

Jing Xiong, Jianghan Shen, Fanghua Ye, Chaofan Tao, Zhongwei Wan, Jianqiao Lu, Xun Wu, Chuanyang Zheng, Zhijiang Guo, Lingpeng Kong, and 1 others. 2024. Uncomp: Uncertainty-aware long-context compressor for efficient large language model inference. *arXiv preprint arXiv:2410.03090*.

Jing Xiong, Jianghan Shen, Chuanyang Zheng, Zhongwei Wan, Chenyang Zhao, Chiwun Yang, Fanghua Ye, Hongxia Yang, Lingpeng Kong, and Ngai Wong. 2025. Parallelcomp: Parallel long-context compressor for length extrapolation. *arXiv preprint arXiv:2502.14317*.

Jing Xiong, Jianhao Shen, Ye Yuan, Haiming Wang, Yichun Yin, Zhengying Liu, Lin Li, Zhijiang Guo, Qingxing Cao, Yinya Huang, and 1 others. 2023b. Trigo: Benchmarking formal mathematical proof reduction for generative language models. *arXiv preprint arXiv:2310.10180*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023a. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin Haeffele, and Yi Ma. 2023b. White-box transformers via sparse rate reduction. *Advances in Neural Information Processing Systems*, 36:9422–9457.

Yaodong Yu, Tianzhe Chu, Shengbang Tong, Ziyang Wu, Druv Pai, Sam Buchanan, and Yi Ma. 2023c. Emergence of segmentation with minimalistic white-box transformers. *arXiv preprint arXiv:2308.16271*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, and 1 others. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Yuchen Zeng and Kangwook Lee. 2023. The expressive power of low-rank adaptation. *arXiv preprint arXiv:2310.17513*.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023a. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909*.

Jintao Zhang, Jia Wei, Pengle Zhang, Xiaoming Xu, Haofeng Huang, Haoxu Wang, Kai Jiang, Jun Zhu, and Jianfei Chen. 2025. Sageattention3: Microscaling fp4 attention for inference and an exploration of 8-bit training. *arXiv preprint arXiv:2505.11594*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023b. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825.

Mo Zhou, Rong Ge, and Chi Jin. 2021. A local convergence theory for mildly over-parameterized two-layer neural network. In *Conference on Learning Theory*, pages 4577–4632. PMLR.

Appendix

A More Related Works

Efficient Transformers and KV Cache. The design of the transformer-based architecture is computationally costly in long context situations due to its quadratic complexity. This has long been an active area of study (Xiong et al., 2024; Addanki et al., 2023; Deng et al., 2024a; Xiong et al., 2025; Xiao et al., 2024; Liu et al., 2024; Han et al., 2024). BigBird (Zaheer et al., 2020) effectively reduces attention complexity through a hybrid approach that integrates local, global, and random attention mechanisms. Subsequent advancements, such as StreamingLLM (Xiao et al., 2023) and H2O (Zhang et al., 2023b), streamline attention patterns by strategically discarding KV caches during context processing. However, a notable limitation of these methods is their diminished capacity to preserve the long-context capabilities inherent in the original models, primarily due to insufficient global context modeling. More recent innovations include SeerAttention (Zhang et al., 2025), which employs a learnable gate to pinpoint block-level attention sparsity. Concurrently, Quest (Tang et al., 2024) introduces dynamic, query-aware sparsity to accelerate decoding, while MInference (Jiang et al., 2024) extends these principles to the prefilling phase. Furthermore, FastGen (Ge et al., 2023; Holmes et al., 2024) enhances decoding efficiency by profiling attention heads to selectively discard tokens.

B Algorithm Details and Computational Complexity Analysis

Here, we give the detailed version of two algorithms of this paper, which are prefix attention and NTK-Attention. Moreover, we comment on each computation step with its corresponding complexity to demonstrate our memory and complexity reduction in detail.

From Algorithm 3 and Algorithm 4, we can see the comparison analysis of memory reduction (from $O(md)$ to $O(rd + r)$) and complexity reduction (from $O(mL + L^2)$ to $O(Ld + L^2)$ since $m \gg L$ and $m \gg d$) between two fine-tuning methods, indicating the efficiency of our NTK-Attention.

Algorithm 3 Prefix Attention (Detailed version of Algorithm 1)

Input: Input matrix $X \in \mathbb{R}^{L \times d}$

Parameters: Frozen query, key and value weights $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$, trainable prefix matrix $P \in \mathbb{R}^{m \times d}$ ▷ Additional memory usage $O(md)$

Output: Exact output $\mathsf{Attn} \in \mathbb{R}^{L \times d}$

1: **procedure** PREFIXATTENTION(X)

2: Concatenate input matrix with prefix matrix $S \leftarrow \begin{bmatrix} P \\ X \end{bmatrix} \in \mathbb{R}^{(m+L) \times d}$

3: Compute query, key, and value matrices for attention $Q \leftarrow XW_Q \in \mathbb{R}^{L \times d}$, $K_P \leftarrow SW_K \in \mathbb{R}^{(m+L) \times d}$, $V_P \leftarrow SW_V \in \mathbb{R}^{(m+L) \times d}$ ▷ Time complexity $O(Ld^2 + 2(m+L)d^2)$

4: Compute exponential matrix $A \leftarrow \exp(QK_P^\top / \sqrt{d}) \in \mathbb{R}^{L \times (m+L)}$ ▷ Time complexity $O(L(m+L)d)$

5: Compute summation of exponential matrix $D \leftarrow \mathrm{diag}(A \mathbf{1}_{m+L}) \in \mathbb{R}^{L \times L}$ ▷ Time complexity $O(L(m+L))$

6: Compute prefix attention output $\mathsf{Attn} \leftarrow D^{-1} A V_P \in \mathbb{R}^{L \times d}$ ▷ Here $D^{-1}A \in \mathbb{R}^{L \times (m+L)}$ is the attention matrix (a.k.a attention scores). This step implements A multiply V_P first, then get $D^{-1} \cdot (AV_P)$ with time complexity $O(L(m+L)d + L^2 d)$

7: **return** Attn

8: **end procedure**

C Experimental Details

C.1 Setup Details

Here, we give the details of the setup for the experiments in Section 6.

Algorithm 4 NTK-Attention (Detailed version of Algorithm 2, w low-rank)

Input: Input matrix $X \in \mathbb{R}^{L \times d}$

Parameters: Frozen query, key and value weights $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$, trainable weights $Z_A \in \mathbb{R}^{r \times s}$, $Z_B \in \mathbb{R}^{s \times d}$ and $k \in \mathbb{R}^r$ ▷ Additional memory usage $O(rs + sd + r)$

Output: Approximating output $T \in \mathbb{R}^{L \times d}$

procedure NTK-ATTENTION(X)

Compute query, key, and value matrices for attention $Q \leftarrow XW_Q \in \mathbb{R}^{L \times d}$, $K \leftarrow XW_K \in \mathbb{R}^{L \times d}$, $V \leftarrow XW_V \in \mathbb{R}^{L \times d}$ ▷ Time complexity $O(3Ld^2)$

Compute approximating exponential matrix $\widehat{A} \leftarrow \exp(QK^\top/\sqrt{d}) \in \mathbb{R}^{L \times L}$ ▷ Time complexity $O(L^2 d)$

Compute approximating summation of exponential matrix $\widehat{D} \leftarrow \mathrm{diag}(\widehat{A}\mathbf{1}_L + \Phi(Q)k) \in \mathbb{R}^{L \times L}$ ▷ Time complexity $O(L^2 + Lr)$

Compute approximation of prefix attention output $T \leftarrow \widehat{D}^{-1}(\widehat{A}V + \Phi(Q)Z_A \cdot Z_B) \in \mathbb{R}^{L \times d}$ ▷ This step implements $Z := Z_A \cdot Z_B$ first, compute $\widehat{A}V + \Phi(Q)Z$ secondly, then implements $\widehat{D}^{-1} \cdot (\widehat{A}V + \Phi(Q)Z_A \cdot Z_B)$, time complexity $O(2L^2 d + Lr^2 + rsd)$

return T

end procedure

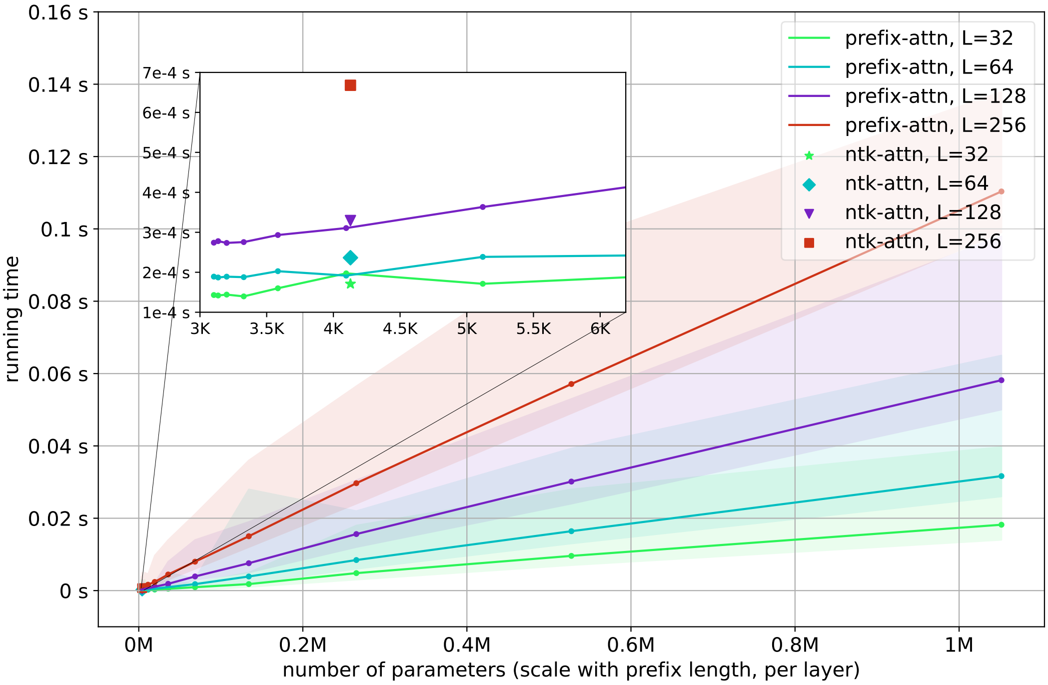
- Learning rate $\eta = 0.001$ (default).
- Learning rate scheduler: Cosine.
- Adam hyper-parameter $\beta_1 = 0.9$ (default).
- Adam hyper-parameter $\beta_2 = 0.999$ (default).
- Adam hyper-parameter $\epsilon = 1 \times 10^{-8}$ (default).
- Platform: PyTorch (Paszke et al., 2019) and Huggingface (Wolf et al., 2019).
- GPU device information: 8 V100 GPUs, 8 4090 GPUs and 4 H800 GPUs.
- Number of training epochs 30.
- Batch size for vision tasks: 256 (for best effort).
- Batch size for natural language task: 32 (for best effort).
- Max input length for natural language task: 128 for each feature, e.g. BoolQ has two dataset features: question and passage, for each data, we select the first 128 tokens in question and passage of the data respectively, and concatenate them to be the input. Note that this is different from (Liu et al., 2021b).
- Quantization: fp16 and bf16.

C.2 Additional Empirical Complexity Analysis

We state an additional empirical complexity analysis here to support our claim practically. We evaluate the complexity reduction on one layer to show how much efficiency our NTK-Attention will demonstrate per layer.

Setup. Firstly, we choose $d = 32$ and $r = d$, and randomly initialize attention weights $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$. For the trainable parameters in NTK-Attention and Prefix Attention, we initialize $P \in \mathbb{R}^{m \times d}$, $Z \in \mathbb{R}^{d \times d}$ and $k \in \mathbb{R}^d$ randomly, either. We then scale the prefix length, denotes m, within the range $\{2^0, 2^1, \cdots, 2^{16}\}$ for comparison. The input length L is chosen from $\{32, 64, 128, 256\}$. For computation, we initialize a new input matrix $X \in \mathbb{R}^{L \times d}$ and compute NTK-Attention and Prefix Attention respectively. We repeat each computation with a different setup 10000 times and record the maximum, minimum, and

Figure 3: Run time and the number of parameters of one-layer NTK-Attention and Prefix Attention (on random input data). x-axis: the number of parameters; y-axis: run time. Input length L is chosen from $\{32, 64, 128, 256\}$, dimension $d = 32$ and prefix length m is chosen from $\{2^0, 2^1, \cdots, 2^{16}\}$.



mean values. The inference is run on an AMD CPU to compare FLOPS fairly between two algorithms (this also works on GPU devices).

Results. We demonstrate our result in Figure 3. The x-axis is the number of parameters (representing memory usage), and the y-axis shows the run time in seconds. Note that the number of parameters is computed by the summation of every number in NTK-Attention or Prefix Attention. For example, $m = 1024$, $d = 32$, the number of parameters of Prefix Attention is $md + 3d^2 = 35840$; the number of parameters if NTK-Attention is $4d^2 + d = 4128$.

As expected, the number of parameters of Prefix Attention increases linearly with the prefix length m, and its running time increases quadratically with m. While our method, NTK-Attention, has computational costs unaffected by the prefix length. It maintains a small running time and low memory usage as shown in the figure. Roughly speaking, the cost of NTK-Attention is close to Prefix Attention with a very small prefix length $m = 32$.

C.3 Additional Ablation Study

Setup. We provide an additional ablation study for the sensitivity of the hyper-parameters of NTK-Attention r and s here and the results are given in Table 4. In particular, this experiment is run on pretrained LLAMA-3.1-8B-Instruct model ($d = 128$ for each head in attention) (Touvron et al., 2023a,b) with dataset WikiText-103 (Merity et al., 2016). We utilize 4 H800 GPU devices to train the model with different settings within 2 epochs on the training dataset and evaluate them on the test dataset. The metric is cross-entropy loss and its smaller value stands for better performance.

Results. We show the NTK-Attention with the weakest setting $r = 128, s = 4$ is able to achieve competitive performance with $r = 256, r = 64$. This further ensures the parameter efficiency of NTK-Attention.

Moreover, Table 4 also demonstrates that choosing a big value for hyper-parameter r primarily will lead to better evaluation loss since NTK-Attention with $(r, s) = (256, 32)$ requires 12.85M parameters but achieve superior performance compared to NTK-Attention with $(r, s) = (128, 64)$ (requires 16.91M parameters).

However, we discover that an increased value for r might cause huge complexity - when setting

$r = 512$, the computational complexity $4Ld$ will lead the GPU out-of-memory (OOM) since it's usually unaffordable even for H800 (80GiB memory). Thus, we also suggest using $r = d$ or $r = 2d$ to make LLMs to learn downstream tasks.

Table 4: The results of ablation study to the NTK-Attention hyper-parameters r and s with pretrained LLM LLAMA-3.1-8B-Instruct and dataset WikiText-103 on H800 GPUs (80GiB).

Hyper-parameters	Num Parameters	Evaluation Loss	Training Loss
(r, s)=(128, 4)	1.18M	2.48	2.38
(r, s)=(128, 8)	2.23M	2.57	2.50
(r, s)=(128, 16)	4.33M	2.74	2.72
(r, s)=(128, 32)	8.52M	2.47	2.38
(r, s)=(128, 64)	16.91M	2.41	2.31
(r, s)=(256, 4)	1.84M	2.47	2.39
(r, s)=(256, 8)	3.41M	2.43	2.36
(r, s)=(256, 16)	6.55M	2.51	2.53
(r, s)=(256, 32)	12.85M	2.28	2.33
(r, s)=(256, 64)	25.43M	2.21	2.15
(r, s)=(512, 4)	3.15M (OOM since $4Ld$ complexity)	-	-

D Naive NTK-Attention Implementation with Flash-Attention

Below, we provide a naive Python code to implement our NTK-Attention that is written in only 10 lines, which supports the simplicity of implementation. Our code utilizes the function of Flash Attention function (Dao et al., 2022; Dao, 2023; Shah et al., 2024).

```python
def ntk_attn_forward(self, query_states, key_states, value_states,
    attention_mask):
    attn_outputs, lse = _flash_attention_forward(
        query_states, key_states, value_states, attention_mask,
        is_causal=self.is_causal, return_attn_probs=True
    ) # Call flash-attn function to get attn_output and logsumexp

    Z = torch.matmul(self.Z_A, self.Z_B) # Low-rank approximate Z
    k = self.k
    phi_query_states = self.phi(query_states)

    se = lse.exp() # Compute sumexp
    scale_factor = (se + torch.matmul(phi_query_states, k)) / se

    attn_output = scale_factor * (attn_output * se + torch.matmul(
    phi_query_states, Z))

    return attn_output
```

Additionally, we also provide the code for the function ϕ, which is implemented by simple Taylor expansions (Alman and Song, 2023).

E Further Discussions

Prior works (Arora et al., 2019; Alemohammad et al., 2020; Hron et al., 2020) had already given exact algorithms for computing the extension of NTK to neural nets and conducted experiments showing enhanced performance from adding NTK into models, while in this paper, our contributions are not limited to this. Our theory about NTK of attention with the infinite-long prefix provides more insights. We clarify this further in the following.

```python
def phi(x, deg=2):
    """x is the input tensor, and deg is the number of degree"""
    ret = [torch.ones_like(x[..., -1:]), ]

    pre_shape, d = x.shape[:-1], x.shape[-1]
    x = x.unsqueeze(-2) / math.pow(d, 0.25)
    factor = 1
    for idx in range(1, deg+1):
        ret.append(
            torch.matmul(ret[-1].unsqueeze(-1), x).view(*pre_shape, -1) / math.sqrt(factor)
        )
        factor *= (idx + 1)
    ret = torch.cat(ret, dim=-1)
    return ret
```

Can LLMs master any advanced reasoning skill through self-planning and prompting? We will answer that it may be possible. Since an attention network can converge on any dataset with the infinite-long prefix, we can tell that for any advanced reasoning skill that is equivalent to training on a well-constructed dataset, there exists an ultra-long prefix matrix satisfying the training objective smaller than any positive value $\epsilon > 0$. It's noteworthy that this conclusion is not only suitable for LLMs with outstanding performance but also can be worked on those small language models with common performance.

What is NTK-Attention used for? What is the meaning of proposing this method? The attention with an infinite-long prefix is superior due to its over-parameterization phenomenon, whereas it is nearly impossible to implement practically, our NTK-Attention method gives us a chance to approximate the infinite-long prefix and makes it possible for us to study its empirical properties in experiments. Besides, any form of prefix learning can be formulated into the training of $Z \in \mathbb{R}^{d\times d}$ and $k \in \mathbb{R}^d$ in NTK-Attention, we can compress prompts into Z and k if $\phi(\cdot)$ by utilizing Lemma L.7, hence, the approaches in Prefix Learning would be much more efficient.

Comparison between NTK-Attention and LoRA. LoRA in (Hu et al., 2021; Zeng and Lee, 2023; Hu et al., 2024) is a popular efficient fine-tuning method for large base models. Usually, LoRA makes adaptation on Query and Value projections $W_Q, W_V \in \mathbb{R}^{d\times d}$; denote the adaptation as $W_{\Delta Q}, W_{\Delta V} \in \mathbb{R}^{d\times d}$. Given an input $X \in \mathbb{R}^{L\times d}$, LoRA computes $\widetilde{D}^{-1}\widetilde{A}X(W_V + W_{\Delta V})$, where $\widetilde{A} := \exp(X(W_Q + W_{\Delta Q})W_K^\top X^\top)$, $\widetilde{D} := \mathrm{diag}(\widetilde{A}\mathbf{1}_L)$, and $W_K \in \mathbb{R}^{d\times d}$ is the Key projection weights. So LoRA updates query and value weights during training, while our NTK-Attention compresses the additional prefix P into Z and k (Algorithm 2), which is a completely different mechanism. Our method also achieves comparable performance to LoRA in our experiments in Section 6. Also, note that the two methods are orthogonal to each other and can be used together.

