A Fast Optimization View: Reformulating Single Layer Attention in LLM Based on Tensor and SVM Trick, and Solving It in Matrix Multiplication Time

Yeqi Gao[1] **Zhao Song**[2,*] **Weixin Wang**[3,†] **Junze Yin**[4,‡]

[1]University of Washington, [2]University of California, Berkeley
[3]Johns Hopkins University, [4]Boston University
[*]magic.linuxkde@gmail.com, [†]weixinw1@uci.edu,[‡]junze@bu.edu

Abstract

Large language models (LLMs) have played a pivotal role in revolutionizing various facets of our daily existence. Solving attention regression is a fundamental task in optimizing LLMs. In this work, we focus on providing a provable guarantee for the one-layer attention network objective function: given input matrices of a layer, $A_1, A_2, A_3 \in \mathbb{R}^{n \times d}$, our goal is to minimize the loss function:

$$L(X,Y) = \sum_{j_0=1}^{n} \sum_{i_0=1}^{d} (\langle \langle \exp(\mathsf{A}_{j_0} x), \mathbf{1}_n \rangle^{-1} \cdot \exp(\mathsf{A}_{j_0} x), A_3 Y_{*,i_0} \rangle - b_{j_0,i_0})^2,$$

where $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ is the j_0-th block of the Kronecker product of A_1 and A_2. The matrix $B \in \mathbb{R}^{n \times d}$ represents the output of a layer, and $b_{j_0,i_0} \in \mathbb{R}$ is the (j_0, i_0)-th entry of B. $Y_{*,i_0} \in \mathbb{R}^d$ is the i_0-th column vector of Y, and $x \in \mathbb{R}^{d^2}$ is the vectorization of X.

In self-attention, $Q, K, V \in \mathbb{R}^{d \times d}$ represent the weight matrices for the query, key, and value, respectively. Unlike prior works that relied on simplified and single-variable versions of the attention computation problem, our multivariate loss function analyzes a complete and unsimplified attention layer, treating all these weights as variables, where $X = QK^\top \in \mathbb{R}^{d \times d}$ and $Y = V \in \mathbb{R}^{d \times d}$. We propose an iterative greedy algorithm to train a neural network using the loss function $L(X,Y)$, achieving an error bound of $\epsilon \in (0, 0.1)$. The algorithm runs in $\widetilde{O}((\mathcal{T}_{\mathrm{mat}}(n,n,d) + \mathcal{T}_{\mathrm{mat}}(n,d,d) + d^{2\omega}) \log(1/\epsilon))$ time, where $\mathcal{T}_{\mathrm{mat}}(a,b,c)$ denotes the time complexity of multiplying an $a \times b$ matrix with a $b \times c$ matrix, and $\omega \approx 2.37$ is the exponent of matrix multiplication.

1 INTRODUCTION

Large language models (LLMs) like GPT-1 [Radford et al., 2018], BERT [Devlin et al., 2019], GPT-2 [Radford et al., 2019], GPT-3 [Brown et al., 2020], ChatGPT [OpenAI, 2022], GPT-4 [OpenAI, 2023], OPT [Zhang et al., 2022], Llama [Touvron et al., 2023a], and Llama 2 [Touvron et al., 2023b] have demonstrated impressive capabilities in natural language processing (NLP). These models understand and generate complex language, enabling a wide range of applications such as sentiment analysis [Zhang et al., 2024], language translation [Alyafeai et al., 2023], question answering [Bian et al., 2024], and text summarization [Liu and Demberg, 2023]. Despite their high-quality performance, there remains untapped potential in optimizing and training these massive models, making it a challenging endeavor in the present day.