Connection to the newest SOTA LLM on math inference tasks, GPT-o1 [1]. On September 12-th, 2024, OpenAI released the newest SOTA LLM on math inference tasks, GPT-o1, which is trained by Reinforcement Learning (RL) methods to enhance the Chain-of-Thought (CoT) ability. Li et al. (2024c) explained the necessity of CoT for LLM on complicated inference tasks, meanwhile, they also emphasized how the embedding size and the CoT length affect the capability to solve high-order problems. Connecting to our work, we believe that these empirical and theoretical results support the conclusion of our work since we consider CoT as a specific application of Prefix Learning. Moreover, we think our *scaling law in prefix learning* is more universal for explaining the LLMs' context-based advanced skills. However, even when we present our theory, we still have a limited understanding of prefix learning, for example, what is the relationship between prefix length and complexity of problems that aim to solve; if we want to solve an NP problem by LLM, how long is the prefix needed for inference? We don't know the answers. Thus, explaining prefix learning, or particularly, CoT, is still a fascinating and challenging problem for future work.

[1] https://openai.com/o1/

Societal impact. This paper presents work whose goal is to advance the understanding of context-based fine-tuning methods (prefix learning) theoretically. There are many positive potential societal consequences of our work, such as inspiring new algorithm design. Since our work is theoretical in nature, we do not foresee any potential negative societal impacts which worth pointing out.

F Preliminary of Analysis

We provide our notations for this paper as follows:

Notations In this paper, we use integer d to denote the dimension of networks. We use integer m to denote the prefix length in prefix learning, we think m is an ultra-big number. We use L to denote the input length in language models. $\nabla_x f(x)$ and $\frac{\mathrm{d}f(x)}{\mathrm{d}x}$ are both means to take the derivative of $f(x)$ with x. Let a vector $z \in \mathbb{R}^n$. We denote the ℓ_2 norm as $\|z\|_2 := (\sum_{i=1}^n z_i^2)^{1/2}$, the ℓ_1 norm as $\|z\|_1 := \sum_{i=1}^n |z_i|$, $\|z\|_0$ as the number of non-zero entries in z, $\|z\|_\infty$ as $\max_{i\in[n]} |z_i|$. We use $z^\top$ to denote the transpose of a z. We use $\langle \cdot, \cdot \rangle$ to denote the inner product. Let $A \in \mathbb{R}^{n\times d}$, we use $\mathrm{vec}(A)$ to denote a length nd vector. We denote the Frobenius norm as $\|A\|_F := (\sum_{i\in[n],j\in[d]} A_{i,j}^2)^{1/2}$. For any positive integer n, we use $[n]$ to denote set $\{1, 2, \cdots, n\}$. We use $\mathbb{E}[]$ to denote the expectation. We use $\Pr[]$ to denote the probability. We use ϵ to denote the error. We define $\lambda_{\min}(\cdot)$ as a function that outputs the minimum eigenvalues of the input matrix, e.g. matrix $A \in \mathbb{R}^{n\times n}$ has eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$, $\lambda_{\min}(A) = \min\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$.

F.1 Facts

Fact F.1. *For any $x \in (-0.01, 0.01)$, we have*

$$\exp(x) = 1 + x + \Theta(1)x^2.$$

Fact F.2. *For any $x \in (0, 0.1)$, we have*

$$\sum_{i=1}^n x^i \leq \frac{1}{1-x}.$$

F.2 Probability

Here, we state a probability toolkit in the following, including several helpful lemmas we'd like to use. Firstly, we provide the lemma about Chernoff bound in (Chernoff, 1952) below.

Lemma F.3 (Chernoff bound, (Chernoff, 1952))**.** *Let $X = \sum_{i=1}^n X_i$, where $X_i = 1$ with probability p_i and $X_i = 0$ with probability $1 - p_i$, and all X_i are independent. Let $\mu = \mathbb{E}[X] = \sum_{i=1}^n p_i$. Then*

- $\Pr[X \geq (1+\delta)\mu] \leq \exp(-\delta^2\mu/3)$, $\forall \delta > 0$;
- $\Pr[X \leq (1-\delta)\mu] \leq \exp(-\delta^2\mu/1)$, $\forall 0 < \delta < 1$.

Next, we offer the lemma about Hoeffding bound as in (Hoeffding, 1994).

Lemma F.4 (Hoeffding bound, (Hoeffding, 1994))**.** *Let $X_1, \cdots, X_n$ denote n independent bounded variables in $[a_i, b_i]$ for $a_i, b_i \in \mathbb{R}$. Let $X := \sum_{i=1}^n X_i$, then we have*

$$\Pr[|X - \mathbb{E}[X]| \geq t] \leq 2\exp(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2})$$

We show the lemma of Bernstein inequality as (Bernstein, 1924).

Lemma F.5 (Bernstein inequality, (Bernstein, 1924))**.** *Let $X_1, \cdots, X_n$ denote n independent zero-mean random variables. Suppose $|X_i| \leq M$ almost surely for all i. Then, for all positive t,*

$$\Pr[\sum_{i=1}^n X_i \geq t] \leq \exp(-\frac{t^2/2}{\sum_{j=1}^n \mathbb{E}[X_j^2] + Mt/3})$$

Then, we give the Khintchine's inequality in (Khintchine, 1923; Haagerup, 1981) as follows:

Lemma F.6 (Khintchine's inequality, (Khintchine, 1923; Haagerup, 1981))**.** *Let $\sigma_1, \cdots, \sigma_n$ be i.i.d sign random variables, and let $z_1 \cdots, z_n$ be real numbers. Then there are constants $C > 0$ so that for all $t > 0$*

$$\Pr[|\sum_{i=1}^{n} z_i \sigma_i| \geq t\|z\|_2] \leq \exp(-Ct^2).$$

We give Hason-wright inequality from (Hanson and Wright, 1971; Rudelson and Vershynin, 2013) below.

Lemma F.7 (Hason-wright inequality, (Hanson and Wright, 1971; Rudelson and Vershynin, 2013))**.** *Let $x \in \mathbb{R}^n$ denote a random vector with independent entries x_i with $\mathbb{E}[x_i] = 0$ and $|x_i| \leq K$ Let A be an $n \times n$ matrix. Then, for every $t \geq 0$*

$$\Pr[|x^\top Ax - \mathbb{E}[x^\top Ax]| > t]$$
$$\leq 2\exp(-c\min\{t^2/(K^4\|A\|_F^2), t/(K^2\|A\|)\}).$$

We state Lemma 1 on page 1325 of Laurent and Massart (Laurent and Massart, 2000).

Lemma F.8 (Lemma 1 on page 1325 of Laurent and Massart, (Laurent and Massart, 2000))**.** *Let $X \sim \mathcal{X}_k^2$ be a chi-squared distributed random variable with k degrees of freedom. Each one has zero mean and σ^2 variance. Then*

$$\Pr[X - k\sigma^2 \geq (2\sqrt{kt} + 2t)\sigma^2] \leq \exp(-t)$$
$$\Pr[X - k\sigma^2 \geq 2\sqrt{kt}\sigma^2] \leq \exp(-t).$$

Here, we provide a tail bound for sub-exponential distribution (Foss et al., 2011).

Lemma F.9 (Tail bound for sub-exponential distribution, (Foss et al., 2011))**.** *We say $X \in \mathrm{SE}(\sigma^2, \alpha)$ with parameters $\sigma > 0$, $\alpha > 0$, if*

$$\mathbb{E}[e^{\lambda X}] \leq \exp(\lambda^2\sigma^2/2), \forall|\lambda| < 1/\alpha.$$

Let $X \in \mathrm{SE}(\sigma^2, \alpha)$ and $\mathbb{E}[X] = \mu$, then:

$$\Pr[|X - \mu| \geq t] \leq \exp(-0.5\min\{t^2/\sigma^2, t/\alpha\}).$$

In the following, we show the helpful lemma of matrix Chernoff bound as in (Tropp, 2011; Lu et al., 2013).

Lemma F.10 (Matrix Chernoff bound, (Tropp, 2011; Lu et al., 2013))**.** *Let $\mathcal{X}$ be a finite set of positive-semidefinite matrices with dimension $d \times d$, and suppose that*

$$\max_{X \in \mathcal{X}} \lambda_{\max}(X) \leq B.$$

Sample $\{X_1, \cdots, X_n\}$ uniformly at random from $\mathcal{X}$ without replacement. We define $\mu_{\min}$ and $\mu_{\max}$ as follows:

$$\mu_{\min} := n \cdot \lambda_{\min}(\mathop{\mathbb{E}}_{X \in \mathcal{X}}(X))$$
$$\mu_{\max} := n \cdot \lambda_{\max}(\mathop{\mathbb{E}}_{X \in \mathcal{X}}(X)).$$

Then

$$\Pr[\lambda_{\min}(\sum_{i=1}^{n} X_i) \leq (1 - \delta)\mu_{\min}]$$
$$\leq d \cdot \exp(-\delta^2\mu_{\min}/B) \text{ for } \delta \in (0, 1],$$
$$\Pr[\lambda_{\max}(\sum_{i=1}^{n} X_i) \geq (1 + \delta)\mu_{\max}]$$
$$\leq d \cdot \exp(-\delta^2\mu_{\max}/(4B)) \text{ for } \delta \geq 0.$$

G Definitions of NTK Analysis

This section provides the fundamental definitions of our NTK analysis in this paper.

To begin with, we re-denote our weight of prefix in attention as $W \in \mathbb{R}^{d \times m}$ and $a \in \{-1, +1\}^m$ as follows[2]:

Definition G.1. *We choose $a \in \{-1, +1\}^m$ to be weights that each entry a_r is randomly sampled from -1 with probability $1/2$ and $+1$ with probability $1/2$.*

Let $W \in \mathbb{R}^{d \times m}$ denote random Gaussian weights, i.e., each entry independently draws from $\mathcal{N}(0, \sigma^2)$. For each $r \in [m]$, we use $w_r \in \mathbb{R}^d$ to denote the r-th column of W.

Equivalence. The attention computation with prefix P. Since the attention parameters are fixed, it can be rewritten as $\mathsf{Softmax}(\widetilde{X} P^\top + b) \cdot \begin{bmatrix} PW_V \\ b' \end{bmatrix}$ where $\widetilde{X} = XW_Q W_K^\top / \sqrt{d}, b = XW_Q W_K^\top X^\top / \sqrt{d}$, and $b' = XW_V$. We view the input sequence as one token (i.e., assuming $L = 1$) such that the input X and thus $\widetilde{X}$ become vectors, simplifying our analysis from matrix-form calculations to vector-form. Furthermore, ignoring the bias terms, and introducing notations $x := \widetilde{X}^\top$ and $W = P^\top$, the attention simplifies to $\mathsf{Softmax}(xW) \cdot W^\top W_V = \frac{\sum_{r \in [m]} \exp(w_r^\top x) w_r W_V}{\sum_{r \in [m]} \exp(w_r^\top x)}$ where w_r is the r-th column of W. We therefore consider the following two-layer attention model:

$$\mathsf{F}(W, x, a) := m \frac{\sum_{r \in [m]} \exp(w_r^\top x) w_r a_r}{\sum_{r \in [m]} \exp(w_r^\top x)} \tag{5}$$

with the hidden-layer weights $W = [w_1, w_2, \dots, w_m] \in \mathbb{R}^{d \times m}$ and output-layer weights $a = [a_1, a_2, \dots, a_m]^\top \in \mathbb{R}^m$. Such a stylized setting has been widely used for studying the learning behavior of transformer-based models (Deng et al., 2023; Chu et al., 2023, 2024; Li et al., 2024a), and they gave detailed derivations and guarantees for its connection to attention. Furthermore, our analysis can be extended to models with bias terms and matrix inputs rigorously.

Since we have established the equivalence between the ultra-long prefix matrix in attention and our theory above, it's reasonable we utilize the following definition of F to decompose the model function and facilitate our analysis. As we give the formal definitions as follows:

Definition G.2. *We define function $\mathsf{F} : \mathbb{R}^{d \times m} \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$*

$$\mathsf{F}(W, x, a) = m \frac{\sum_{r \in [m]} a_r \exp(w_r^\top x) w_r}{\sum_{r \in [m]} \exp(w_r^\top x)}$$

Here we use $w_r \in \mathbb{R}^d$ to denote the r-th column of $W \in \mathbb{R}^{d \times m}$.

To further break down the complicated F for more convenience analysis. We give an operator function α as follows:

Definition G.3. *We define $\alpha(x)$ as follows*

$$\alpha(x) := \langle \exp(\underbrace{W^\top}_{m \times d} \underbrace{x}_{d \times 1}), \mathbf{1}_m \rangle$$

Thus, we can rewrite F in the following claim.

Claim G.4. *We can rewrite $\mathsf{F}(W, x, a) \in \mathbb{R}^d$ as follows*

$$\mathsf{F}(W, x, a) = m \underbrace{\alpha(x)^{-1}}_{\text{scalar}} \underbrace{W}_{d \times m} (\underbrace{a}_{m \times 1} \circ \underbrace{\exp(W^\top x)}_{m \times 1})$$

[2] Note that the proof of the case with a and without a are similar. We mainly focus on the proofs under the setting that includes a.

Proof. We can show

$$\begin{aligned}\mathsf{F}(W,x,a) &= m\frac{\sum_{r\in[m]} a_r \exp(w_r^\top x) w_r}{\sum_{r\in[m]} \exp(w_r^\top x)} \\ &= m\alpha(x)^{-1} \sum_{r\in[m]} a_r \exp(w_r^\top x) w_r \\ &= m\alpha(x)^{-1} W(a \circ \exp(W^\top x))\end{aligned}$$

where the first step follows from Definition G.2, the second step follows from Definition G.3 and simple algebras, the third step follows from $w_r \in \mathbb{R}^d$ is denoting the r-th column of $W \in \mathbb{R}^{d\times m}$ and simple algebras. □

In the following Definition G.6 and Definition G.5, we further derive and define two operator functions to convenient our analysis.

Definition G.5. *We define β as follows*

$$\beta_k := W_{k,*} \circ a, \forall k \in [d]$$

Let $\beta \in \mathbb{R}^{d\times m}$ be defined as $\underbrace{\beta}_{d\times m} = \underbrace{W}_{d\times m} \underbrace{\mathrm{diag}(a)}_{m\times m}$

Here, we define softmax.

Definition G.6. *We define $\mathsf{S} \in \mathbb{R}^m$ as follows*

$$\mathsf{S} := \underbrace{\alpha(x)^{-1}}_{\text{scalar}} \cdot \underbrace{\exp(W^\top x)}_{m\times 1}.$$

Here, we use β and S to re-denote the model function F.

Definition G.7. *For each $k \in [d]$, let $W_{k,*}^\top$ denote the k-th row of W, we define*

$$\mathsf{F}_k(W,x,a) := m \underbrace{\alpha(x)^{-1}}_{\text{scalar}} \langle \underbrace{W_{k,*}}_{m\times 1} \circ \underbrace{a}_{m\times 1}, \underbrace{\exp(W^\top x)}_{m\times 1}\rangle$$

Then, we can rewrite it as

$$\mathsf{F}_k(W,x,a) := m\langle \beta_k, \mathsf{S}\rangle.$$

G.1 Loss function

Here, we state the training objective that we aim to solve in the analysis.

Definition G.8. *Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}^d$. Let function $\mathsf{F} : \mathbb{R}^{d\times m} \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ be defined as Definition G.2, we define the training objective $\mathcal{L} : \mathbb{R}^{m\times d} \to \mathbb{R}$ as follows:*

$$\mathcal{L}(W) := 0.5 \sum_{i=1}^n \|\mathsf{F}(W, x_i, a) - y_i\|_2^2$$

H Gradient Computation

In this section, we first compute the gradients that we need for the analysis of NTK. Then we define the training dynamic of our model in the process of gradient descent.

H.1 Computing Gradient

We give our computation of the gradients as the following lemma.

Lemma H.1. *If the following conditions hold*

- *Let $W \in \mathbb{R}^{d \times m}$ and $a \in \mathbb{R}^m$ be defined as Definition G.1.*
- *Let $\alpha(x) \in \mathbb{R}$ be defined as Definition G.3*
- *Let $\mathsf{S} \in \mathbb{R}^m$ be defined as Definition G.6*
- *Let $\mathsf{F} \in \mathbb{R}^d$ be defined as Definition G.7*

Then, we can show that for each $r \in [m]$

- **Part 1.** *For $k_1 \in [d]$, we have*

$$\frac{\mathrm{d} W^\top x}{\mathrm{d} w_{r,k_1}} = x_{k_1} e_r$$

- **Part 2.** *For $k_1 \in [d]$, we have*

$$\frac{\mathrm{d} \exp(W^\top x)}{\mathrm{d} w_{r,k_1}} = (x_{k_1} e_r) \circ \exp(W^\top x)$$

- **Part 3.** *For $k_1 \in [d]$, we have*

$$\frac{\mathrm{d} \alpha(x)}{\mathrm{d} w_{r,k_1}} = \langle x_{k_1} e_r, \exp(W^\top x) \rangle$$

- **Part 4.** *For $k_1 \in [d]$, we have*

$$\frac{\mathrm{d} \alpha(x)^{-1}}{\mathrm{d} w_{r,k_1}} = -\alpha(x)^{-1} \langle x_{k_1} e_r, \mathsf{S} \rangle$$

- **Part 5.** *For $k_1 \in [d]$, we have*

$$\frac{\mathrm{d} \mathsf{S}}{\mathrm{d} w_{r,k_1}} = -\langle x_{k_1} e_r, \mathsf{S} \rangle \cdot \mathsf{S} + (x_{k_1} e_r) \circ \mathsf{S}$$

- **Part 6.** *For $k_1, k \in [d]$ and $k_1 \neq k$, we have*

$$\begin{aligned} \frac{\mathrm{d} \mathsf{F}(W, x, a)_k}{\mathrm{d} w_{r,k_1}} = & + 0 - m x_{k_1} \cdot \mathsf{S}_r \cdot \langle \beta_k, \mathsf{S} \rangle \\ & + m x_{k_1} \mathsf{S}_r \beta_{k,r} \end{aligned}$$

- **Part 7.** *For $k_1, k \in [d]$ and $k_1 = k$, we have*

$$\begin{aligned} & \frac{\mathrm{d} \mathsf{F}(W, x, a)_k}{\mathrm{d} w_{r,k}} \\ = & + m \langle a \circ e_r, \mathsf{S} \rangle - m x_k \cdot \mathsf{S}_r \cdot \langle \beta_k, \mathsf{S} \rangle \\ & + m x_k \mathsf{S}_r \beta_{k,r} \end{aligned}$$

- **Part 8.** *For $k \in [d]$, we have*

$$\begin{aligned}&\frac{\mathrm{d}\mathsf{F}(W,x,a)_k}{\mathrm{d}w_r}\\ =&\, m a_r \mathsf{S}_r \cdot e_k - m\langle \beta_k, \mathsf{S}\rangle \mathsf{S}_r \cdot x\\ &+ m\beta_{k,r}\mathsf{S}_r \cdot x\end{aligned}$$

Proof. **Proof of Part 1.**

$$\frac{\mathrm{d}W^\top x}{\mathrm{d}w_{r,k_1}} = x_{k_1} e_r$$

where this step follows from simple differential rules.

Proof of Part 2.

$$\begin{aligned}\frac{\mathrm{d}\exp(W^\top x)}{\mathrm{d}w_{r,k_1}} =&\ \exp(W^\top x) \circ \frac{\mathrm{d}W^\top x}{\mathrm{d}w_{r,k_1}}\\ =&\ (x_{k_1} e_r) \circ \exp(W^\top x)\end{aligned}$$

where the first step follows from chain rules, the second step follows from Part 1 of this Lemma.

Proof of Part 3.

$$\begin{aligned}\frac{\mathrm{d}\alpha(x)}{\mathrm{d}w_{r,k_1}} =&\ \langle \frac{\mathrm{d}\exp(W^\top x)}{\mathrm{d}w_{r,k_1}}, \mathbf{1}_m\rangle\\ =&\ \langle x_{k_1} e_r, \exp(W^\top x)\rangle\end{aligned}$$

where the first step follows from Definition G.3 and simple algebras, the second step follows from Part 2 of this Lemma.

Proof of Part 4.

$$\begin{aligned}\frac{\mathrm{d}\alpha(x)^{-1}}{\mathrm{d}w_{r,k_1}} =&\ -\alpha(x)^{-2}\frac{\mathrm{d}\alpha(x)}{\mathrm{d}w_{r,k_1}}\\ =&\ -\alpha(x)^{-1}\langle x_{k_1} e_r, \mathsf{S}\rangle\end{aligned}$$

where this step follows from chain rules, the second step follows from Part 3 of this Lemma.

Proof of Part 5.

$$\begin{aligned}&\frac{\mathrm{d}\mathsf{S}}{\mathrm{d}w_{r,k_1}}\\ =&\frac{\mathrm{d}\alpha(x)^{-1}}{\mathrm{d}w_{r,k_1}} \cdot \exp(W^\top x) + \alpha(x)^{-1} \cdot \frac{\mathrm{d}\exp(W^\top x)}{\mathrm{d}w_{r,k_1}}\\ =&\ -\alpha(x)^{-1}\langle x_{k_1} e_r, \mathsf{S}\rangle \cdot \exp(W^\top x)\\ &+ \alpha(x)^{-1} \cdot (x_{k_1} e_r) \circ \exp(W^\top x)\\ =&\ -\langle x_{k_1} e_r, \mathsf{S}\rangle \cdot \mathsf{S} + (x_{k_1} e_r) \circ \mathsf{S}\end{aligned}$$

where the first step follows from Definition G.6 and differential rules, the second step follows from Part 2 and Part 4 of this Lemma, the last step follows from simple algebras.

Proof of Part 6. For $k_1 \neq k$

$$\frac{\mathrm{d}\mathsf{F}(W,x,a)_k}{\mathrm{d}w_{r,k_1}}$$

$$
\begin{aligned}
&= +m\langle \frac{\mathrm{d}\beta_k}{\mathrm{d}w_{r,k_1}}, \mathsf{S}\rangle + m\langle \beta_k, \frac{\mathrm{d}\mathsf{S}}{\mathrm{d}w_{r,k_1}}\rangle \\
&= -m\langle x_{k_1} e_r, \mathsf{S}\rangle \cdot \langle \beta_k, \mathsf{S}\rangle + m\langle \beta_k, (x_{k_1} e_r) \circ \mathsf{S}\rangle \\
&= +0 - m x_{k_1} \cdot \mathsf{S}_r \cdot \langle \beta_k, \mathsf{S}\rangle + m x_{k_1} \mathsf{S}_r \beta_{k,r}
\end{aligned}
$$

where the first step follows from Definition G.7 and simple algebras, the second step follows from Definition G.5, simple algebras and Part 5 of this Lemma, the last step follows from simple algebras.

Proof of Part 7. For $k_1 = k$

$$
\begin{aligned}
& \frac{\mathrm{d}\mathsf{F}(W,x,a)_k}{\mathrm{d}w_{r,k}} \\
&= +m\langle \frac{\mathrm{d}\beta_k}{\mathrm{d}w_{r,k}}, \mathsf{S}\rangle + m\langle \beta_k, \frac{\mathrm{d}\mathsf{S}}{\mathrm{d}w_{r,k}}\rangle \\
&= +m\langle a \circ e_r, \mathsf{S}\rangle - m\langle x_k e_r, \mathsf{S}\rangle \cdot \langle \beta_k, \mathsf{S}\rangle \\
&\quad + m\langle \beta_k, (x_k e_r) \circ \mathsf{S}\rangle \\
&= +m\langle a \circ e_r, \mathsf{S}\rangle - m x_k \cdot \mathsf{S}_r \cdot \langle \beta_k, \mathsf{S}\rangle \\
&\quad + m x_k \mathsf{S}_r \beta_{k,r}
\end{aligned}
$$

where the first step follows from Definition G.7 and simple algebras, the second step follows from Definition G.5, simple algebras and Part 5 of this Lemma, the last step follows from simple algebras.

Proof of Part 8.

This part of proof follows from the combination of Part 6 and Part 7 of this Lemma. □

H.2 Gradient Descent

After we computed the gradient of the model function above, we are now able to define the training dynamic of F by updating weight using gradient descent.

We use e_r to denote a vector where the r-th coordinate is 1 and everywhere else is 0. $\forall r \in [m], \forall k \in [d]$, we have $\frac{\mathrm{d}\mathsf{F}(W,x,a)_k}{\mathrm{d}w_r} \in \mathbb{R}^d$ can be written as

$$
\underbrace{\frac{\mathrm{d}\mathsf{F}_k(W,x,a)}{\mathrm{d}w_r}}_{d\times 1} \tag{6}
$$

$$
= m a_r \mathsf{S}_r \cdot e_k - m\langle \beta_k, \mathsf{S}\rangle \mathsf{S}_r \cdot x + m\beta_{k,r}\mathsf{S}_r \cdot x. \tag{7}
$$

Hence, by defining several following dynamical operator functions, we can further convenient our proofs.

We first define $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ for simplification as follows:

Definition H.2. *For each $i \in [n]$, we define $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ as*

$$
\underbrace{\mathsf{u}_i(\tau)}_{m\times 1} := \exp(\underbrace{W(\tau)^\top}_{m\times d} \underbrace{x_i}_{d\times 1})
$$

Secondly, we re-denote $\alpha_i(\tau) \in \mathbb{R}$ below, which holds due to the definition of $\alpha(x)$ and the updating of $W \in \mathbb{R}^{d\times m}$.

Definition H.3. *For each $i \in [n]$, we define $\alpha_i(\tau) \in \mathbb{R}$ as*

$$
\underbrace{\alpha_i(\tau)}_{\text{scalar}} := \langle \underbrace{\mathsf{u}_i(\tau)}_{m\times 1}, \underbrace{\mathbf{1}_m}_{m\times 1}\rangle.
$$

We define $\beta_k(\tau) \in \mathbb{R}^m$ for convenience.

Definition H.4. *For each $k \in [d]$, we define $\beta_k(\tau) \in \mathbb{R}^m$ as*

$$\underbrace{\beta_k(\tau)}_{m\times 1} = \underbrace{(W_{k,*}(\tau))}_{m\times 1} \circ \underbrace{a}_{m\times 1}$$

Remark H.5. *The purpose of defining notation β is to make our proofs more aligned with softmax NTK proofs in previous work (Li et al., 2024a).*

We define $\theta_{k,i}(\tau) \in \mathbb{R}^m$ for convenience as follows :

Definition H.6. *For each $i \in [n]$, for each $k \in [d]$, we define $\theta_{k,i}(\tau) \in \mathbb{R}^m$ as follows*

$$\underbrace{\theta_{k,i}(\tau)}_{m\times 1} := \underbrace{\beta_k(\tau)}_{m\times 1} \cdot \underbrace{\alpha_i(\tau)^{-1}}_{\text{scalar}}$$

We denote $\mathsf{S}_r(\tau)$.

Definition H.7. *For each $i \in [n]$. Let $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ be defined as*

$$\underbrace{\mathsf{S}_i(\tau)}_{m\times 1} := \underbrace{\alpha_i(\tau)^{-1}}_{\text{scalar}} \cdot \underbrace{\mathsf{u}_i(\tau)}_{m\times 1}$$

for integer $\tau \geq 0$. For $r \in [m]$, we denote $\mathsf{S}_{i,r}(\tau) \in \mathbb{R}$ as the r-th entry of vector $\mathsf{S}_i(\tau)$.

Now, we can define F at different timestamps.

Definition H.8 ($\mathsf{F}(\tau)$, dynamic prediction)**.** *For each $k \in [d]$, for each $i \in [n]$, we define $\mathsf{F}_i(\tau) \in \mathbb{R}^d$, for any timestamp τ, as*

$$\mathsf{F}_{k,i}(\tau) := m\langle \mathsf{u}(\tau), \mathbf{1}_m\rangle^{-1} \langle W(\tau)_{k,*} \circ a, \mathsf{u}(\tau)\rangle.$$

Here $x_i \in \mathbb{R}^d$. It can be rewritten as

$$\mathsf{F}_{k,i}(\tau) = m \cdot \langle \underbrace{\beta_k(\tau)}_{m\times 1}, \underbrace{\mathsf{S}_i(\tau)}_{m\times 1}\rangle.$$

and also

$$\mathsf{F}_{k,i}(\tau) = m \cdot \langle \underbrace{\theta_{k,i}(\tau)}_{m\times 1}, \underbrace{u_i(\tau)}_{m\times 1}\rangle$$

We consider d-dimensional MSE loss.

Definition H.9 (Loss function over time)**.** *We define the objective function $\mathcal{L}$ as below:*

$$\mathcal{L}(W(\tau)) := \frac{1}{2} \sum_{i\in[n]} \sum_{k\in[d]} (\mathsf{F}_{k,i}(\tau) - y_{k,i})^2.$$

Thus, we define the gradient of w.

Definition H.10 ($\Delta w_r(\tau)$)**.** *For any $r \in [m]$, we define $\Delta w_r(\tau) \in \mathbb{R}^d$ as below:*

$$\begin{aligned} & \Delta w_r(\tau) \\ := & \, m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\ & \cdot \Big(a_r \mathsf{S}_{i,r}(\tau) \cdot e_k - \langle \beta_k(\tau), \mathsf{S}_i(\tau)\rangle \mathsf{S}_{i,r}(\tau) \cdot x \\ & + \beta_{k,r} \mathsf{S}_{i,r}(\tau) \cdot x \Big) \end{aligned}$$

Here, we utilize v to simplify $\Delta w_r(\tau)$, we have the following:

Definition H.11. *For each $k \in [d]$, for each $r \in [m]$, we define $v_{k,r}(\tau) \in \mathbb{R}^m$ as follows*

$$v_{k,r}(\tau) := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau).$$

Note that we can simplify the gradient calculation by the fact $1 = \langle \mathbf{1}_m, \mathsf{S}_i(\tau) \rangle$ for $i \in [n]$. Thus, we have the following claim.

Claim H.12. *We can rewrite $\Delta w_r(\tau)$ as follows*

$$\begin{aligned}\Delta w_r(\tau) = \ & m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\ & \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_i(\tau) \rangle \cdot \mathsf{S}_{i,r}(\tau) \cdot x_i \\ & + a_r \mathsf{S}_{i,r}(\tau) e_k \Big)\end{aligned}$$

Proof. We have

$$\begin{aligned}& \Delta w_r(\tau) \\ = & \ m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \cdot \Big(a_r \mathsf{S}_{i,r}(\tau) \cdot e_k \\ & - \langle \beta_k(\tau), \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x + \beta_{k,r} \mathsf{S}_{i,r}(\tau) \cdot x \Big) \\ = & \ m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\ & \cdot \Big(a_r \mathsf{S}_{i,r}(\tau) \cdot e_k - \langle \beta_k(\tau), \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x \\ & + \beta_{k,r} \langle \mathbf{1}_m, \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x \Big) \\ = & \ m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\ & \cdot \Big(a_r \mathsf{S}_{i,r}(\tau) \cdot e_k - \langle \beta_k(\tau), \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x \\ & + \langle \beta_{k,r} \cdot \mathbf{1}_m, \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x \Big) \\ = & \ m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \cdot \Big(a_r \mathsf{S}_{i,r}(\tau) \cdot e_k \\ & + \langle \beta_{k,r} \cdot \mathbf{1}_m - \beta_k(\tau), \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x \Big) \\ = & \ m \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \cdot \Big(a_r \mathsf{S}_{i,r}(\tau) \cdot e_k \\ & + \langle v_{k,r}(\tau), \mathsf{S}_i(\tau) \rangle \mathsf{S}_{i,r}(\tau) \cdot x \Big)\end{aligned}$$

where the first step follows from Definition H.10, the second step follows from the fact $1 = \langle \mathbf{1}_m, \mathsf{S}_i(\tau) \rangle$ for $i \in [n]$, the third and fourth steps follow from simple algebras, the last step follows from Definition H.11. □

We use the gradient descent (GD) algorithm with the learning rate η to train the network. As we only train the hidden layer W and fix a, we have the following gradient update rule.

Definition H.13 (Gradient descent)**.** *The gradient descent algorithm for optimizing the weight matrix W is defined as:*

$$W(\tau + 1) = W(\tau) - \eta \Delta W(\tau).$$

where $\Delta W(\tau) \in \mathbb{R}^{d\times m}$ and $\Delta w_r(\tau) \in \mathbb{R}^d$ is the r-th column of $\Delta W(\tau)$ defined in Definition H.10.

I Neural Tangent Kernel

Now in this section, we give the exact computation of NTK in our analysis below.

Definition I.1 (Kernel function, Definition 3.6 in (Li et al., 2024a))**.** *For simplicity, we denote $\mathsf{S}(W^\top x_i)$ as $\mathsf{S}_i \in \mathbb{R}^m_{\geq 0}$ and $v_{k,r} = \beta_{k,r} \cdot \mathbf{1}_m - \beta_k \in \mathbb{R}^m$. We define the function (Gram matrix) $H : \mathbb{R}^{d\times m} \to \mathbb{R}^{nd\times nd}$ as following*

$$H(W) := \begin{bmatrix} H_{1,1} & H_{1,2} & \cdots & H_{1,d} \\ H_{2,1} & H_{2,2} & \cdots & H_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ H_{d,1} & H_{d,2} & \cdots & H_{d,d} \end{bmatrix},$$

and for each $k_1, k_2 \in [d]$, we have $H_{k_1,k_2} \in \mathbb{R}^{n\times n}$ is defined as

$$\begin{aligned} &[H_{k_1,k_2}]_{i,j}(W) \\ :=& \frac{1}{m} x_i^\top x_j \sum_{r=1}^m \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot m\mathsf{S}_{i,r} \cdot \\ &\langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot m\mathsf{S}_{j,r}. \end{aligned}$$

For any timestamp τ, for simplicity, we denote $H(\tau) := H(W(\tau))$ and denote $H(0)$ as H^*.

I.1 Kernel Perturbation

The purpose of this section is to prove Lemma I.3. In the proof, we do not use concentration inequality. Please see Remark I.2 for more details.

Remark I.2. *In the proof of Lemma I.3, we do not use concentration bound as previous work (Song and Yang, 2019; Munteanu et al., 2022; Gao et al., 2023a). The reason is that we consider the worst case. In general, $\mathbb{E}[H(W) - H(\widetilde{W})] \neq \mathbf{0}_{nd\times nd}$. Thus, using the concentration bound may not gain any benefits.*

Lemma I.3. *If the following conditions hold*

- *Let $C > 10$ denote a sufficiently large constant*
- *Let $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.*
- *Let $R \in (0, 0.01)$.*
- *Let $x_i \in \mathbb{R}^d$ and $\|x_i\|_2 \leq 1$ for all $i \in [n]$.*
- *Let $\widetilde{W} = [\widetilde{w}_1, \cdots, \widetilde{w}_m] \in \mathbb{R}^{d\times m}$, where $\widetilde{w}_1, \cdots, \widetilde{w}_m$ are are i.i.d. draw from $\mathcal{N}(0, \sigma^2 I_d)$.*
- *Let $W = [w_1, \cdots, w_m] \in \mathbb{R}^{d\times m}$ and satisfy $\|\widetilde{w}_r - w_r\|_2 \leq R$ for any $r \in [m]$.*
- *Let $v_{k,r} = \beta_{k,r} \cdot \mathbf{1}_m - \beta_k \in \mathbb{R}^m$, for any $k \in [d]$ and for any $r \in [m]$. Note that $\beta_{k,r}$ is the r-th in β_k.*
- *Let $\alpha_i = \langle \mathbf{1}_m, \exp(W^\top x_i)\rangle$ and $\widetilde{\alpha}_i = \langle \mathbf{1}_m, \exp(\widetilde{W}^\top x_i)\rangle$, $\forall i \in [n]$.*
- *Let H be defined as Definition I.1.*

Then, we have

- *Part 1. Then with probability at least* $1 - \delta / \operatorname{poly}(nd)$*,*

$$\begin{aligned} & |[H_{k_1,k_2}]_{i,j}(W) - [H_{k_1,k_2}]_{i,j}(\widetilde{W})| \\ \leq & 8R \cdot \exp(22B). \end{aligned}$$

- *Part 2. Then with probability at least* $1 - \delta$*, we have*

$$\|H(W) - H(\widetilde{W})\|_F \leq 8R\sqrt{nd} \cdot \exp(22B).$$

Proof. For simplicity, we give the following notations:

- Note that $\widetilde{\mathsf{S}}_i := \exp(\widetilde{W}(\tau)^\top x_i) \cdot \widetilde{\alpha}_i^{-1}$.
- Note that $\widetilde{\beta}_k := \widetilde{W}_{k,*} \circ a$.
- Note that $\widetilde{v}_{k,r} := \widetilde{\beta}_{k,r} \cdot \mathbf{1}_m - \widetilde{\beta}_k$.

Proof of Part 1. We have

$$\begin{aligned} & |[H_{k_1,k_2}]_{i,j}(W) - [H_{k_1,k_2}]_{i,j}(\widetilde{W})| \\ = & m x_i^\top x_j \sum_{r=1}^m (B_{1,r} + B_{2,r} + B_{3,r} \\ & + B_{4,r} + B_{5,r} + B_{6,r}) \end{aligned}$$

here, we define:

$$\begin{aligned} B_{1,r} := & \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot \mathsf{S}_{j,r} \\ & - \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ B_{2,r} := & \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ & - \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ B_{3,r} := & \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ & - \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ B_{4,r} := & \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ & - \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \widetilde{\mathsf{S}}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ B_{5,r} := & \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \widetilde{\mathsf{S}}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ & - \langle v_{k_1,r}, \widetilde{\mathsf{S}}_i \rangle \cdot \widetilde{\mathsf{S}}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ B_{6,r} := & \langle v_{k_1,r}, \widetilde{\mathsf{S}}_i \rangle \cdot \widetilde{\mathsf{S}}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\ & - \langle \widetilde{v}_{k_1,r}, \widetilde{\mathsf{S}}_i \rangle \cdot \widetilde{\mathsf{S}}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \end{aligned}$$

Before we bound all terms, we provide a tool as follows:

$$\begin{aligned} & \|v_{k,r} - \widetilde{v}_{k,r}\|_2^2 \\ = & \sum_{r_1=1}^m (v_{k,r,r_1} - \widetilde{v}_{k,r,r_1})^2 \\ = & \sum_{r_1=1}^m (\beta_{k,r} - \beta_{k,r_1} - \widetilde{\beta}_{k,r} + \widetilde{\beta}_{k,r_1})^2 \end{aligned}$$

$$
\begin{aligned}
&= \sum_{r_1=1}^{m} (a_r W_{k,r} - a_{r_1} W_{k,r} - a_r \widetilde{W}_{k,r} \\
&\quad + a_{r_1} \widetilde{W}_{k,r})^2 \\
&= \sum_{r_1=1}^{m} (a_r (W_{k,r} - \widetilde{W}_{k,r}) \\
&\quad + a_{r_1} (\widetilde{W}_{k,r_1} - W_{k,r_1}))^2 \\
&\leq \sum_{r_1=1}^{m} (|W_{k,r} - \widetilde{W}_{k,r}| \ + |\widetilde{W}_{k,r_1} - W_{k,r_1}|)^2 \\
&\leq \sum_{r_1=1}^{m} 4R^2 \\
&\leq m 4R^2
\end{aligned} \tag{8}
$$

where the first step follows from the definition of ℓ_2 norm, the second step follows from the definition of $v_{k,r}$, the third step follows from Definition G.5, the fourth and fifth steps follow from simple algebras, the sixth step follows from $\|w_r - v_r\|_\infty \leq \|w_r - v_r\|_2 \leq R$, the last step follows from simple algebras.