The primary technical foundation supporting the capabilities of LLMs is the attention matrix $A \in \mathbb{R}^{n \times n}$ [Radford et al., 2018, Vaswani et al., 2017, Brown et al., 2020, Devlin et al., 2019]. The central concept of attention is to learn representations that emphasize the most relevant parts of the input. To be more specific, the attention mechanism finds the correlations of the query vectors and the key vectors using the inner product. The attention weights are then determined based on the similarity of this comparison, indicating the relative importance of each input token. These attention weights are used to compute weighted averages of the value vectors, resulting in the output representation. By leveraging attention, LLMs acquire the ability to focus on the crucial aspects of the input, allowing them to gather pertinent information more efficiently and precisely. This capability enables LLMs to process longer texts and comprehend intricate semantic relationships. Notably, the self-attention mechanism enables LLMs to establish connections between various segments of the input sequence, enhancing their contextual understanding. Mathematically, the attention computation is defined as follows:

Definition 1.1 (The ℓ-th layer forward computation)**.** *Let*

Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025), PMLR 244:1381–1452.

n, d be positive integers, where n denotes the number of input tokens and d represents the dimensionality of the token embeddings. Let $\mathbf{1}_n$ be the n-dimensional vector whose entries are all 1. Let $\mathrm{diag} : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ be a function: each entry of the vector in $\mathbb{R}^n$ is mapped to the diagonal entry of the matrix in $\mathbb{R}^{n \times n}$ and other entries of this matrix are all 0's. Given weights $Q, K, V \in \mathbb{R}^{d \times d}$, we let $X_\ell \in \mathbb{R}^{n \times d}$ denote the ℓ-th layer input and $X_{\ell+1} \in \mathbb{R}^{n \times d}$ is as follows:

$$X_{\ell+1} \leftarrow D^{-1} \exp(X_\ell Q K^\top X_\ell^\top) X_\ell V$$

where $D := \mathrm{diag}(\exp(X_\ell Q K^\top X_\ell^\top) \mathbf{1}_n)$ and $\exp(A)_{i,j} = \exp(A_{i,j})$ for all matrices A.

Traditionally, $D^{-1} \underbrace{\exp(X_\ell Q K^\top X_\ell^\top)}_{:=A} \in \mathbb{R}^{n \times n}$ is denoted by $\mathrm{Softmax}(\frac{QK^\top}{\sqrt{d}}) \in \mathbb{R}^{n \times n}$, where each entry of A represents how much focus one part of the input should pay to another part. D^{-1} is used to normalize each row of the attention matrix, i.e., the sum of each row of $D^{-1}A \in \mathbb{R}^{n \times n}$ is equal to 1. $X_\ell V \in \mathbb{R}^{n \times d}$ is the value matrix that stores the representations or features of each input element. This results in an output representing a combination of the input values, with more important values (as determined by the attention mechanism) contributing more to the final output. In Definition 1.1, we fully expand the $\mathrm{Softmax}$ unit and change the notation system from the traditional definition to highlight the focus of our paper, which is to look for $X = QK^\top \in \mathbb{R}^{d \times d}$ and $Y = V \in \mathbb{R}^{d \times d}$ that minimizes the following optimization problem with respect to attention computation:

Definition 1.2 (Attention optimization)**.** *Let $B \in \mathbb{R}^{n \times d}$ and $X, Y \in \mathbb{R}^{d \times d}$. Given inputs $A_1, A_2, A_3 \in \mathbb{R}^{n \times d}$, we define the attention optimization $\min_{X,Y \in \mathbb{R}^{d \times d}} L(X, Y)$ as:*

$$\min_{X,Y \in \mathbb{R}^{d \times d}} \|D(X)^{-1} \exp(A_1 X A_2^\top) A_3 Y - B\|_F^2,$$

where the diagonal matrix $D(X) \in \mathbb{R}^{n \times n}$ is defined as $D(X) := \mathrm{diag}(\exp(A_1 X A_2^\top) \mathbf{1}_n)$.