To bound $B_{1,r}$, we have

$$
\begin{aligned}
& |B_{1,r}| \\
:= & |\langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot \mathsf{S}_{j,r} \\
& - \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r}| \\
= & |\langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot (\mathsf{S}_{j,r} - \widetilde{\mathsf{S}}_{j,r})| \\
\leq & \frac{\exp(15B)}{m} \cdot |\mathsf{S}_{j,r} - \widetilde{\mathsf{S}}_{j,r}| \\
\leq & \frac{R \exp(19B + 3R)}{m^2}
\end{aligned}
$$

where the first step follows from the definition of $B_{1,r}$, the second step follows from simple algebras, the third step follows from Part 6 of Lemma M.2 and $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, the last step follows from Part 12 of Lemma M.1.

To bound $B_{2,r}$, we have

$$
\begin{aligned}
& |B_{2,r}| \\
:= & |\langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r} \\
& - \langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r}| \\
= & |\langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \mathsf{S}_j - \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r}| \\
\leq & \frac{2B \exp(12B)}{m^2} \cdot |\langle \frac{1}{2B} v_{k_2,r}, \mathsf{S}_j - \widetilde{\mathsf{S}}_j \rangle| \\
\leq & \frac{2BR \exp(16B + 3R)}{m^2}
\end{aligned}
$$

where the first step follows from the definition of $B_{2,r}$, the second step follows from simple algebras, the third step follows from Part 6 of Lemma M.2 and $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, the last step follows from Part 13 of Lemma M.1 and $\|v_{k,r}\|_\infty \leq 2B$ by simple algebras.

To bound $B_{3,r}$, we have

$$
\begin{aligned}
& |B_{3,r}| \\
:= & |\langle v_{k_1,r}, \mathsf{S}_i \rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r}, \widetilde{\mathsf{S}}_j \rangle \cdot \widetilde{\mathsf{S}}_{j,r}
\end{aligned}
$$

$$
\begin{aligned}
&- \langle v_{k_1,r}, \mathsf{S}_i\rangle \cdot \mathsf{S}_{i,r} \cdot \langle \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j\rangle \cdot \widetilde{\mathsf{S}}_{j,r}| \\
&= |\langle v_{k_1,r}, \mathsf{S}_i\rangle \cdot \mathsf{S}_{i,r} \cdot \langle v_{k_2,r} - \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j\rangle \cdot \widetilde{\mathsf{S}}_{j,r}| \\
&\leq \frac{\exp(12B)}{m^2} \cdot |\langle v_{k_2,r} - \widetilde{v}_{k_2,r}, \widetilde{\mathsf{S}}_j\rangle| \\
&\leq \frac{2R\exp(15B)}{m^2}
\end{aligned}
$$

where the first step follows from the definition of $B_{3,r}$, the second step follows from simple algebras, the third step follows from Part 6 of Lemma M.2 and $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, the last step follows from Cauchy-Schwarz inequality, Eq. (8) and $\|\mathsf{S}_i\|_2 \leq \frac{\exp(3B)}{\sqrt{m}}$.

The proof of bounding $B_{4,r}$ is similar to the proof of bounding $B_{1,r}$, we have $|B_{4,r}| \leq \frac{R\exp(19B+3R)}{m^2}$.
The proof of bounding $B_{5,r}$ is similar to the proof of bounding $B_{2,r}$, we have $|B_{5,r}| \leq \frac{2BR\exp(16B+3R)}{m^2}$.
The proof of bounding $B_{6,r}$ is similar to the proof of bounding $B_{3,r}$, we have $|B_{6,r}| \leq \frac{2R\exp(15B)}{m^2}$.
Now we combine all terms, we have

$$
\begin{aligned}
& |[H_{k_1,k_2}]_{i,j}(W) - [H_{k_1,k_2}]_{i,j}(\widetilde{W})| \\
&= m x_i^\top x_j \sum_{r=1}^m (B_{1,r} + B_{2,r} + B_{3,r} \\
&\quad + B_{4,r} + B_{5,r} + B_{6,r}) \\
&\leq m \sum_{r=1}^m (B_{1,r} + B_{2,r} + B_{3,r} + B_{4,r} + B_{5,r} + B_{6,r}) \\
&\leq m \sum_{r=1}^m (|B_{1,r}| + |B_{2,r}| + |B_{3,r}| \\
&\quad + |B_{4,r}| + |B_{5,r}| + |B_{6,r}|) \\
&\leq m \sum_{r=1}^m \frac{8R\exp(22B)}{m^2} \\
&\leq 8R \cdot \exp(22B)
\end{aligned}
$$

where the second step follows from $\|x_i\|_2 \leq 1$, the third step follows from simple algebras, the fourth step follows from $R \leq B$, $B \leq \exp(B)$ and the combination of all terms, the last step follows from simple algebras.

Proof of Part 2. This proof follows from Part 1 of this Lemma and the definition of Frobenius norm. □

I.2 Kernel PSD during Training Process

Claim I.4. *If the following conditions hold:*

- *Let* $\lambda = \lambda_{\min}(H^*)$
- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let timestamp* $\tau \geq 0$ *denotes as a integer.*
- *Denote* H^* *as* $H(W)$ *in Definition I.1.*
- *Denote* $H(\tau)$ *as* $H(\widetilde{W})$ *in Definition I.1.*

- *Let* $D := 2\lambda^{-1} \cdot \exp(20B)\frac{\sqrt{n}d}{m}\|Y - \mathsf{F}(0)\|_F$
- *Let* $\|w_r(t) - w_r(0)\|_2 \leq D < R = \lambda/\operatorname{poly}(n, d, \exp(B))$, $\forall r \in [m]$, $\forall t \geq 0$

Then, with a probability at least $1 - \delta$*, we have*

$$\lambda_{\min}(H(\tau)) \geq \lambda/2$$

Proof. By Lemma I.3, with a probability at least $1 - \delta$, we have

$$\begin{aligned}\|H^* - H(\tau)\|_F &\leq 8R\sqrt{nd}\exp(22B) \\ &\leq \lambda/2 \end{aligned} \tag{9}$$

where the first step follows from Part 2 of Lemma I.3, the second step follows by choice of R.

By eigenvalue perturbation theory, we have

$$\begin{aligned}\lambda_{\min}(H(\tau)) &\geq \lambda_{\min}(H^*) - \|H(\tau) - H^*\| \\ &\geq \lambda_{\min}(H^*) - \|H(\tau) - H^*\|_F \\ &\geq \lambda_{\min}(H^*) - \lambda/2 \\ &\geq \lambda/2\end{aligned}$$

where the first step comes from triangle inequality, the second step is due to Frobenius norm, the third step is due to Eq. (9), the last step follows from $\lambda_{\min}(H^*) = \lambda$. □

J Loss Decomposition

In this section, we provide the lemma below to decompose it into five terms, and then we will give bounds to four terms.

Lemma J.1. *Assuming the following condition is met:*

- *Let* $W \in \mathbb{R}^{d\times m}$ *and* $a \in \mathbb{R}^m$ *as Definition G.1.*
- *Let* $\lambda = \lambda_{\min}(H^*)$
- *For* $i, j \in [n]$ *and* $k_1, k_2 \in [d]$*.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n\times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n\times d}$ *denote the labels.*
- *Let* $\eta > 0$ *denote the learning rate.*
- *Let scalar* $v_{0,k,i} \in \mathbb{R}$ *be defined as follows*

$$\begin{aligned}&v_{0,k,i} \\ &:= m\sum_{r\in[m]}(\theta_{k,i,r}(\tau+1) \\ &\quad - \theta_{k,i,r}(\tau)) \cdot \mathsf{u}_{i,r}(\tau+1)\end{aligned}$$

- *Let scalar* $v_{1,k,i} \in \mathbb{R}$ *be defined as follows*

$$\begin{aligned}&v_{1,k,i} \\ &:= m\sum_{r=1}^{m}\theta_{k,i,r}(\tau) \\ &\quad \cdot \mathsf{u}_{i,r}(\tau) \cdot (-\eta\langle\Delta w_r(\tau), x_i\rangle)\end{aligned}$$

- *Let scalar $v_{2,k,i} \in \mathbb{R}$ be defined as follows*

$$\begin{aligned} & v_{2,k,i} \\ & := m \sum_{r=1}^{m} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \cdot \eta^2 \\ & \cdot \Theta(1) \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \end{aligned}$$

- **Gradient Property.** $\eta \|\Delta w_r(i)\|_2 \leq 0.01$, $\forall r \in [m]$, $\forall i \in [\tau]$
- $C_0 = 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_0) \rangle$
- $C_1 = 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_1) \rangle$
- $C_2 = 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_2) \rangle$
- $C_3 = \|\mathsf{F}(\tau + 1) - \mathsf{F}(\tau)\|_F^2$

Then, we can show

$$\begin{aligned} & \|\mathsf{F}(\tau + 1) - Y\|_F^2 \\ = & \|\mathsf{F}(\tau) - Y\|_F^2 + C_0 + C_1 + C_2 + C_3. \end{aligned}$$

Proof. The expression $\|Y - \mathsf{F}(\tau + 1)\|_F^2 = \|\mathrm{vec}(Y - \mathsf{F}(\tau + 1))\|_2^2$ can be rewritten in the following:

$$\begin{aligned} & \|\mathrm{vec}(Y - \mathsf{F}(\tau + 1))\|_2^2 \\ = & \|\mathrm{vec}(Y - \mathsf{F}(\tau) - (\mathsf{F}(\tau + 1) - \mathsf{F}(\tau)))\|_2^2 \\ = & \|\mathrm{vec}(Y - \mathsf{F}(\tau))\|_2^2 \\ & - 2\,\mathrm{vec}(Y - \mathsf{F}(\tau))^\top \mathrm{vec}(\mathsf{F}(\tau + 1) - \mathsf{F}(\tau)) \\ & + \|\mathrm{vec}(\mathsf{F}(\tau + 1) - \mathsf{F}(\tau))\|_2^2. \end{aligned} \tag{10}$$

where the first step follows from simple algebra, the last step follows from simple algebra.

Recall the update rule (Definition H.13),

$$w_r(\tau + 1) = w_r(\tau) - \eta \cdot \Delta w_r(\tau)$$

In the following manner, $\forall k \in [d]$, we can express $\mathsf{F}_k(\tau + 1) - \mathsf{F}_k(\tau) \in \mathbb{R}^n$:

$$\begin{aligned} & \mathsf{F}_{k,i}(\tau + 1) - \mathsf{F}_{k,i}(\tau) \\ = & m \sum_{r \in [m]} (\theta_{k,i,r}(\tau + 1)\mathsf{u}_{i,r}(\tau + 1) \\ & - \theta_{k,i,r}(\tau)\mathsf{u}_{i,r}(\tau)) \\ = & + \sum_{r \in [m]} (\theta_{k,i,r}(\tau + 1) - \theta_{k,i,r}(\tau)) \cdot \mathsf{u}_{i,r}(\tau + 1) \\ & + m \sum_{r \in [m]} \theta_{k,i,r} \cdot (\mathsf{u}_{i,r}(\tau + 1) - \mathsf{u}_{i,r}(\tau)) \\ = & + \sum_{r \in [m]} (\theta_{k,i,r}(\tau + 1) - \theta_{k,i,r}(\tau)) \cdot u_{i,r}(\tau + 1) \\ & + m \sum_{r \in [m]} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \end{aligned}$$

$$
\begin{aligned}
&\cdot (\exp(-\eta\langle \Delta w_r(\tau), x_i\rangle) - 1) \\
=& + \sum_{r\in[m]} (\theta_{k,i,r}(\tau+1) - \theta_{k,i,r}(\tau)) \cdot \mathsf{u}_{i,r}(\tau+1) \\
&+ m \sum_{r\in[m]} \theta_{k,i,r}(\tau)\mathsf{u}_{i,r}(\tau) \\
&\cdot (-\eta\langle \Delta w_r(\tau), x_i\rangle + \Theta(1)\eta^2\langle \Delta w_r(\tau), x_i\rangle^2) \\
=& v_{0,k,i} + v_{1,k,i} + v_{2,k,i} \qquad (11)
\end{aligned}
$$

where the first step is due to the definition of $\mathsf{F}_{k,i}(\tau)$, the second step is from the simple algebra, the third step is due to $|\eta\Delta w_r(\tau)^\top x_i| \leq 0.01$ (due to **Gradient Property** and $\|x_i\|_2 \leq 1$), the fourth step follows from the Taylor series approximation, the last step follows from

$$
\begin{aligned}
v_{0,k,i} :=& m \sum_{r\in[m]} (\theta_{k,i,r}(\tau+1) \\
&- \theta_{k,i,r}(\tau)) \cdot \mathsf{u}_{i,r}(\tau+1) \\
v_{1,k,i} :=& m \sum_{r=1}^{m} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \cdot \\
&(-\eta\langle \Delta w_r(\tau), x_i\rangle) \\
v_{2,k,i} :=& m \sum_{r=1}^{m} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \\
&\cdot \eta^2 \cdot \Theta(1) \cdot \langle \Delta w_r(\tau), x_i\rangle^2
\end{aligned}
$$

Here $v_{0,k,i}$ and $v_{1,k,i}$ are linear in η and $v_{2,k,i}$ is quadratic in η. Thus, $v_{0,k,i}$ and $v_{1,k,i}$ are the first order term, and $v_{2,k,i}$ is the second order term.

We can rewrite the second term in the Eq. (10) above as below:

$$
\begin{aligned}
&\langle \mathrm{vec}(Y - \mathsf{F}(\tau)), \mathrm{vec}(\mathsf{F}(\tau+1) - \mathsf{F}(\tau))\rangle \\
=& \langle \mathrm{vec}(Y - \mathsf{F}(\tau)), \mathrm{vec}(v_0 + v_1 + v_2)\rangle \\
=& \langle \mathrm{vec}(Y - \mathsf{F}(\tau)), \mathrm{vec}(v_0)\rangle + \langle \mathrm{vec}(Y \\
&- \mathsf{F}(\tau)), \mathrm{vec}(v_1)\rangle + \langle \mathrm{vec}(Y - \mathsf{F}(\tau)), \mathrm{vec}(v_2)\rangle
\end{aligned}
$$

where the first step follows from Eq.(11), the second step follows from simple algebras.

Therefore, we can conclude that

$$
\begin{aligned}
&\|\mathsf{F}(\tau+1) - Y\|_F^2 \\
=& \|\mathsf{F}(\tau) - Y\|_F^2 + C_0 + C_1 + C_2 + C_3.
\end{aligned}
$$

□

The below lemma analyzes the first-order term that is making progress.

Lemma J.2 (Progress terms). *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$
- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2}n^2d^2\exp(30B)\sqrt{\log(nd/\delta)})$

- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_2 \in [d]$.
- *Let* $\beta_k(\tau) \in \mathbb{R}^m$ *be defined as Definition G.5.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_{k,r} := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*
- *Let* $\eta > 0$ *denote the learning rate.*
- *Let scalar* $v_{1,1,k,i} \in \mathbb{R}$ *be defined as follows (we omit* (τ) *in the following terms)*

$$\begin{aligned} v_{1,1,k,i} = m^2 \sum_{r \in [m]} & \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \\ & \cdot (-\eta \sum_{j=1}^{n} \sum_{k_2=1}^{d} (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\ & \cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau) \rangle) \cdot \mathsf{S}_{j,r}(\tau) \Big) \cdot x_j^\top) x_i \end{aligned}$$

- *Let* $C_{1,1} := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_{1,1}) \rangle$

Then, we have

- $C_{1,1} \leq -1.6 m \eta \, \mathrm{vec}(\mathsf{F}(\tau) - Y)^\top H(\tau) \, \mathrm{vec}(\mathsf{F}(\tau) - Y)$

Proof. We have

$$\begin{aligned} v_{1,1,k,i} = & m^2 \sum_{r \in [m]} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \\ & \cdot (-\eta \sum_{j=1}^{n} \sum_{k_2=1}^{d} (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\ & \cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau) \rangle) \cdot \mathsf{S}_{j,r}(\tau) \Big) \cdot x_j^\top) x_i \\ = & m^2 \sum_{r \in [m]} \beta_{k,r}(\tau) \cdot \alpha_i(\tau)^{-1} \cdot \mathsf{u}_{i,r}(\tau) \\ & \cdot (-\eta \sum_{j=1}^{n} \sum_{k_2=1}^{d} (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\ & \cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau) \rangle) \cdot \mathsf{S}_{j,r}(\tau) \Big) \cdot x_j^\top) x_i \\ = & m^2 \sum_{r \in [m]} \beta_{k,r}(\tau) \cdot \mathsf{S}_{i,r}(\tau) \\ & \cdot (-\eta \sum_{j=1}^{n} \sum_{k_2=1}^{d} (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\ & \cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau) \rangle) \cdot \mathsf{S}_{j,r}(\tau) \Big) \cdot x_j^\top) x_i \end{aligned}$$

$$
\begin{aligned}
&= m^2 \sum_{r\in[m]} \langle \beta_{k,r}(\tau)\cdot \mathbf{1}_m, \mathsf{S}_i(\tau)\rangle \cdot \mathsf{S}_{i,r}(\tau)\\
&\quad \cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j} - y_{k_2,j})\\
&\quad \cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau)\rangle) \cdot \mathsf{S}_{j,r}(\tau)\Big) \cdot x_j^\top) x_i\\
&= m^2 \sum_{r\in[m]} (\langle v_{k,r}, \mathsf{S}_i(\tau)\rangle\\
&\quad + \langle \beta_k(\tau), \mathsf{S}_i(\tau)\rangle) \cdot \mathsf{S}_{i,r}\\
&\quad \cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j})\\
&\quad \cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau)\rangle) \cdot \mathsf{S}_{j,r}(\tau)\Big) \cdot x_j^\top) x_i\\
&= m^2 (Q_{1,1,k,i} + Q_{1,2,k,i})
\end{aligned}
$$

where the first step follows from the definition of $v_{1,1,k,i}$, the second step follows from Definition H.6, the third step follows from Definition H.7, the fourth step follows from $\langle \beta_{k,r}(\tau)\cdot \mathbf{1}_m, \mathsf{S}_i\rangle = \beta_{k,r}(\tau)$, the fifth step follows from the definition of v_k for $k \in [d]$ and simple algebras, the last step holds since we define

$$
\begin{aligned}
Q_{1,1,k,i} :=& \sum_{r\in[m]} \langle v_{k,r}, \mathsf{S}_i(\tau)\rangle \cdot \mathsf{S}_{i,r}(\tau)\\
&\cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j})\\
&\cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau)\rangle) \cdot \mathsf{S}_{j,r}(\tau)\Big) \cdot x_j^\top) x_i,\\
Q_{1,2,k,i} :=& \sum_{r\in[m]} \langle \beta_k(\tau), \mathsf{S}_i(\tau)\rangle \cdot \mathsf{S}_{i,r}(\tau)\\
&\cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j})\\
&\cdot \Big((\langle v_{k_2,r}, \mathsf{S}_j(\tau)\rangle) \cdot \mathsf{S}_{j,r}(\tau)\Big) \cdot x_j^\top) x_i.
\end{aligned}
$$

Bounding first term. Then for the first term $Q_{1,1,k,i}$, we have its quantity

$$
\begin{aligned}
&\sum_{i=1}^n \sum_{k=1}^d Q_{1,1,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})\\
=& -\frac{1}{m}\eta \,\mathrm{vec}(\mathsf{F}(\tau) - Y)^\top H(\tau) \,\mathrm{vec}(\mathsf{F}(\tau) - Y)
\end{aligned}
$$

where this step follows from Definition I.1.