Here, $X = QK^\top$ and $Y = V$ are the weights we want to learn, while A_1, A_2, A_3 are the inputs of a layer X_ℓ, and B is the output layer $X_{\ell+1}$. Solving the attention optimization problem exactly takes $O(n^2 d)$ time. Since the attention matrix $A = \exp(A_1 X A_2^\top)$ has n^2 entries, explicitly computing all entries of A makes it impossible to achieve a sub-quadratic time algorithm. In real-world applications, $n \gg d$ [Alman and Song, 2023], so prior works mainly focus on approximating the attention computation to obtain a sub-quadratic time algorithm in n.

Limitations of Prior Works Attention computation has been analyzed in many recent works [Alman and Song, 2023, Brand et al., 2024, Gao et al., 2025b, Deng et al., 2023b, Song et al., 2024a, Deng et al., 2023a, Gao et al., 2023a,c], but none of them provide a complete approximation of the full version of the attention optimization problem. Each of these works simplifies the problem (Definition 1.2) using different strategies. For example, Zandieh et al. [2023], Brand et al. [2024] merge $A_1 X$ and $A_3 Y$ into a single matrix, respectively, by approximating

$$D(X)^{-1} \exp(QK^\top) V.$$

Kacham et al. [2023] replaces the $\exp$ function in Definition 1.2 with polynomials. Another major branch of studies on attention regression simplification focuses on the softmax regression problem, where the matrix $A_3 Y$ is completely ignored, along with its variants.

Definition 1.3 (Single softmax regression [Deng et al., 2023a] and multiple softmax regression [Gao et al., 2023b])**.** *Given a matrix $A \in \mathbb{R}^{n \times d}$ and a vector $c \in \mathbb{R}^n$, the single softmax regression problem is defined as*

$$\textbf{Part 1}. \min_{x \in \mathbb{R}^d} \|\langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - c\|_2^2.$$

Let $D(X) \in \mathbb{R}^{n \times n}$ be defined as in Definition 1.2 and $C \in \mathbb{R}^{n \times n}$. Given $A_1, A_2 \in \mathbb{R}^{n \times d}$ and $X \in \mathbb{R}^{d \times d}$, the multiple softmax regression problem is defined as

$$\textbf{Part 2}. \min_{X \in \mathbb{R}^{d \times d}} \|D(X)^{-1} \exp(A_1 X A_2^\top) - C\|_F^2.$$

Based on the observation in Gao et al. [2023b,c], the equation in **Part 1** of Definition 1.3 can be viewed as a single row of the equation in **Part 2** of Definition 1.3. When studying multiple softmax regression, Deng et al. [2023b] impose an additional assumption by considering only symmetric matrices:

$$D(X)^{-1} \exp(A_2 A_2^\top),$$

but in exchange, they consider the stronger ℓ_∞ norm in multiple softmax regression. Gao et al. [2025c, 2023b] respectively study the rescaled version of single and multiple softmax regression, namely

$$\min_{x \in \mathbb{R}^d} \|\exp(Ax) - \langle \exp(Ax), \mathbf{1}_n \rangle c\|_2^2$$

and

$$\min_{X \in \mathbb{R}^{d \times d}} \|\exp(A_1 X A_2^\top) - D(X) C\|_F^2.$$

We note that all of these softmax-related regression problems consider simpler variants to achieve sub-quadratic time algorithms: they focus only on single-variable loss functions. Specifically, they minimize the loss by adjusting the weights of the key and query matrix, $X = QK^\top$, while ignoring the weight of the value matrix, $Y = V$. However, simplifying the attention optimization problem in this way may significantly degrade model performance, potentially requiring additional training or fine-tuning. This, in turn, creates deployment challenges [Dong et al., 2023]. Therefore, it is natural to ask:

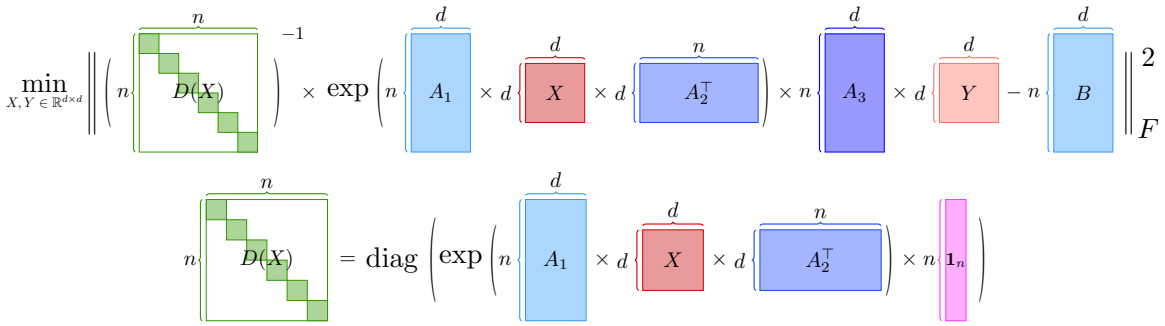

Figure 1: The visualization of the attention optimization problem (see Definition 1.2). Let $A_1, A_2, A_3, B \in \mathbb{R}^{n \times d}$ and $X, Y \in \mathbb{R}^{d \times d}$. We first get $\exp(A_1 X A_2^\top) \in \mathbb{R}^{n \times n}$ by multiplying A_1, X, and $A_2^\top$. Then, we have $D(X) \in \mathbb{R}^{n \times n}$ by computing $\mathrm{diag}(\exp(A_1 X A_2^\top)\mathbf{1}_n)$. After that, we multiply $D(X)^{-1}$, $\exp(A_1 X A_2^\top)$, A_3, and Y and subtract B from their product. Finally, we compute the minimum of the Frobenius norm of their difference. The blue rectangles represent the $n \times d$ matrices, the purple rectangle represents the n-dimensioal vector, the red squares represent the $d \times d$ matrices, and the green squares represent the $n \times n$ diagonal matrices.

How fast can we optimize the training process of the attention matrix without making any simplification to Definition 1.2?

Our Result Although Alman and Song [2023] shows that a one-step forward approximation of attention can be achieved in $o(n^2)$ time without explicitly formulating the $n \times n$ matrix, the speed at which the loss function can be optimized via iterative methods remains an open problem. Therefore, in this paper, we provide a complete, unsimplified analysis of the attention optimization problem as defined in Definition 1.2–a task that, to the best of our knowledge, has not been previously undertaken. Additionally, we establish a provable guarantee for optimizing the attention function in the case of a single-layer attention network.

Theorem 1.4 (Informal version of our main theorem (Theorem L.1))**.** *Given $A_1, A_2, A_3 \in \mathbb{R}^{n \times d}$, there exists an algorithm (Algorithm 1) that runs in $O((\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(n,d,d) + d^{2\omega})\log(1/\epsilon))$ and solves the attention optimization problem (Defintion 1.2) up to ϵ accuracy with probability $1 - 1/\mathrm{poly}(n)$. Here $\omega \approx 2.37$*[1].

Optimizing the attention objective is a necessary subproblem that needs to be solved as part of the overall LLM training process, even if it's not sufficient on its own due to the presence of additional layers. Developing faster, more scalable algorithms for attention optimization can help reduce the computational burden of training LLMs.

To establish the correctness of our algorithm, we conduct a comprehensive analysis of the positive semi-definite (PSD) property and the Lipschitz continuity of the Hessian matrix constructed from the attention matrix. These two properties provide the necessary assurance for employing TensorSRHT and Newton's method, ensuring both fast computation and convergence, respectively.

[1] ω denotes the exponent of matrix multiplication [Williams, 2012, Le Gall, 2014, Alman and Williams, 2021, Duan et al., 2023, Le Gall, 2024, Williams et al., 2024], $\mathcal{T}_{\mathrm{mat}}(a,b,c)$ denotes the time of multiplying an $a \times b$ size matrix with another $b \times c$ size matrix, and $\mathcal{T}_{\mathrm{mat}}(n,n,n) = n^\omega$. See more details of matrix multiplication notation in Section A.7.