Bounding second term. On the other hand, for the second term $Q_{1,2,k,i}$, we have its quantity,

$$
\begin{aligned}
&|\sum_{i=1}^n \sum_{k=1}^d Q_{1,2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})|\\
\leq& \eta | \frac{\exp(9B)}{m^3} \sum_{i=1}^n \sum_{j=1}^n \sum_{r=1}^m \sum_{k=1}^d \sum_{k_2=1}^d\\
&\sigma_r C_{k,k_2,r} (\mathsf{F}_{k,i}(\tau) - y_{k,i})(\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j})|
\end{aligned}
$$

$$
\begin{aligned}
&\leq \eta \frac{\exp(9B)}{m^3} \cdot |\sum_{r=1}^{m} \sigma_r \max_{k,k_2 \in [d]} C_{k,k_2,r}| \\
&\quad \cdot \|(\mathsf{F}(\tau) - Y) \otimes (\mathsf{F}(\tau) - Y)\|_1 \\
&\leq \eta \frac{\exp(9B)}{m^3} \cdot |\sum_{r=1}^{m} \sigma_r \max_{k,k_2 \in [d]} C_{k,k_2,r}| \\
&\quad \cdot \|\mathsf{F}(\tau) - Y\|_1^2 \\
&\leq \eta \frac{nd\exp(9B)}{m^3} \cdot |\sum_{r=1}^{m} \sigma_r \max_{k,k_2 \in [d]} C_{k,k_2,r}| \\
&\quad \cdot \|\mathsf{F}(\tau) - Y\|_F^2 \\
&\leq \eta \frac{\exp(9B)}{m^3\lambda} |\sum_{r=1}^{m} \sigma_r \max_{k,k_2 \in [d]} C_{k,k_2,r}| \\
&\quad \cdot \mathrm{vec}(\mathsf{F}(\tau) - Y)^\top H(\tau) \mathrm{vec}(\mathsf{F}(\tau) - Y)
\end{aligned}
$$

where the first step follows from $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, $\|\mathsf{S}_i\|_2 \leq \frac{\exp(3B)}{\sqrt{m}}$, $\|x_i\| \leq 1$ and

$$
C_{k,k_2,r} := \|\beta_k(\tau)\|_2 \cdot \|v_{k_2,r}\|_2, \sigma_r \in \{+1, -1\}
$$

the second and third steps follow from the definition of Kronecker product, the fourth step follows from $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n \times d}$, the last step follows from $\mathrm{vec}(\mathsf{F}(\tau) - Y)^\top H(\tau) \mathrm{vec}(\mathsf{F}(\tau) - Y) \geq \lambda \|\mathsf{F} - Y\|_F^2$.

Thus, by following Part 2 and Part 3 of Lemma M.2, we have

$$
C_{k,k_2,r} = \|\beta_k(\tau)\|_2 \cdot \|v_{k_2,r}\|_2 \leq 2mB^2.
$$

Besides, we apply Hoeffding inequality (Lemma F.4) to all random variables $\sigma_r \max_{k,k_2 \in [d]} C_{k,k_2,r}$ for $r \in [m]$, especially $\mathbb{E}[\sum_{r=1}^{m} \sigma_r \max_{k,k_2 \in [d]} C_{k,k_2,r}] = 0$ due to the symmetry of a_r, we have

$$
\begin{aligned}
& |\sum_{i=1}^{n} \sum_{k=1}^{d} Q_{1,2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & C\eta \frac{nd\exp(9B)}{m^3\lambda} \\
& \cdot \mathrm{vec}(\mathsf{F}(\tau) - Y)^\top H(\tau) \mathrm{vec}(\mathsf{F}(\tau) - Y) \\
& \cdot mB^2 \sqrt{m \log(nd/\delta)}
\end{aligned}
$$

with probability at least $1 - \delta/\mathrm{poly}(nd)$.

Note that by Lemma condition, we have

$$
C \frac{nd\exp(9B)}{m^3\lambda} \cdot mB^2 \sqrt{m \log(nd/\delta)} \leq 0.2 \frac{1}{m}.
$$

Finally, we complete the proof with the result

$$
\begin{aligned}
& C_{1,1} \\
\leq & -1.6m\eta \,\mathrm{vec}(\mathsf{F}(\tau) - Y)^\top H(\tau) \mathrm{vec}(\mathsf{F}(\tau) - Y)
\end{aligned}
$$

□

Below, we prove all other terms are small when m is large enough compared to the progressive term.

Lemma J.3 (Minor effects on non-progress term)**.** *If the following*

- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B) \sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_2 \in [d]$
- *Let scalar* $v_{0,k,i} \in \mathbb{R}$ *be defined as follows*

$$\begin{aligned} & v_{0,k,i} \\ := & m \sum_{r \in [m]} (\theta_{k,i,r}(\tau+1) \\ & - \theta_{k,i,r}(\tau)) \cdot \mathsf{u}_{i,r}(\tau+1) \end{aligned}$$

- *Let scalar* $v_{1,2,k,i} \in \mathbb{R}$ *be defined as follows (we omit* (τ) *in the following terms)*

$$\begin{aligned} & v_{1,2,k,i} \\ = & m^2 \sum_{r \in [m]} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \\ & \cdot (-\eta \sum_{j=1}^{n} \sum_{k_2=1}^{d} (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\ & \cdot a_r \mathsf{S}_{j,r}(\tau) e_{k_2}^\top) x_i \end{aligned}$$

- *Let scalar* $v_{2,k,i} \in \mathbb{R}$ *be defined as follows*

$$\begin{aligned} v_{2,k,i} := & m \sum_{r=1}^{m} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \cdot \eta^2 \\ & \cdot \Theta(1) \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \end{aligned}$$

- *Let* $C_0 := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_0) \rangle$
- *Let* $C_{1,2} := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_{1,2}) \rangle$
- *Let* $C_2 := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_2) \rangle$
- *Let* $C_3 := \|\mathsf{F}(\tau+1) - \mathsf{F}(\tau)\|_F^2$

Then, we have

- $|C_0| \leq 0.1 m\eta\lambda \cdot \|\mathsf{F}(\tau) - Y\|_F^2$
- $|C_{1,2}| \leq 0.1 m\eta\lambda \cdot \|\mathsf{F}(\tau) - Y\|_F^2$
- $|C_2| \leq \eta^2 m \cdot n^2 d^2 \exp(16B) \cdot \|\mathsf{F}(\tau) - Y\|_F^2$
- $|C_3| \leq \eta^2 m^2 \cdot \|\mathsf{F}(\tau) - Y\|_F^2$

Proof. This proof follows from Lemma J.4, Lemma J.5, Lemma J.6 and Lemma J.7. □

J.1 Bounding C_0

Lemma J.4. *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$
- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.

- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B)\sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_1 \in [d]$.
- *Let* $\beta_k(\tau) \in \mathbb{R}^m$ *be defined as Definition G.5.*
- *Let* $\alpha_i(\tau) \in \mathbb{R}$ *be defined as Definition G.3.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_k := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*
- *Let* $\eta \in (0, 1/m)$ *denote the learning rate.*
- *Let scalar* $v_{0,k,i} \in \mathbb{R}$ *be defined as follows (we omit* (τ) *in the following terms)*

$$v_{0,k,i} = m \sum_{r \in [m]} (\theta_{k,i,r}(\tau+1) - \theta_{k,i,r}(\tau)) \cdot \mathsf{u}_{i,r}(\tau+1)$$

- *Let* $C_0 := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_0) \rangle$

Then, with a probability at least $1 - \delta/\operatorname{poly}(nd)$, *we have*

$$|C_0| \leq 0.1 \eta m \lambda \|\mathsf{F}(\tau) - Y\|_F^2.$$

Proof. By Claim H.12, we have

$$\begin{aligned}
&\Delta w_r(\tau) \\
&= m \sum_{i=1}^{n} \sum_{k=1}^{d} (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_i(\tau) \rangle \\
&\quad \cdot \mathsf{S}_{i,r}(\tau) \cdot x_i + a_r \mathsf{S}_{i,r}(\tau) e_k \Big)
\end{aligned}$$

Then the k_1-th entry $\Delta w_{r,k}(\tau)$ for $k_1 \in [d]$ should be

$$\begin{aligned}
&\Delta w_{r,k_1}(\tau) \\
&= m \sum_{i=1}^{n} \sum_{k=1}^{d} (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\
&\quad \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_i(\tau) \rangle \cdot \mathsf{S}_{i,r}(\tau) \cdot x_{i,k_1} \\
&\quad + a_r \mathsf{S}_{i,r}(\tau) e_{k,k_1} \Big) \qquad (12)
\end{aligned}$$

We have

$$v_{0,k,i}$$

$$
\begin{aligned}
&= m \sum_{r\in[m]} (\theta_{k,i,r}(\tau+1) - \theta_{k,i,r}(\tau)) \\
&\quad \cdot \mathsf{u}_{i,r}(\tau+1) \\
&= m \sum_{r\in[m]} (\beta_{k,r}(\tau+1)\alpha_i(\tau+1)^{-1} \\
&\quad - \beta_{k,r}(\tau)\alpha_i(\tau)^{-1}) \cdot \mathsf{u}_{i,r}(\tau+1) \\
&= m \sum_{r\in[m]} (\beta_{k,r}(\tau+1)\alpha_i(\tau+1)^{-1} \\
&\quad - \beta_{k,r}(\tau+1)\alpha_i(\tau)^{-1} \\
&\quad + \beta_{k,r}(\tau+1)\alpha_i(\tau)^{-1} - \beta_{k,r}(\tau)\alpha_i(\tau)^{-1}) \\
&\quad \cdot \mathsf{u}_{i,r}(\tau+1) \\
&= m \sum_{r\in[m]} (\beta_{k,r}(\tau+1) \cdot (\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1}) \\
&\quad + (\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \cdot \alpha_i(\tau)^{-1}) \\
&\quad \cdot \mathsf{u}_{i,r}(\tau+1) \\
&= m(Q_{0,1,k,i} + Q_{0,2,k,i})
\end{aligned}
$$

where the first step follows from the definition of $v_{0,k,i}$, the second step follows from Definition H.6, the third and fourth steps follow from simple algebras, the last step hold since we define

$$
\begin{aligned}
&Q_{0,1,k,i} \\
:=& \sum_{r\in[m]} \beta_{k,r}(\tau+1) \cdot (\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1}) \\
& \cdot \mathsf{u}_{i,r}(\tau+1), \\
&Q_{0,2,k,i} \\
:=& \sum_{r\in[m]} (\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \cdot \alpha_i(\tau)^{-1}) \\
& \cdot \mathsf{u}_{i,r}(\tau+1).
\end{aligned}
$$

Bounding first term. For the first term $Q_{0,1,k,i}$, we have its quantity

$$
\begin{aligned}
& |\sum_{i=1}^n \sum_{k=1}^d Q_{0,1,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m \beta_{k,r}(\tau+1) \cdot (\alpha_i(\tau+1)^{-1} \\
& - \alpha_i(\tau)^{-1}) \cdot \mathsf{u}_{i,r}(\tau+1)(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \exp(B) \cdot |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m \beta_{k,r}(\tau+1) \cdot (\alpha_i(\tau+1)^{-1} \\
& - \alpha_i(\tau)^{-1})(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & B\exp(B) \cdot |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m a_r(\alpha_i(\tau+1)^{-1} \\
& - \alpha_i(\tau)^{-1}) \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & B\exp(B) \cdot |\sum_{r=1}^m a_r(\alpha_i(\tau+1)^{-1}
\end{aligned}
$$

$$- \alpha_i(\tau)^{-1})| \cdot \sqrt{nd} \|\mathsf{F}(\tau) - Y\|_F \tag{13}$$

where the first step follows from the definition of $Q_{0,1,k,i}$, the second step follows from Part 4 of Lemma M.1 and Definition H.2, the third step follows from Part 1 of Lemma M.1 and $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n \times d}$.

By Part 2 of Lemma J.9, we have

$$\begin{aligned}
& \alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1} \\
\leq & \eta \frac{\sqrt{nd} \exp(15B)}{m^3} \\
& \cdot \|\mathsf{F}(\tau) - Y\|_F \\
& + \eta^2 \frac{nd \exp(27B)}{\sqrt{m}} \cdot \|\mathsf{F}(\tau) - Y\|_F.
\end{aligned}$$

Then we apply Hoeffding inequality (Lemma F.4) to random variables $a_r(\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1})$ for $r \in [m]$, and by $\mathbb{E}[\sum_{r=1}^m a_r(\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1})] = 0$, we have

$$\begin{aligned}
& |\sum_{r=1}^m a_r(\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1})| \\
\leq & (\eta \frac{\sqrt{nd} \exp(15B)}{m^3} + \eta^2 \frac{nd \exp(27B)}{\sqrt{m}}) \\
& \cdot \|\mathsf{F}(\tau) - Y\|_F \cdot \sqrt{m \log(nd/\delta)}.
\end{aligned} \tag{14}$$

with probability at least $1 - \delta/\operatorname{poly}(nd)$.

Through combining Eq. (14) and Eq.(13), we can show that

$$\begin{aligned}
& |\sum_{i=1}^n \sum_{k=1}^d Q_{0,1,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & (\eta \frac{nd \exp(17B)}{m^3} \cdot \|\mathsf{F}(\tau) - Y\|_F^2 \\
& + \eta^2 \frac{nd\sqrt{nd} \exp(29B)}{\sqrt{m}} \cdot \|\mathsf{F}(\tau) - Y\|_F^2) \\
& \cdot \sqrt{m \log(nd/\delta)}
\end{aligned}$$

with a probability at least $1 - \delta/\operatorname{poly}(nd)$.

Thus, by Lemma condition, we can show

$$\begin{aligned}
& \eta \frac{nd \exp(17B)}{m^3} \cdot \sqrt{m \log(nd/\delta)} \leq 0.01 \eta \lambda, \\
& \eta^2 \frac{nd\sqrt{nd} \exp(29B)}{\sqrt{m}} \cdot \sqrt{m \log(nd/\delta)} \\
\leq & \eta \frac{nd\sqrt{nd} \exp(29B)}{m} \cdot \sqrt{\log(nd/\delta)} \leq 0.01 \eta \lambda.
\end{aligned}$$

Bounding second term. On the other hand, for the second term $Q_{0,2,k,i}$, we have its quantity

$$\begin{aligned}
& |\sum_{i=1}^n \sum_{k=1}^d Q_{0,2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m (\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \cdot \alpha_i(\tau)^{-1})
\end{aligned}$$

$$
\begin{aligned}
&\cdot \mathsf{u}_{i,r}(\tau+1) \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq\ & \exp(B) \cdot |\sum_{i=1}^{n}\sum_{k=1}^{d}\sum_{r=1}^{m}(\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \\
&\cdot \alpha_i(\tau)^{-1}) \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq\ & \frac{\exp(2B)}{m} \cdot |\sum_{i=1}^{n}\sum_{k=1}^{d}\sum_{r=1}^{m}(\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \\
&\cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq\ & \frac{\exp(2B)}{m} \cdot |\sum_{i=1}^{n}\sum_{k=1}^{d}\sum_{r=1}^{m}(W_{k,r}(\tau+1) \cdot a_r \\
&- W_{k,r}(\tau) \cdot a_r) \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq\ & \eta\frac{\exp(2B)}{m} \cdot |\sum_{i=1}^{n}\sum_{k=1}^{d}\sum_{r=1}^{m} a_r \cdot m \\
&\cdot \sum_{j=1}^{n}\sum_{k_1=1}^{d}(\mathsf{F}_{k_1,j}(\tau) - y_{k_1,j}) \\
&\cdot \Big(\langle v_{k_1,r}(\tau), \mathsf{S}_j(\tau)\rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_{j,k} \\
&+ a_r \mathsf{S}_{j,r}(\tau) e_{k_1,k}\Big) \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq\ & \eta\frac{\exp(5B)}{m} \cdot |\sum_{r=1}^{m}\sigma_r \max_{j,k,k_1\in[d]} C_{j,k,k_1,r}| \\
&\cdot \|(\mathsf{F}(\tau) - Y) \otimes (\mathsf{F}(\tau) - Y)\|_1 \\
\leq\ & \eta\frac{\exp(5B)}{m} \cdot |\sum_{r=1}^{m}\sigma_r \max_{j,k,k_1\in[d]} C_{j,k,k_1,r}| \\
&\cdot \|\mathsf{F}(\tau) - Y\|_1^2 \\
\leq\ & \eta\frac{nd\exp(5B)}{m} \cdot |\sum_{r=1}^{m}\sigma_r \max_{j,k,k_1\in[d]} C_{j,k,k_1,r}| \\
&\cdot \|\mathsf{F}(\tau) - Y\|_F^2
\end{aligned}
$$

where the first step follows from the definition of $Q_{0,2,k,i}$, the second and third steps follow from Part 4 of Lemma M.1, the fourth step follows from Definition G.5, the fifth step follows from Eq.(12), the sixth step follows from the definition of Kronecker product, $1 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, $\|x_i\|_2 \leq 1$ and defining

$$
C_{j,k,k_1,r} := \langle \mathsf{S}_j, v_{k_1,r}\rangle + e_{k_1,k}, \sigma_r \in \{+1,-1\},
$$

the seventh step follows from the definition of ℓ_1 norm, the last step follows from $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n\times d}$.

Thus, by following Part 6 of Lemma M.2, we have

$$
\begin{aligned}
C_{j,k,k_1,r} &= \langle \mathsf{S}_j, v_{k_1,r}\rangle + e_{k_1,k} \\
&\leq \exp(6B) + 1 \\
&\leq \exp(7B)
\end{aligned}
$$

where the last step follows from simple algebras.

We apply Hoeffding inequality (Lemma F.4) to $\sigma_r \max_{j,k,k_1\in[d]} C_{j,k,k_1,r}$ for $r \in [m]$.

By $\mathbb{E}[\sum_{r=1}^m \sigma_r \max_{j,k,k_1 \in [d]} C_{j,k,k_1,r}] = 0$, we have

$$\begin{aligned}
&|\sum_{i=1}^n \sum_{k=1}^d Q_{0,2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta \frac{nd \exp(5B)}{m} \\
&\cdot \|\mathsf{F}(\tau) - Y\|_F^2 \cdot \exp(6B)\sqrt{m \log(nd/\delta)}.
\end{aligned}$$

with probability at least $1 - \delta/\operatorname{poly}(nd)$.