Notation We use $\mathbb{N}$ to denote the set of positive integers. Let $n, d \in \mathbb{N}$. We define $[n] := \{1, 2, \ldots, n\}$. Let $x, y \in \mathbb{R}^d$. For all $i \in [d]$, we define $x_i \in \mathbb{R}$ as the i-th entry of x. We define $\langle \cdot, \cdot \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as $\langle x, y \rangle := \sum_{i=1}^d x_i y_i$. For all $p \in \{1, 2, \infty\}$, we define $\|x\|_p := (\sum_{i \in [d]} |x_i|^p)^{1/p}$. We use $\mathbf{1}_d$ and $\mathbf{0}_d$ to denote the d-dimensional vectors whose entries are all 1's and 0's, respectively.

Let $A \in \mathbb{R}^{n \times d}$. For all $i \in [n]$ and $j \in [d]$, we use $A_{i,j} \in \mathbb{R}$ to denote the (i,j)-th entry of A, use $A_{i,*} \in \mathbb{R}^d$ and $A_{*,j} \in \mathbb{R}^n$ to denote vectors, where $(A_{i,*})_j = A_{i,j} = (A_{*,j})_i$. We use $A^\top \in \mathbb{R}^{d \times n}$ to denote the transpose of the matrix A. For $X \in \mathbb{R}^{d \times d}$, we define $x = \mathrm{vec}(X) \in \mathbb{R}^{d^2}$ as $X_{i,j} = \mathrm{vec}(X)_{(i-1) \times d + j}$. For $x \in \mathbb{R}^d$, we define $\mathrm{diag}(x) \in \mathbb{R}^{d \times d}$ as $\mathrm{diag}(x)_{i,i} = x_i$, for all $i \in [d]$ and other entries of $\mathrm{diag}(x)$ are all 0's. $\|A\|_F \in \mathbb{R}$ and $\|A\| \in \mathbb{R}$ denote the Frobenius norm and the spectral norm of $A \in \mathbb{R}^{n \times d}$, respectively, where $\|A\|_F := \sqrt{\sum_{i \in [n]} \sum_{j \in [d]} |A_{i,j}|^2}$ and $\|A\| := \max_{x \in \mathbb{R}^d} \|Ax\|_2 / \|x\|_2$. Let $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$. For each $j_1 \in [n]$, we use $\mathsf{A}_{j_1} \in \mathbb{R}^{n \times d^2}$ to denote one $n \times d^2$ block from $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$. Let $C, D \in \mathbb{R}^{d \times d}$ be symmetric matrices, $C \succeq D$ if for all $y \in \mathbb{R}^d$, $y^\top C y \geq y^\top D y$. C is said to be a positive semidefinite (PSD) matrix if $y^\top C y \geq 0$. We use I_d to denote the $d \times d$ identity matrix. Let $A \in \mathbb{R}^{n_1 \times d_1}$ and $B \in \mathbb{R}^{n_2 \times d_2}$. We define the Kronecker product between matrices A and B, denoted $A \otimes B \in \mathbb{R}^{n_1 n_2 \times d_1 d_2}$, as $(A \otimes B)_{(i_1-1)n_2 + i_2, (j_1-1)d_2 + j_2}$ is equal to $A_{i_1,j_1} B_{i_2,j_2}$, where $i_1 \in [n_1], j_1 \in [d_1], i_2 \in [n_2], j_2 \in [d_2]$.

Roadmap In Section 2, we introduce related research work. In Section 3, we provide an overview of the techniques we will use throughout the rest of the paper. In Section 4, we present a discussion of our theoretical results. In Section 5, we draw a conclusion for this paper.

2 RELATED WORK

Attention Transformer models, proposed by Vaswani et al. [2017], revolutionized attention computation with their self-attention mechanism. This allowed for parallel processing of input sequences and captured long-range dependencies more effectively than previous recurrent architectures. After that, there has been a substantial body of work on attention computation [Deng et al., 2023b, Alman and Song, 2023, Zandieh et al., 2023, Chen et al., 2021, Li et al., 2023c, Brand et al., 2024, Kitaev et al., 2020]. Notably, recent research by Zandieh et al. [2023], Chen et al. [2021], Kitaev et al. [2020] employs Locality Sensitive Hashing (LSH) techniques to approximate attention mechanisms. In particular, Zandieh et al. [2023] introduces KDEformer, an efficient algorithm for approximating dot-product attention. This algorithm provides provable spectral norm bounds and outperforms various pre-trained models. Additionally, current research explores both static and dynamic approaches to calculating attention, as evidenced by the works of Brand et al. [2024] and Alman and Song [2023]. Furthermore, Li et al. [2023c] delves into the regularization of hyperbolic regression problems, which involve functions like $\exp$, $\sinh$, and $\cosh$. Deng et al. [2023b] proposes randomized and deterministic algorithms for reducing the dimensionality of attention matrices in LLMs, achieving high accuracy while significantly reducing feature dimensions.

Additionally, numerous studies have attempted to analyze theoretical attention from the perspectives of optimization and convergence [Li et al., 2023b, Gao et al., 2023a, Snell et al., 2021, Zhang et al., 2020a]. Li et al. [2023b] investigated how transformers acquire knowledge about word co-occurrence patterns. Gao et al. [2023a] focused on studying regression problems inspired by neural networks that employ exponential activation functions. Snell et al. [2021] analyzed why models occasionally prioritize significant words and explained how the attention mechanism evolves during the training process. Zhang et al. [2020a] demonstrated that the presence of a heavy-tailed noise distribution contributes to the bad performance of stochastic gradient descent (SGD) compared to adaptive methods.

Theoretical LLMs There are numerous amount of works focusing on the theoretical aspects of LLMs. In Reif et al. [2019], the syntactic representations of the attention matrix and the individual word embeddings are presented, together with the mathematical justification of elucidating the geometrical properties of these representations. Hewitt and Manning [2019] introduces a structural probe that analyzes, under the linear transformation of a word representation space of a neural network, whether or not syntax trees are embedded.

Cai et al. [2021], Liu et al. [2024], Rafailov et al. [2023], Kaplan et al. [2020] study the optimization of LLMs. Cai et al. [2021] proposes a new algorithm called ZO-BCD. It has favorable overall query complexity and a smaller computational complexity in each iteration. Liu et al. [2024] creates a simple and scalable second-order optimizer, called Sophia. In different parts of the parameter, Sophia adapts to the curvature. This may be strongly heterogeneous for language modeling tasks. The bound of the running time does not rely on the condition number of the loss.

Other theoretical LLM papers study the knowledge and skills of LLMs. Wang et al. [2022] analyzes distinct "skill" neurons, which are regarded as robust indicators of downstream tasks when employing the process of soft prompt-tuning, as discussed in Li and Liang [2021], for language models. Dai et al. [2021] finds a positive relationship between the activation of these neurons and the expression of their corresponding facts, through analyzing BERT. Simultaneously, Burns et al. [2023] employs a fully unsupervised approach to extract latent knowledge from a language model's internal activations. In addition, Hase et al. [2023] and Meng et al. [2022] show that in the feed-forward layers of pre-trained models, language models localize knowledge. Xie et al. [2022] explores the feasibility of selecting a specific subset of layers for modification and determining the optimal location for integrating a classifier. Li et al. [2023d] demonstrates that large trained transformers exhibit sparsity in their feedforward activations. Zero-th order algorithm for training LLM has been analyzed [Malladi et al., 2023, Deng et al., 2024, Zelikman et al., 2023].