Then, by Lemma condition, we have

$$\eta \frac{nd \exp(5B)}{m} \cdot \exp(7B)\sqrt{m \log(nd/\delta)} \leq 0.01\eta\lambda.$$

Now we can complete the proof by combining all terms, we have

$$|C_0| \leq 0.1\eta m \lambda \|\mathsf{F}(\tau) - Y\|_F^2.$$

□

J.2 Bounding $C_{1,2}$

Lemma J.5. *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$
- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B)\sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_1 \in [d]$.
- *Let* $\beta_k(\tau) \in \mathbb{R}^m$ *be defined as Definition G.5.*
- *Let* $\alpha_i(\tau) \in \mathbb{R}$ *be defined as Definition G.3.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_k := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*
- *Let* $\eta > 0$ *denote the learning rate.*
- *Let scalar* $v_{1,2,k,i} \in \mathbb{R}$ *be defined as follows (we omit* (τ) *in the following terms)*

$$\begin{aligned}
&v_{1,2,k,i} \\
=& m^2 \sum_{r \in [m]} \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \\
&\cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\
&\cdot a_r \mathsf{S}_{j,r}(\tau) e_{k_2}^\top) x_i
\end{aligned}$$

- *Let* $C_{1,2} := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_{1,2})\rangle$

Then, with a probability at least $1 - \delta/\operatorname{poly}(nd)$*, we have*

$$|C_{1,2}| \leq 0.1\eta m\lambda \|\mathsf{F}(\tau) - Y\|_F^2$$

Proof. We have the quantity of $v_{1,2,k,i}$

$$\begin{aligned}
&|\sum_{i=1}^n \sum_{k=1}^d v_{1,2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq &|\sum_{i=1}^n \sum_{k=1}^d m^2 \sum_{r=1}^m \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \\
&\cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\
&\cdot a_r \mathsf{S}_{j,r}(\tau) e_{k_2}^\top) x_i \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq &|\sum_{i=1}^n \sum_{k=1}^d m^2 \sum_{r=1}^m \beta_{k,r}(\tau)\alpha_i(\tau)^{-1} \cdot \mathsf{u}_{i,r}(\tau) \\
&\cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\
&\cdot a_r \mathsf{S}_{j,r}(\tau) e_{k_2}^\top) x_i \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq &|\sum_{i=1}^n \sum_{k=1}^d m^2 \sum_{r=1}^m \beta_{k,r}(\tau)\mathsf{S}_{i,r}(\tau) \\
&\cdot (-\eta \sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\
&\cdot a_r \mathsf{S}_{j,r}(\tau) e_{k_2}^\top) x_i \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq &\eta m^2 |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m \beta_{k,r}(\tau)\mathsf{S}_{i,r}(\tau) \\
&\cdot (-\sum_{j=1}^n \sum_{k_2=1}^d (\mathsf{F}_{k_2,j}(\tau) - y_{k_2,j}) \\
&\cdot a_r \mathsf{S}_{j,r}(\tau) e_{k_2}^\top) x_i \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq &\eta \exp(6B) \sum_{r=1}^m |a_r \cdot \max_{k\in[d]} \beta_{k,r}(\tau)| \\
&\cdot \|(\mathsf{F}(\tau) - Y) \otimes (\mathsf{F}(\tau) - Y)\|_1 \\
\leq &\eta \exp(6B) \sum_{r=1}^m |a_r \cdot \max_{k\in[d]} \beta_{k,r}(\tau)| \\
&\cdot \|\mathsf{F}(\tau) - Y\|_1^2 \\
\leq &\eta nd \exp(6B) \sum_{r=1}^m |a_r \cdot \max_{k\in[d]} \beta_{k,r}(\tau)| \\
&\cdot \|\mathsf{F}(\tau) - Y\|_F^2
\end{aligned}$$

where the first step follows from the definition of $v_{1,2,k,i}$, the second step follows from Definition H.6, the third step follows from Definition G.5, the fourth step follows from Definition H.7, the fifth step follows from simple algebras, the sixth step follows from $0 \leq \mathsf{S}_{j,r} \leq \frac{\exp(3B)}{m}$, $\|x_i\|_2 \leq 1$ and the definition of Kronecker product, the seventh step follows from the definition of ℓ_1 norm, the last step follows from $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n \times d}$.

Then by Part 1 of Lemma M.1, we have

$$|\max_{k \in [d]} \beta_{k,r}(\tau)| \leq B$$

We apply Hoeffding inequality (Lemma F.4) to random variables $a_r \cdot \max_{k \in [d]} \beta_{k,r}(\tau)$ for $r \in [m]$. By $\mathbb{E}[\sum_{r=1}^m a_r \cdot \max_{k \in [d]} \beta_{k,r}(\tau)] = 0$, we have

$$\begin{aligned} & |\sum_{i=1}^n \sum_{k=1}^d v_{1,2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\ \leq & \eta nd \exp(6B) B \|\mathsf{F}(\tau) - Y\|_F^2 \end{aligned}$$

with a probability at least $1 - \delta / \operatorname{poly}(nd)$.

By the Lemma condition, we have

$$nd \exp(6B) B \leq 0.1 m \lambda$$

□

J.3 Bounding C_2

Lemma J.6. *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$
- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B)\sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_1 \in [d]$.
- *Let* $\beta_k(\tau) \in \mathbb{R}^m$ *be defined as Definition G.5.*
- *Let* $\alpha_i(\tau) \in \mathbb{R}$ *be defined as Definition G.3.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_k := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*
- *Let* $\eta > 0$ *denote the learning rate.*
- *Let scalar* $v_{2,k,i} \in \mathbb{R}$ *be defined as follows (we omit* (τ) *in the following terms)*

$$\begin{aligned} v_{2,k,i} := & m \sum_{r=1}^m \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \cdot \eta^2 \\ & \cdot \Theta(1) \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \end{aligned}$$

- *Let* $C_2 := 2\langle \mathrm{vec}(\mathsf{F}(\tau) - Y), \mathrm{vec}(v_2)\rangle$

Then, with a probability at least $1 - \delta/\operatorname{poly}(nd)$*, we have*

$$|C_2| \leq \eta^2 m \cdot n^2 d^2 \exp(16B) \|\mathsf{F}(\tau) - Y\|_F^2$$

Proof. We have

$$\begin{aligned}
&\langle \Delta w_r(\tau), x_i \rangle^2 \\
\leq & \Big(m \sum_{j=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\
& \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_j(\tau) \rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_j^\top \\
& + a_r \mathsf{S}_{j,r}(\tau) e_k^\top \Big) x_i \Big)^2 \\
\leq & \exp(12B) \cdot \|\mathsf{F}(\tau) - Y\|_1^2 \\
\leq & nd \exp(12B) \cdot \|\mathsf{F}(\tau) - Y\|_F^2 \qquad (15)
\end{aligned}$$

where the first step follows from Claim H.12, the second step follows from the definition of ℓ_1 norm, $0 \leq \mathsf{S}_{j,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1 and Part 6 of Lemma M.2, last step follows from $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n \times d}$.

Then, we can show that

$$\begin{aligned}
& |\sum_{i=1}^n \sum_{k=1}^d v_{2,k,i} (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & |\sum_{i=1}^n \sum_{k=1}^d m \sum_{r=1}^m \theta_{k,i,r}(\tau) \cdot \mathsf{u}_{i,r}(\tau) \cdot \eta^2 \\
& \cdot \Theta(1) \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta^2 |\sum_{i=1}^n \sum_{k=1}^d m \sum_{r=1}^m \theta_{k,i,r}(\tau) \\
& \cdot \mathsf{u}_{i,r}(\tau) \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta^2 |\sum_{i=1}^n \sum_{k=1}^d m \sum_{r=1}^m \beta_{k,r}(\tau) \cdot \alpha_i(\tau)^{-1} \\
& \cdot \mathsf{u}_{i,r}(\tau) \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta^2 |\sum_{i=1}^n \sum_{k=1}^d m \sum_{r=1}^m \beta_{k,r}(\tau) \cdot \mathsf{S}_{i,r}(\tau) \\
& \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta^2 \exp(3B) |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m \beta_{k,r}(\tau) \\
& \cdot \langle \Delta w_r(\tau), x_i \rangle^2 \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta^2 \exp(4B) |\sum_{i=1}^n \sum_{k=1}^d \sum_{r=1}^m a_r \langle \Delta w_r(\tau), x_i \rangle^2 \\
& \cdot (\mathsf{F}_{k,i}(\tau) - y_{k,i})| \\
\leq & \eta^2 \exp(4B) |\sum_{r=1}^m a_r \max_{i \in [n]} \langle \Delta w_r(\tau), x_i \rangle^2|
\end{aligned}$$

$$\cdot \sqrt{nd} \|\mathsf{F}(\tau) - Y\|_F$$
$$\leq \eta^2 \sqrt{m} nd \exp(4B) |\sum_{r=1}^m a_r \max_{i\in[n]} \langle \Delta w_r(\tau), x_i \rangle^2|$$

where the first step follows from the definition of $v_{2,k,i}$, the second step follows from simple algebras, the third step follows from Definition H.6, the fourth step follows from Definition H.7, the fifth step follows from $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, the sixth step follows from Part 1 of Lemma M.1 and Definition G.5, the seventh step follows from definition of ℓ_1 norm and $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n\times d}$, the last step follows from Lemma J.8.

Next, by Eq.(15), applying Hoeffding inequality (Lemma F.4) to $a_r \max_{i\in[n]} \langle \Delta w_r(\tau), x_i\rangle^2$ for $r \in [m]$ and $\mathbb{E}[\sum_{r=1}^m a_r \max_{i\in[n]} \langle \Delta w_r(\tau), x_i\rangle^2] = 0$, we have

$$|\sum_{i=1}^n \sum_{k=1}^d v_{2,k,i}(\mathsf{F}_{k,i}(\tau) - y_{k,i})|$$
$$\leq \eta^2 \sqrt{m} n^2 d^2 \exp(16B) \cdot \|\mathsf{F}(\tau) - Y\|_F^2$$
$$\cdot \sqrt{m \log(nd/\delta)}$$

with a probability at least $1 - \delta/\operatorname{poly}(nd)$.

By the Lemma condition, we have

$$\eta^2 \sqrt{m} n^2 d^2 \exp(16B) \cdot \sqrt{m \log(nd/\delta)}$$
$$\leq \eta^2 m \cdot n^2 d^2 \exp(16B)$$

Then we complete the proof. □

J.4 Bounding C_3

Lemma J.7. *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$
- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B)\sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_1 \in [d]$.
- *Let* $\beta_k(\tau) \in \mathbb{R}^m$ *be defined as Definition G.5.*
- *Let* $\alpha_i(\tau) \in \mathbb{R}$ *be defined as Definition G.3.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_k := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n\times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n\times d}$ *denote the labels.*

- *Let $\eta > 0$ denote the learning rate.*
- *Let $C_3 := \|\mathsf{F}(\tau+1) - \mathsf{F}(\tau)\|_F^2$*

Then, with a probability at least $1 - \delta/\operatorname{poly}(nd)$, we have

$$|C_3| \leq \eta^2 m^2 \|\mathsf{F}(\tau) - Y\|_F^2$$

Proof. We have

$$\begin{aligned}
& |C_3| \\
&= \|\mathsf{F}(\tau+1) - \mathsf{F}(\tau)\|_F^2 \\
&= \sum_{i=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau+1) - \mathsf{F}_{k,i}(\tau))^2 \\
&= \sum_{i=1}^n \sum_{k=1}^d m^2 (\langle \beta_k(\tau+1), \mathsf{S}_i(\tau+1)\rangle \\
&\quad - \langle \beta_k(\tau), \mathsf{S}_i(\tau)\rangle)^2 \\
&= \sum_{i=1}^n \sum_{k=1}^d m^2 \Big(\sum_{r=1}^m (\beta_{k,r}(\tau+1) \cdot \mathsf{S}_{i,r}(\tau+1) \\
&\quad - \beta_{k,r}(\tau) \cdot \mathsf{S}_{i,r}(\tau)) \Big)^2 \\
&= \sum_{i=1}^n \sum_{k=1}^d m^2 \Big(\sum_{r=1}^m (\beta_{k,r}(\tau+1) \cdot \mathsf{S}_{i,r}(\tau+1) \\
&\quad - \beta_{k,r}(\tau+1) \cdot \mathsf{S}_{i,r}(\tau) \\
&\quad + \beta_{k,r}(\tau+1) \cdot \mathsf{S}_{i,r}(\tau) - \beta_{k,r}(\tau) \cdot \mathsf{S}_{i,r}(\tau)) \Big)^2 \\
&= \sum_{i=1}^n \sum_{k=1}^d m^2 \Big(\sum_{r=1}^m (\beta_{k,r}(\tau+1) \cdot (\mathsf{S}_{i,r}(\tau+1) \\
&\quad - \mathsf{S}_{i,r}(\tau)) + (\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \cdot \mathsf{S}_{i,r}(\tau)) \Big)^2 \\
&= \sum_{i=1}^n \sum_{k=1}^d m^2 (Q_{3,1,i,k} + Q_{3,2,i,k})^2
\end{aligned}$$

where the first step follows from the definition C_2, the second step follows from the definition of Frobenius norm, the third step follows from Definition H.8, the fourth, fifth and sixth steps follow from simple algebras, the last step follows from defining

$$\begin{aligned}
Q_{3,1,i,k} &= \sum_{r=1}^m \beta_{k,r}(\tau+1) \cdot (\mathsf{S}_{i,r}(\tau+1) - \mathsf{S}_{i,r}(\tau)), \\
Q_{3,2,i,k} &= \sum_{r=1}^m (\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \cdot \mathsf{S}_{i,r}(\tau).
\end{aligned}$$

Bounding first term. For the first term, we have

$$\begin{aligned}
& |Q_{3,1,i,k}| \\
&= |\sum_{r=1}^m \beta_{k,r}(\tau+1) \cdot (\mathsf{S}_{i,r}(\tau+1) - \mathsf{S}_{i,r}(\tau))|
\end{aligned}$$

$$
\begin{aligned}
&= |\sum_{r=1}^m a_r \cdot w_{r,k}(\tau+1) \cdot (\mathsf{S}_{i,r}(\tau+1) - \mathsf{S}_{i,r}(\tau))| \\
&\leq |B \cdot \sum_{r=1}^m a_r \cdot (\mathsf{S}_{i,r}(\tau+1) - \mathsf{S}_{i,r}(\tau))| \\
&\leq |\exp(3B) \cdot \sum_{r=1}^m a_r \cdot \max_{i\in[n]}(\alpha_i(\tau+1)^{-1} \\
&\quad - \alpha_i(\tau)^{-1})|
\end{aligned}
$$

where the first step follows from the definition of $Q_{3,1,i,k}$, the second step follows from Definition G.5, the third step follows from Part 1 of Lemma M.1, last step follows from Part 4 of Lemma M.1, Definition H.7 and $B \leq \exp(B)$.

Then by Part 2 of Lemma J.9, applying Hoeffding inequality (Lemma F.4) to the random variables $a_r \cdot \max_{i\in[n]}(\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1}$ for $r \in [m]$ and $\mathbb{E}[\sum_{r=1}^m a_r \cdot \max_{i\in[n]}(\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1}] = 0$, we have

$$
\begin{aligned}
&|Q_{3,1,i,k}| \\
&\leq (\eta \frac{\sqrt{nd}\exp(18B)}{m^3} \cdot \|\mathsf{F}(\tau) - Y\|_F \\
&\quad + \eta^2 \frac{nd\exp(30B)}{\sqrt{m}} \cdot \|\mathsf{F}(\tau) - Y\|_F) \\
&\quad \cdot \sqrt{m\log(nd/\delta)}
\end{aligned}
$$

with a probability of at least $1 - \delta/\operatorname{poly}(nd)$.

By the Lemma condition, we have

$$
\begin{aligned}
&(\eta \frac{\sqrt{nd}\exp(18B)}{m^3} + \eta^2 \frac{nd\exp(30B)}{\sqrt{m}}) \\
&\cdot \sqrt{m\log(nd/\delta)} \leq \frac{1}{2\sqrt{nd}}\eta
\end{aligned}
$$

Bounding second term. On the other hand, for the second term $Q_{3,2,k,i}$, we have

$$
\begin{aligned}
&|Q_{3,2,k,i}| \\
&= |\sum_{r=1}^m (\beta_{k,r}(\tau+1) - \beta_{k,r}(\tau)) \cdot \mathsf{S}_{i,r}(\tau)| \\
&= \eta|\sum_{r=1}^m a_r \Delta w_{r,k}(\tau) \cdot \mathsf{S}_{i,r}(\tau)| \\
&\leq \eta \frac{\exp(3B)}{m} |\sum_{r=1}^m a_r \Delta w_{r,k}(\tau)| \\
&\leq \eta \exp(3B) \Big| \sum_{r=1}^m a_r \sum_{j=1}^n \sum_{k_1=1}^d (\mathsf{F}_{k_1,j}(\tau) \\
&\quad - y_{k_1,j}) \cdot \Big(\langle v_{k_1,r}(\tau), \mathsf{S}_j(\tau) \rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_{i,k} \\
&\quad + a_r \mathsf{S}_{j,r}(\tau) e_{k,k_1} \Big) \Big| \\
&\leq \eta \frac{\exp(6B)}{m} |\sum_{r=1}^m a_r \max_{j\in[n],k,k_1\in[d]} C_{j,k,k_1,r}|
\end{aligned}
$$

$$
\begin{aligned}
&\cdot \|\mathsf{F}(\tau) - Y\|_1 \\
\leq\ & \eta \frac{\sqrt{nd}\exp(6B)}{m} |\sum_{r=1}^m a_r \max_{j \in [n], k, k_1 \in [d]} C_{j,k,k_1,r}| \\
&\cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}
$$

where the first step follows from the definition of $Q_{3,2,k,i}$, the second step follows from Definition H.13, the third step follows from $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, the fourth step follows from Claim H.12, the fifth step follows from $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, $\|x_i\|_2 \leq 1$ and defining

$$
C_{j,k,k_1,r} := \langle v_{k_1,r}(\tau), \mathsf{S}_j(\tau) \rangle + e_{k,k_1},
$$

the last step follows from $\|U\|_1 \leq \sqrt{nd}\|U\|_F$ for $U \in \mathbb{R}^{n \times d}$.

Now we follow from Part 6 of Lemma M.2, applying Hoeffding inequality (Lemma F.4) to random variables $a_r \max_{j \in [n], k, k_1 \in [d]} C_{j,k,k_1,r}$ for $r \in [m]$ and $\mathbb{E}[\sum_{r=1}^m a_r \max_{j \in [n], k, k_1 \in [d]} C_{j,k,k_1,r}] = 0$, we have

$$
\begin{aligned}
& |Q_{3,2,k,i}| \\
\leq\ & \eta \frac{\sqrt{nd}\exp(13B)}{m} \cdot \|\mathsf{F}(\tau) - Y\|_F \\
& \cdot \sqrt{m \log(nd/\delta)} \\
\leq\ & \frac{1}{2\sqrt{nd}} \eta
\end{aligned}
$$

Finally, we combine all terms, we have

$$
\begin{aligned}
& |C_3| \\
=\ & \sum_{i=1}^n \sum_{k=1}^d m^2 ((\frac{1}{2\sqrt{nd}}\eta + \frac{1}{2\sqrt{nd}}\eta) \\
& \cdot \|\mathsf{F}(\tau) - Y\|_F)^2 \\
\leq\ & \eta^2 m^2 \|\mathsf{F}(\tau) - Y\|_F^2
\end{aligned}
$$

□

J.5 Bounding Loss during Training Process

Lemma J.8. *If the following conditions hold*

- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*

Then we have

$$
\|\mathsf{F}(\tau) - Y\|_F \leq O(\sqrt{nmd})
$$

Proof. This proof follows from $\|y_i\| \leq 1$ for $i \in [n]$ and Definition H.8. □

J.6 Helpful Lemma

Lemma J.9. *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$.
- *Let* $C > 10$ *denote a sufficiently large constant.*

- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B)\sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$, *let* $i, j \in [n]$, *let* $k, k_1 \in [d]$.
- *Let* $\alpha_i(\tau) \in \mathbb{R}$ *be defined as Definition G.3.*
- *Let* $\beta_k(\tau) \in \mathbb{R}^m$ *be defined as Definition G.5.*
- *Let* $\theta_{k,i}(\tau) \in \mathbb{R}^m$ *be defined as Definition H.6.*
- *Let* $\mathsf{u}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.2.*
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_k := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$.
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n\times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n\times d}$ *denote the labels.*

Then with a probability at least $1 - \delta/\operatorname{poly}(nd)$*, we have*

- *Part 1.*

$$\begin{aligned}&\alpha_i(\tau+1) - \alpha_i(\tau)\\ \leq\ &\eta\frac{\sqrt{nd}\exp(9B)}{m}\cdot\|\mathsf{F}(\tau) - Y\|_F\\ &+\eta^2 m^{1.5}\cdot nd\exp(21B)\cdot\|\mathsf{F}(\tau) - Y\|_F\end{aligned}$$

- *Part 2.*

$$\begin{aligned}&\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1}\\ \leq\ &\eta\frac{\sqrt{nd}\exp(15B)}{m^3}\cdot\|\mathsf{F}(\tau) - Y\|_F\\ &+\eta^2\frac{nd\exp(27B)}{\sqrt{m}}\cdot\|\mathsf{F}(\tau) - Y\|_F\end{aligned}$$

Proof. **Proof of Part 1.**

We have

$$\begin{aligned}&\alpha_i(\tau+1) - \alpha_i(\tau)\\ =\ &\langle \mathsf{u}_i(\tau+1), \mathbf{1}_m\rangle - \langle \mathsf{u}_i(\tau), \mathbf{1}_m\rangle\\ =\ &\langle \mathsf{u}_i(\tau+1) - \mathsf{u}_i(\tau), \mathbf{1}_m\rangle\\ =\ &\langle \exp(W(\tau+1)^\top x_i) - \exp(W(\tau)^\top x_i), \mathbf{1}_m\rangle\\ =\ &\langle \exp(W(\tau)^\top x_i)\\ &\circ(\exp(-\eta\Delta W(\tau)^\top x_i) - \mathbf{1}_m), \mathbf{1}_m\rangle\\ =\ &\langle \exp(W(\tau)^\top x_i)\circ(-\eta\Delta W(\tau)^\top x_i + \Theta(1)\eta^2\\ &\cdot(\Delta W(\tau)^\top x_i)^2), \mathbf{1}_m\rangle\\ =\ &\langle -\eta\Delta W(\tau)^\top x_i + \Theta(1)\eta^2\cdot(\Delta W(\tau)^\top x_i)^2,\\ &\exp(W(\tau)^\top x_i)\rangle\end{aligned}$$

$$\leq \exp(B) \cdot \langle -\eta \Delta W(\tau)^\top x_i + \Theta(1)\eta^2 \cdot (\Delta W(\tau)^\top x_i)^2, \mathbf{1}_m) \rangle$$
$$\leq \eta \frac{\sqrt{nd} \exp(9B)}{m} \cdot \|\mathsf{F}(\tau) - Y\|_F + \eta^2 m^{1.5} \cdot nd \exp(21B) \cdot \|\mathsf{F}(\tau) - Y\|_F$$

where the first step follows from Definition G.3, the second step follows from simple algebras, the third step follows from Definition H.2, the fourth step follows from simple algebra, the fifth step follows from Fact F.1, the sixth step follows from simple algebras, the seventh step follows from Part 4 of Lemma M.1, last step follows from Part 1 and Part 2 of Lemma J.10.

Proof of Part 2. We have

$$\begin{aligned}
&\alpha_i(\tau+1)^{-1} - \alpha_i(\tau)^{-1} \\
&= \alpha_i(\tau+1)^{-1}\alpha_i(\tau)^{-1} \cdot (\alpha_i(\tau+1) - \alpha_i(\tau)) \\
&\leq \frac{\exp(6B)}{m^2} \cdot (\alpha_i(\tau+1) - \alpha_i(\tau)) \\
&\leq \eta \frac{\sqrt{nd}\exp(15B)}{m^3} \cdot \|\mathsf{F}(\tau) - Y\|_F \\
&\quad + \eta^2 \frac{nd \exp(27B)}{\sqrt{m}} \cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}$$

where the first step follows from simple algebras, the second step follows from Part 4 of Lemma M.2, the last step follows from Part 1 of this Lemma. □

Lemma J.10. *If the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*)$.
- *Let* $W(\tau) \in \mathbb{R}^{m \times d}$ *be defined as Definition H.13, let* $a \in \mathbb{R}^m$ *be defined as Definition G.1.*
- *Let* $C > 10$ *denote a sufficiently large constant.*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $\delta \in (0, 0.1)$.
- *Let* $m \geq \Omega(\lambda^{-2} n^2 d^2 \exp(30B)\sqrt{\log(nd/\delta)})$.
- *Let* $r \in [m]$*, let* $i, j \in [n]$*, let* $k, k_2 \in [d]$.
- *Let* $\mathsf{S}_i(\tau) \in \mathbb{R}^m$ *be defined as Definition H.7.*
- *Let* $v_{k,r} := \beta_{k,r}(\tau) \cdot \mathbf{1}_m - \beta_k(\tau) \in \mathbb{R}^m$.
- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*
- *Let* $\eta = \lambda / (m \cdot \mathrm{poly}(n, d, \exp(B)))$ *denote the learning rate.*

Then with a probability at least $1 - \delta/\mathrm{poly}(nd)$*, we have*

- *Part 1.*

$$\begin{aligned}
&|\langle \eta \Delta W(\tau)^\top x_i, \mathbf{1}_m \rangle| \\
&\leq \eta \frac{\sqrt{nd}\exp(8B)}{m} \cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}$$

- *Part 2.*

$$
\begin{aligned}
&|\langle \eta^2 (\Delta W(\tau)^\top x_i)^2, \mathbf{1}_m \rangle| \\
&\le \eta^2 m^{1.5} \cdot nd \exp(20B) \cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}
$$

Proof. **Proof of Part 1.** We have

$$
\begin{aligned}
&|\langle \eta \Delta W(\tau)^\top x_i, \mathbf{1}_m \rangle| \\
&= \eta |\sum_{r=1}^m \langle \Delta w_r(\tau), x_i \rangle| \\
&\le \eta \Big| \sum_{r=1}^m m \sum_{j=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\
&\quad \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_j(\tau) \rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_j^\top \\
&\quad + a_r \mathsf{S}_{j,r}(\tau) e_k^\top \Big) x_i \Big| \\
&\le \eta \Big| \sum_{r=1}^m m \sum_{j=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \cdot \Big(\langle \beta_{k,r}(\tau) \cdot \mathbf{1}_m \\
&\quad - \beta_k(\tau), \mathsf{S}_j(\tau) \rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_j^\top \\
&\quad + a_r \mathsf{S}_{j,r}(\tau) e_k^\top \Big) x_i \Big| \\
&\le \eta \Big| \sum_{r=1}^m m \sum_{j=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \cdot \Big(a_r w_{r,k} \\
&\quad + \langle -a \circ W_{k,*}(\tau), \mathsf{S}_j(\tau) \rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_j^\top \\
&\quad + a_r \mathsf{S}_{j,r}(\tau) e_k^\top \Big) x_i \Big| \\
&\le \eta \frac{\exp(3B)}{m} \sum_{r=1}^m \sigma_r \max_{j \in [n], k \in [d]} C_{j,k,r} \\
&\quad \cdot \|\mathsf{F}(\tau) - Y\|_1 \\
&\le \eta \frac{\sqrt{nd} \exp(3B)}{m} \sum_{r=1}^m \sigma_r \max_{j \in [n], k \in [d]} C_{j,k,r} \\
&\quad \cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}
$$

where the first step follows from simple algebras, the second step follows from Claim H.12, the third step follows from the definition of $v_{k,r}$, the fourth step follows from Definition G.5 and simple algebras, the fifth step follows from $\|x_i\|_2 \le 1$, $1 \le \mathsf{S}_{i,r} \le \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, definition of ℓ_1 norm and defining

$$
\begin{aligned}
&C_{j,k,r} \\
&:= |w_{r,k}| + |\langle -W_{k,*}(\tau), \mathsf{S}_j(\tau) \rangle| \\
&\quad + \|e_k\|, \sigma_r \in \{+1, -1\},
\end{aligned}
$$

the last step follows from $\|U\|_1 \le \sqrt{nd} \|U\|_F$ for $U \in \mathbb{R}^{n \times d}$.

Thus, by following Part 1 and Part 11 of Lemma M.2 and Hoeffding inequality (Lemma F.4), we have

$$
\begin{aligned}
&|\langle \eta \Delta W(\tau)^\top x_i, \mathbf{1}_m \rangle| \\
&\le \eta \frac{\sqrt{nd} \exp(8B)}{m} \cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}
$$

with a probability at least $1 - \delta/\operatorname{poly}(nd)$.

Proof of Part 2. We have

$$\begin{aligned}
& |\langle \eta^2 (\Delta W(\tau)^\top x_i)^2, \mathbf{1}_m \rangle| \\
\leq & \eta^2 \sum_{r=1}^m (\langle \Delta w_r(\tau), x_i \rangle)^2 \\
\leq & \eta^2 \sum_{r=1}^m \Big(m \sum_{j=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) - y_{k,i}) \\
& \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_j(\tau) \rangle \cdot \mathsf{S}_{j,r}(\tau) \cdot x_j^\top \\
& + a_r \mathsf{S}_{j,r}(\tau) e_k^\top \Big) x_i \Big)^2 \\
\leq & \eta^2 \exp(6B) \sum_{r=1}^m \Big(\sum_{j=1}^n \sum_{k=1}^d (\mathsf{F}_{k,i}(\tau) \\
& - y_{k,i}) \cdot \Big(\langle v_{k,r}(\tau), \mathsf{S}_j(\tau) \rangle \cdot x_j^\top + a_r e_k^\top \Big) x_i \Big)^2 \\
\leq & \eta^2 m \exp(20B) \cdot \|\mathsf{F}(\tau) - Y\|_1^2 \\
\leq & \eta^2 m \sqrt{nmd} \exp(20B) \cdot \|\mathsf{F}(\tau) - Y\|_1 \\
\leq & \eta^2 m^{1.5} \cdot nd \exp(20B) \cdot \|\mathsf{F}(\tau) - Y\|_F
\end{aligned}$$

where the first step follows from simple algebras, the second step follows from Claim H.12, the third step follows from $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, the fourth step follows from $\langle v_{k,r}(\tau), \mathsf{S}_j(\tau) \rangle \leq \exp(6B)$ by Part 6 of Lemma M.2, $\|x_i\|_2 \leq 1$, $\exp(6B) + 1 \leq \exp(7B)$ and the definition of ℓ_1 norm, the fifth step follows from Lemma J.8, the last step follows from $\|U\|_1 \leq \|U\|_F$ for $U \in \mathbb{R}^{n \times d}$. $\square$

K Convergence of Prefix Learning

Here, we provide all the properties we need for math induction for NTK happening.

Definition K.1 (Properties)**.** *We state the following properties*

- *General Condition 1. Let* $\lambda = \lambda_{\min}(H^*) > 0$
- *General Condition 2. Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *General Condition 3. Let* η *be defined as*

$$\eta := \lambda/(m \operatorname{poly}(n, d, \exp(B))).$$

- *General Condition 4. Let* $D := 2\lambda^{-1} \cdot \exp(20B) \frac{\sqrt{nd}}{m} \|Y - \mathsf{F}(0)\|_F$
- *General Condition 5. Let* w_r *and* a_r *be defined as Definition G.1.*
- *General Condition 6.* $D < R = \lambda/\operatorname{poly}(n, d, \exp(B))$
- *General Condition 7.* $m = \lambda^{-2} \operatorname{poly}(n, d, \exp(B))$
- **Weight Condition.** $\|w_r(t) - w_r(0)\|_2 \leq D < R$, $\forall r \in [m]$
- **Loss Condition.** $\|\operatorname{vec}(\mathsf{F}(i) - Y)\|_2^2 \leq \|\operatorname{vec}(\mathsf{F}(0) - Y)\|_2^2 \cdot (1 - m\eta\lambda/2)^i$, $\forall i \in [t]$
- **Gradient Condition.** $\eta \|\Delta w_r(i)\|_2 \leq 0.01$ $\forall r \in [m]$, $\forall i \in [t]$

K.1 Main Result

Our main result is presented as follows.

Theorem K.2 (Main result, formal version of Theorem 4.2)**.** *For any* $\epsilon, \delta \in (0, 0.1)$*, if the following conditions hold*

- *Let* $\lambda = \lambda_{\min}(H^*) > 0$
- *Let* $B = \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$
- *Let* $m = \lambda^{-2} \operatorname{poly}(n, d, \exp(B))$
- *Let* $\eta = \lambda/(m \operatorname{poly}(n, d, \exp(B)))$
- *Let* $\widehat{T} = \Omega((m\eta\lambda)^{-1} \log(nd/\epsilon))$

Then, after $\widehat{T}$ *iterations, with probability at least* $1 - \delta$*, we have*

$$\|\mathsf{F}(\widehat{T}) - Y\|_F^2 \leq \epsilon.$$

Proof. We have $\|\mathsf{F}(0) - Y\|_F^2 \leq nd$ as Lemma K.6. Using the choice of $\widehat{T}$, it follows directly from the alternative application of Lemma K.3 and Lemma K.4. □

K.2 Induction Part 1. For Weights

In this section, we introduce the induction lemma for weights.

Lemma K.3 (Induction Part 1 for weights)**.** *If the following conditions hold*

- *Suppose properties in Definition K.1 are true*

For $t + 1$ *and* $\forall r \in [m]$*, it holds that:*

$$\|w_r(t+1) - w_r(0)\|_2 \leq D.$$

Proof. We have

$$\eta \sum_{i=0}^{\infty} (1 - m\eta\lambda/2)^i \leq \eta \frac{4}{m\lambda} \tag{16}$$

where this step follows from Fact F.2.

$$\begin{aligned}
& \|w_r(t+1) - w_r(0)\|_2 \\
\leq & \eta \sum_{\tau=0}^{t} \|\Delta w_r(\tau)\|_2 \\
\leq & \eta \sum_{\tau=0}^{t} \sqrt{n} d \exp(11B) \cdot \|\mathsf{F}(t) - Y\|_F \\
\leq & \eta \sqrt{n} d \exp(11B) \cdot \sum_{\tau=0}^{t} (1 - m\eta\lambda/2)^i \\
& \cdot \|\mathsf{F}(0) - Y\|_F \\
\leq & 2\eta \frac{1}{m\lambda} \sqrt{n} d \exp(11B) \cdot \|\mathsf{F}(0) - Y\|_F \\
\leq & D
\end{aligned}$$

where the third step follows from the triangle inequality, the second step follows from Eq. (20), the third step follows from Lemma K.4, the fourth step follows from Eq. (16), the last step follows from *General Condition 4.* in Definition K.1.

□

K.3 Induction Part 2. For Loss

Now, we present our next induction lemma.

Lemma K.4 (Induction Part 2 for loss)**.** *Let t be a fixed integer.*

If the following conditions hold

- *Suppose properties in Definition K.1 are true*

Then we have

$$\|\mathsf{F}(t+1) - y\|_F^2 \leq (1 - m\eta\lambda/2)^{t+1} \cdot \|\mathsf{F}(0) - y\|_F^2.$$

Proof. We have

$$\begin{aligned}
& \|\mathsf{F}(t+1) - y\|_F^2 \\
\leq & \|\mathsf{F}(t) - y\|_F^2 + C_0 + C_1 + C_2 + C_3 \\
= & \|\mathsf{F}(t) - y\|_F^2 + C_0 + C_{1,1} + C_{1,2} + C_2 + C_3 \\
\leq & \|\mathsf{F}(t) - y\|_F^2 \cdot (1 + 0.1\eta m\lambda - 1.6\eta m\lambda \\
& + 0.1\eta m\lambda + \eta^2 m \cdot n^2 d^2 \exp(16B) + \eta^2 m^2) \\
\leq & \|\mathsf{F}(t) - y\|_F^2 \cdot (1 - 1.4\eta m\lambda + \eta^2 m \cdot n^2 \\
& \cdot d^2 \exp(16B) + \eta^2 m^2)
\end{aligned} \tag{17}$$

where the first step follows from Lemma J.1, the second step follows from the definitions of C_1, $C_{1,1}$ and $C_{1,2}$, the third step follows from Lemma J.2 and Lemma J.3.

Choice of parameter. Here, we explain the condition setting in Definition K.1:

- To get our results in Lemma J.2 and Lemma J.3, we have to let $m \geq \Omega(\lambda^{-2} n^2 d^2 \cdot \exp(30B) \cdot \sqrt{\log(nd/\delta)})$.
- If we let $\eta \leq O(\lambda/(mn^2 d^2 \exp(16B)))$, we can have

$$\eta^2 m \cdot n^2 d^2 \exp(16B) + \eta^2 m^2 \leq 0.9\eta m\lambda. \tag{18}$$

Thus, combining Eq. (17) and Eq. (18), we have

$$\|\mathsf{F}(t+1) - y\|_F^2 \leq (1 - m\eta\lambda/2) \cdot \|\mathsf{F}(t) - y\|_F^2 \tag{19}$$

Then by Eq. (19), we conclude all $\|\mathsf{F}(\tau) - y\|_F^2$ for $\tau \in [t]$, we have

$$\|\mathsf{F}(t+1) - y\|_F^2 \leq (1 - m\eta\lambda/2)^{t+1} \cdot \|\mathsf{F}(0) - y\|_F^2$$

□

K.4 Induction Part 3. For Gradient

In this section, we present the induction lemma for gradients.

Lemma K.5 (Induction Part 3 for gradient)**.** *Let t be a fixed integer.*

If the following conditions hold

- *Suppose properties in Definition K.1 are true*

Then we have

$$\eta \|\Delta w_r(t)\|_2 \leq 0.01, \forall r \in [m]$$

Proof. Firstly, we have

$$\begin{aligned}
& \|\Delta w_r(t)\|_2 \\
\leq & \|\Delta w_r(t)\|_1 \\
\leq & \sum_{k_1=1}^{d} \Big| m \sum_{i=1}^{n} \sum_{k=1}^{d} (\mathsf{F}_{k,i}(t) - y_{k,i}) \\
& \cdot \Big(\langle v_{k,r}(t), \mathsf{S}_i(t) \rangle \cdot \mathsf{S}_{i,r}(t) \cdot x_{i,k_1} \\
& + a_r \mathsf{S}_{i,r}(t) e_{k,k_1} \Big) \Big| \\
\leq & \sqrt{n} d \exp(11B) \|\mathsf{F}(t) - Y\|_F
\end{aligned} \tag{20}$$

where the first step follows from $\|U\|_F \leq \|U\|_1$ for $U \in \mathbb{R}^{n \times d}$, the second step follows from Claim H.12, the last step follows from the definition of 4 ℓ_1 norm, $0 \leq \mathsf{S}_{i,r} \leq \frac{\exp(3B)}{m}$ by Part 11 of Lemma M.1, $\|x_i\|_2 \leq 1$ and Part 6 of Lemma M.2.

Then by the property of η in Definition K.1, we have

$$\eta \|\Delta w_r(t)\|_2 \leq 0.01, \forall r \in [m]$$

□

K.5 Bounding Loss at Initialization

Lemma K.6. *If the following conditions hold*

- *Denote* $\mathsf{F}(\tau) \in \mathbb{R}^{n \times d}$ *as Definition H.8.*
- *Let* $Y \in \mathbb{R}^{n \times d}$ *denote the labels.*

Then we have

$$\|\mathsf{F}(0) - Y\|_F \leq O(\sqrt{nd})$$

Proof. This proof follows from $\|y_i\| \leq 1$ for $i \in [n]$ and Definition H.8. □

L NTK-Attention

In this section, we compute the error bound of our NTK-Attention in approximating prefix matrix $P \in \mathbb{R}^{m \times d}$. In Appendix L.1, we provide the formal definition of our NTK-Attention. In Appendix L.2, we give our main theorem of error bound. In Appendix L.3, we state tools from (Alman and Song, 2023).

L.1 Definitions

Definition L.1. *If the following conditions hold:*

- *Given input* $X \in \mathbb{R}^{L \times d}$*, prefix matrix* $P \in \mathbb{R}^{m \times d}$.
- *Let* $S := \begin{bmatrix} P \\ X \end{bmatrix} \in \mathbb{R}^{(m+L) \times d}$.
- *Given projections* $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$
- *Let* $Q := X W_Q \in \mathbb{R}^{L \times d}$.
- *Let* $K_P := S W_Q \in \mathbb{R}^{(m+L) \times d}$
- *Let* $V_P := S W_V \in \mathbb{R}^{(m+L) \times d}$
- *Let* $A := \exp(Q K_P^\top) \in \mathbb{R}^{L \times (m+L)}$.

- *Let* $D := \operatorname{diag}(A\mathbf{1}_{(m+L)}) \in \mathbb{R}^{L \times L}$.

We define:

$$\mathsf{Attn}(Q, K, V) := D^{-1} A V_P.$$

L.2 Error Bound

Here, we provide our two statements about error bound.

Theorem L.2 (Formal version of Theorem 5.1)**.** *Given an input matrix* $X \in \mathbb{R}^{L \times d}$ *and prefix matrix* $P \in \mathbb{R}^{m \times d}$*, we denote* $Q = XW_Q$*,* $K_C = PW_K$ *and* $V_C = PW_V$*. If the condition Eq.* (3)*,* $\|Q\|_\infty \leq o(\sqrt{\log m}), \|K_C\|_\infty \leq o(\sqrt{\log m}), \|V_C\|_\infty \leq o(\sqrt{\log m})$ *and* $d = O(\log m)$ *holds, then Algorithm 2 outputs a matrix* $T \in \mathbb{R}^{L \times d}$ *within time complexity of* $O(L^2 d)$ *that satisfies:*

$$\|T - \mathsf{PrefixAttn}(X, P)\|_\infty \leq 1/\operatorname{poly}(m).$$

Proof. Following Definition L.1, we can have matrix $A \in \mathbb{R}^{L \times (m+L)}$ as follows:

$$\begin{aligned} A &= QK^\top \\ &= \begin{bmatrix} \exp(XW_Q W_K^\top X^\top) & \exp(XW_Q W_K^\top P^\top) \end{bmatrix} \end{aligned}$$

where the second step follows from $K = SW_K$ and $S = \begin{bmatrix} P \\ X \end{bmatrix}$.

Our Algorithm 2 actually implement on using $Q = XW_Q$ and PW_K to approximate $\exp(XW_Q W_K^\top P^\top)$ by Lemma L.7.

Trivially, this proof follows from Theorem L.5 and Lemma L.7. □

Corollary L.3. *Given an input matrix* $X \in \mathbb{R}^{L \times d}$ *and prefix matrix* $P \in \mathbb{R}^{m \times d}$*, we denote* $Q = XW_Q$*,* $K_C = PW_K$ *and* $V_C = PW_V$*. If the condition Eq.* (3)*,* $\|Q\|_\infty \leq o(\sqrt{\log m}), \|K_C\|_\infty \leq o(\sqrt{\log m}), \|V_C\|_\infty \leq o(\sqrt{\log m})$ *and* $d = O(\log m)$ *holds, then there exists an algorithm that outputs a matrix* $T \in \mathbb{R}^{L \times d}$ *within time complexity of* $O(L^{1+o(1)} d)$ *that satisfies:*

$$\|T - \mathsf{PrefixAttn}(X, P)\|_\infty \leq 1/\operatorname{poly}(m).$$

Proof. The algorithm and proof can trivially follow from Algorithm 1, 2, 3 and Theorem 1 in HyperAttention (Han et al., 2024). □

L.3 Tools from Fast Attention

In this section, we introduce some tools from previous work which we have used.

Definition L.4 (Approximate Attention Computation $\mathsf{AAttC}(n, d, B, \epsilon_a)$, Definition 1.2 in (Alman and Song, 2023))**.** *Let* $\epsilon_a > 0$ *and* $B > 0$ *be parameters. Given three matrices* $Q, K, V \in \mathbb{R}^{n \times d}$*, with the guarantees that* $\|Q\|_\infty \leq B$*,* $\|K\|_\infty \leq B$*, and* $\|V\|_\infty \leq B$*, output a matrix* $T \in \mathbb{R}^{n \times d}$ *which is approximately equal to* $D^{-1}AV$*, meaning,*

$$\|T - D^{-1}AV\|_\infty \leq \epsilon_a.$$

Here, for a matrix $M \in \mathbb{R}^{n \times n}$*, we write* $\|M\|_\infty := \max_{i,j} |M_{i,j}|$.

Theorem L.5 (Upper bound, Theorem 1.4 in (Alman and Song, 2023))**.** *There is an algorithm that solves* $\mathsf{AAttC}(n, d = O(\log n), B = o(\sqrt{\log n}), \epsilon_a = 1/\operatorname{poly}(n))$ *in time* $n^{1+o(1)}$.

Definition L.6 (Definition 3.1 in (Alman and Song, 2023))**.** *Let* $r \geq 1$ *denote a positive integer. Let* $\epsilon \in (0, 0.1)$ *denote an accuracy parameter. Given a matrix* $A \in \mathbb{R}_{\geq 0}^{n \times n}$*, we say* $\widetilde{A} \in \mathbb{R}_{\geq 0}^{n \times n}$ *is an* (ϵ, r)*-approximation of* A *if*

- $\widetilde{A} = U_1 \cdot U_2^\top$ *for some matrices* $U_1, U_2 \in \mathbb{R}^{n \times r}$ *(i.e.,* $\widetilde{A}$ *has rank at most* r*), and*

- $|\widetilde{A}_{i,j} - A_{i,j}| \leq \epsilon \cdot A_{i,j}$ *for all* $(i,j) \in [n]^2$.

Lemma L.7 (Lemma 3.4 in (Alman and Song, 2023))**.** *Suppose* $Q, K \in \mathbb{R}^{n \times d}$*, with* $\|Q\|_\infty \leq B$*, and* $\|K\|_\infty \leq B$*. Let* $A := \exp(QK^\top/d) \in \mathbb{R}^{n \times n}$*. For accuracy parameter* $\epsilon \in (0,1)$*, there is a positive integer* g *bounded above by*

$$g = O\Big(\max\Big\{ \frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B^2)}, B^2 \Big\} \Big),$$

and a positive integer r *bounded above by*

$$r \leq \binom{2(g+d)}{2g}$$

such that: There is a matrix $\widetilde{A} \in \mathbb{R}^{n \times n}$ *that is an* (ϵ, r)*-approximation (Definition L.6) of* $A \in \mathbb{R}^{n \times n}$*. Furthermore, we can construct the matrices* $U_1 := \phi(Q)$ *and* $U_2 := \phi(K)$ *through a function* $\phi(\cdot)$ *defining* $\widetilde{A} = U_1 U_2^\top$ *can be computed in* $O(n \cdot r)$ *time.*

L.4 Derivation of Low-Rank Approximation on Z

Re-state Conditions. Given an input matrix $X \in \mathbb{R}^{L \times d}$ and prefix matrix $P \in \mathbb{R}^{m \times d}$, we denote $Q = XW_Q, K = XW_K V = XW_V \in \mathbb{R}^{L \times d}$. Besides, we also denote $K_C = PW_K, V_C = PW_V \in \mathbb{R}^{m \times d}$.

We relax the infinite norm ℓ_∞ on these state matrices to $o(\sqrt{\log(m+L)})$. Formally, we assume the following setting to align with (Alman and Song, 2023):

- $\|Q\|_\infty \leq o(\sqrt{\log(m+L)})$.
- $\|K\|_\infty \leq o(\sqrt{\log(m+L)})$.
- $\|V\|_\infty \leq o(\sqrt{\log(m+L)})$.
- $\|K_C\|_\infty \leq o(\sqrt{\log(m+L)})$.
- $\|V_C\|_\infty \leq o(\sqrt{\log(m+L)})$.

Note that our kernel mapping function ϕ doesn't enlarge the values of input entries, we have:

- $\|\phi(Q_i)\|_\infty \leq o(\sqrt{\log(m+L)})$, for any $i \in [L]$.
- $\|\phi(K_{C,r})\|_\infty \leq o(\sqrt{\log(m+L)})$, for any $r \in [m]$.
- $\|\Phi(Q)\|_\infty \leq o(\sqrt{\log(m+L)})$.
- $\|\Phi(K_C)\|_\infty \leq o(\sqrt{\log(m+L)})$.

Note that $Z = \Phi(K_C)^\top \cdot V_C$, where $K_C, V_C \in \mathbb{R}^{m \times d}$ and $\Phi(K_C) \in \mathbb{R}^{m \times r}$. The minimum error of low-rank approximation is given by:

$$\begin{aligned}
&\min_{Z_A \in \mathbb{R}^{r \times s}, Z_B \in \mathbb{R}^{s \times d}} \|Z_A \cdot Z_B - Z\|_\infty \\
\leq &\min_{Z_A \in \mathbb{R}^{r \times s}, Z_B \in \mathbb{R}^{s \times d}} \|Z_A \cdot Z_B - Z\|_F \\
\leq &\sqrt{d-s}\|\Phi(K_P)\|_\infty \cdot \|V_P\|_\infty \\
\leq &\, o(\log(m+L)) \cdot \sqrt{d-s}
\end{aligned}$$

Recall that we follow from Lemma L.7, for $i \in [L]$, we have:

$$\begin{aligned}
&\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \phi(Q_i)^\top k \\
\geq &\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \exp(Q_i^\top K_C^\top/\sqrt{d})\mathbf{1}_m
\end{aligned}$$

$$- \epsilon \cdot m$$

where we choose this $\epsilon = 1/\operatorname{poly}(m)$.

So we obtain the error of low-rank approximation on prefix attention computation:

$$\begin{aligned}
& \| \frac{\exp(Q_i^\top K^\top/\sqrt{d})V + \phi(Q_i)^\top Z}{\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \phi(Q_i)^\top k} \\
& - \frac{\exp(Q_i^\top K^\top/\sqrt{d})V + \phi(Q_i)^\top Z_A \cdot Z_B}{\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \phi(Q_i)^\top k} \|_\infty \\
= & \|\phi(Q_i)^\top (Z - Z_A \cdot Z_B) \\
& /(\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L + \phi(Q_i)^\top k)\|_\infty \\
\leq & \|\phi(Q_i)^\top (Z - Z_A \cdot Z_B) \\
& /(\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L \\
& + \exp(Q_i^\top K_C^\top/\sqrt{d})\mathbf{1}_m - 1/\operatorname{poly}(m))\|_\infty \\
= & \|\phi(Q_i)^\top (Z - Z_A \cdot Z_B)\|_\infty \\
& /(\exp(Q_i^\top K^\top/\sqrt{d})\mathbf{1}_L \\
& + \exp(Q_i^\top K_C^\top/\sqrt{d})\mathbf{1}_m - 1/\operatorname{poly}(m)) \\
\leq & \frac{\|\phi(Q_i)^\top (Z - Z_A \cdot Z_B)\|_\infty}{\frac{1}{2}(m+L)\exp(-o(\log(m+L))/\sqrt{d})} \\
\leq & \frac{\|\phi(Q_i)\|_\infty \|Z - Z_A \cdot Z_B\|_\infty}{\frac{1}{2}(m+L)\exp(-o(\log(m+L))/\sqrt{d})} \\
\leq & \frac{o(\sqrt{\log(m+L)})\|Z - Z_A \cdot Z_B\|_\infty}{\frac{1}{2}(m+L)\exp(-o(\log(m+L))/\sqrt{d})} \\
\leq & \frac{o(\log^{1.5}(m+L)) \cdot \sqrt{d-s} \cdot o(m+L)^{1/\sqrt{d}}}{\frac{1}{2}(m+L)} \\
\leq & \frac{1}{m^C}
\end{aligned}$$

where the last step follows from choosing $m \geq \Omega(\exp(d))$ and $C \in (0, (\sqrt{d}-1)/\sqrt{d})$ is some constant.

M Taylor Series

In this section, we provide some perturbation analysis for NTK analysis.

Lemma M.1 (Lemma B.1 in (Li et al., 2024a))**.** *If the following conditions hold*

- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $W = [w_1, \cdots, w_m]$ *and* w_r *be random Gaussian vectors from* $\mathcal{N}(0, \sigma^2 I_d)$.
- *Let* $V = [v_1, \cdots, v_m]$ *and* v_r *denote the vector where* $\|v_r - w_r\|_2 \leq R$, $\forall r \in [m]$.
- *Let* $x_i \in \mathbb{R}^d$ *and* $\|x_i\|_2 \leq 1$, $\forall i \in [n]$.
- *Let* $R \in (0, 0.01)$.
- *Let* S_i *and* $\widetilde{\mathsf{S}}_i$ *be the softmax function corresponding to* W *and* V *respectively.*
- *Let* $\alpha_i = \langle \mathbf{1}_m, \exp(W^\top x_i)\rangle$ *and* $\widetilde{\alpha}_i = \langle \mathbf{1}_m, \exp(V^\top x_i)\rangle$, $\forall i \in [n]$.

Then, with probability at least $1 - \delta/\operatorname{poly}(nd)$*, we have*

- *Standard inner product*
  - *Part 1.* $|\langle w_r, x_i \rangle| \le B$, $\forall i \in [n]$, $\forall r \in [m]$
  - *Part 2.* $|\langle v_r, x_i \rangle| \le B + R$, $\forall i \in [n]$, $\forall r \in [m]$
  - *Part 3.* $|\langle w_r - v_r, x_i + x_j \rangle| \le 2R$, $\forall i, j \in [n]$, $\forall r \in [m]$
- $\exp$ *function*
  - *Part 4.* $\exp(-B) \le \exp(\langle w_r, x_i \rangle) \le \exp(B)$, $\forall i \in [n]$, $\forall r \in [m]$
  - *Part 5.* $\exp(-B - R) \le \exp(\langle v_r, x_i \rangle) \le \exp(B + R)$, $\forall i \in [n]$, $\forall r \in [m]$
  - *Part 6.* $|\exp(\langle w_r - v_r, x_i + x_j \rangle) - 1| \le 4R$, $\forall i, j \in [n]$, $\forall r \in [m]$
  - *Part 7.* $|\exp(\langle w_r, x_i \rangle) - \exp(\langle v_r, x_i \rangle)| \le R \exp(B + R)$, $\forall i \in [n]$, $\forall r \in [m]$
- *softmax* S *function*
  - *Part 8.* $|\alpha_i - \widetilde{\alpha}_i| \le mR \exp(B + R), \forall i \in [n]$
  - *Part 9.* $|\alpha_i^{-1} - \widetilde{\alpha}_i^{-1}| \le \frac{R}{m} \exp(3B + 2R), \forall i \in [n]$
  - *Part 10.* $|\mathsf{S}_{i,r}| \le \exp(2B)/m, \forall i \in [n], \forall r \in [m]$
  - *Part 11.* $|\widetilde{\mathsf{S}}_{i,r}| \le \exp(2B + 2R)/m, \forall i \in [n], \forall r \in [m]$
  - *Part 12.* $|\mathsf{S}_{i,r} - \widetilde{S}_{i,r}| \le \frac{R}{m} \exp(4B + 3R), \forall i \in [n], \forall r \in [m]$
  - *Part 13. for any* $z \in \mathbb{R}^m$ *and* $\|z\|_\infty \le 1$*, we have* $|\langle z, \mathsf{S}_i \rangle - \langle z, \widetilde{\mathsf{S}}_i \rangle| \le R \exp(4B + 3R), \forall i \in [n]$

Lemma M.2. *If the following conditions hold*

- *Let* $C > 10$ *denote a sufficiently large constant*
- *Let* $B := \max\{C\sigma\sqrt{\log(nd/\delta)}, 1\}$.
- *Let* $W = [w_1, \cdots, w_m]$ *and* w_r *be random Gaussian vectors from* $\mathcal{N}(0, \sigma^2 I_d)$.
- w_r *for* $r \in [m]$ *satisfies* $\|w_r\|_2 \le B$ *with probability at least* $1 - \delta/\operatorname{poly}(nd)$ *as in Lemma M.1.*
- *Let* $a \in \mathbb{R}^m$ *be defined as Definition G.1.*
- *Define* $\beta_k := W_{k,*} \circ a \in \mathbb{R}^m$ *for* $k \in [d]$ *as Definition G.5.*
- *Define* $v_{k,r} := \beta_{k,r} \cdot \mathbf{1}_m - \beta_k \in \mathbb{R}^m$ *for* $k \in [d]$ *and* $r \in [m]$ *as Definition I.1.*
- *Define* α_i *for* $i \in [n]$ *as Definition G.3.*

Then, with probability at least $1 - \delta/\operatorname{poly}(nd)$*, we have*

- *Part 1.* $|\beta_{k,r}| \le B$
- *Part 2.* $\|\beta_k\|_2 \le B\sqrt{m}$
- *Part 3.* $\|v_{k,r}\|_2 \le 2\sqrt{m}B$
- *Part 4.* $|\alpha^{-1}| \le \exp(B)/m$
- *Part 5.* $\langle \beta_k, \mathsf{S}_i \rangle \le \exp(4B)$
- *Part 6.* $\langle v_{k,r}, \mathsf{S}_i \rangle \le \exp(6B)$

Proof. **Proof of Part 1.** We can get the proof by Gaussian tail bound.

Proof of Part 2. We have

$$\begin{aligned}\|\beta_k\|_2 &= \sqrt{\sum_{r=1}^m \beta_{k,r}^2} \\ &\leq \sqrt{\sum_{r=1}^m B^2} \\ &\leq \sqrt{m} \cdot B\end{aligned}$$

where the first step follows from the definition of ℓ_2 norm, the second step follows from Part 1 of this Lemma, the last step follows from simple algebras.

Proof of Part 3. We have

$$\begin{aligned}\|v_{k,r}\|_2 &= \sqrt{\sum_{r_1=1}^m (\beta_{k,r} - \beta_{k,r_1})^2} \\ &\leq \sqrt{\sum_{r_1=1}^m \beta_{k,r}^2 + \beta_{k,r_1}^2 + |2\beta_{k,r}\beta_{k,r_1}|} \\ &\leq \sqrt{\sum_{r_1=1}^m 4B^2} \\ &\leq 2\sqrt{m} \cdot B\end{aligned}$$

where the first step follows from the definition of ℓ_2 norm, the second step follows from simple algebras, the third step follows from Part 1 of this Lemma, the last step follows from simple algebras.

Proof of Part 4. This proof follows from Part 4 of Lemma M.1 and Definition G.3.

Proof of Part 5. We have

$$\begin{aligned}\langle \beta_k, \mathsf{S}_i \rangle &\leq \|\beta_k\|_2 \cdot \|\mathsf{S}_i\|_2 \\ &\leq \sqrt{m} B \cdot \|\mathsf{S}_i\|_2 \\ &\leq \sqrt{m} B \cdot \sqrt{\sum_{r=1}^m \mathsf{S}_{i,r}^2} \\ &\leq \sqrt{m} B \cdot \sqrt{\sum_{r=1}^m \frac{\exp(6B)}{m^2}} \\ &\leq \sqrt{m} B \cdot \sqrt{\frac{\exp(6B)}{m}} \\ &\leq B \exp(3B) \\ &\leq \exp(4B)\end{aligned}$$

where the first step follows from Cauchy-Schwarz inequality, the second step follows from Part 2 of this Lemma, the third step follows from the definition of ℓ_2 norm, the fourth step follows from Part 11 of Lemma M.1, the fifth step follows from triangle inequality, the sixth step follows from $B \leq \exp(B)$, last step follows from simple algebras.

Proof of Part 6. This proof follows from Part 3 of this Lemma, $B \leq \exp(B)$ and Part 11 of Lemma M.1. □