A notable line of research is analyzing the theoretical limits of LLMs and discussing how to overcome these limitations. Recent works have shown that a wide range of LLM architectures fall into a weaker class of logical circuits Li et al. [2024, 2025a], Chen et al. [2024a, 2025c], which resonates with similar results in other neural architectures Li et al. [2025b], Ke et al. [2025], and such limitation may be improved by chain-of-thought Li et al. [2024] or positional encoding Yang et al. [2025]. Another line of research shows that Transformers may not be able to learn the support set of some simple Boolean functions under gradient descent Chen et al. [2025a,b], Hu et al. [2025e], Kim and Suzuki [2025] without the help of chain-of-thoughts. There are also works discussing the conditions deciding whether we can approximate Transformer computation efficiently, such as bounded entries Alman and Song [2023, 2024a,b, 2025a,b], statistical rates Hu et al. [2025d, 2024], and model pruning Frantar and Alistarh [2023], Liang et al. [2025], Gao et al. [2025a]. These theoretical results extend to universal approximation Kratsios et al. [2022], Chen et al. [2025d],

Liu et al. [2025], Hu et al. [2025a], model tuning Hu et al. [2025b,c], and in-context learning Wu et al. [2025b,a].

LLMs Application and Evaluation Recently, there has been much interest in developing LLM-based systems for conversational AI and task-oriented dialogue, like Google's Meena chatbot Rathee [2020], Microsoft 365 Copilot Spataro [2023], Adobe firefly, Adobe Photoshop, GPT series Radford et al. [2018, 2019], Brown et al. [2020], OpenAI [2022, 2023], and BERT Devlin et al. [2019]. Moreover, numerous fine-tuning methods such as Hu et al. [2022], Meng et al. [2024], Cao and Song [2025] appear in order to adapt models for different conversational tasks better.

Moreover, LLM evaluation is also a popular research area. Within the field of NLP, LLMs are evaluated based on natural language understanding Bang et al. [2023], Liang et al. [2023], Laskar et al. [2023], Choi et al. [2023], reasoning Bian et al. [2024], Wu et al. [2023], Xu et al. [2025a], natural language generation Wang et al. [2023b], Qin et al. [2023a], Liu and Demberg [2023], Chia et al. [2023], Chen et al. [2023], and multilingual tasks Abdelali et al. [2024], Ahuja et al. [2023], Lai et al. [2023], Zhang et al. [2023]. Robustness Li et al. [2023a], Wang et al. [2023a], Zhao et al. [2023], ethics Cao et al. [2023], biases Ferrara [2023], and trustworthiness Hagendorff and Fabi [2023] are also important aspects. More specifically, the abilities of LLMs in social science Deroy et al. [2023], Frank [2023], Nay et al. [2023], mathematics Arora et al. [2023], Dao and Le [2023], Wei et al. [2023], Bubeck et al. [2023], science Castro Nascimento and Pimentel [2023], Guo et al. [2023], engineering Bubeck et al. [2023], Liu et al. [2023a], Pallagani et al. [2023], Sridhara et al. [2023], and medical applications Chervenak et al. [2023], Johnson et al. [2023] are evaluated. LLMs are also core for different modalities, including speech Chen et al. [2024b], Ju et al. [2024], image Ho et al. [2020], Rombach et al. [2022], Cao et al. [2025a] and video Brooks et al. [2024], Yang et al. [2024], Cao et al. [2025b]. Evaluation on these multi-modal aspects of language models includes image generation Lin et al. [2024], Cao et al. [2025c], video generation Guo et al. [2025a,b,c], and multi-modal reasoning Xu et al. [2025b], Tie et al. [2025].

Sketching Sketching is a powerful tool that is used to accelerate the performance of machine learning algorithms and optimization processes. The fundamental concept of sketching is to partition a large input matrix into a significantly smaller sketching matrix but still preserve the main characteristics of the original matrix. Therefore, the algorithms may work with the smaller matrix instead of the huge original, which leads to a substantial reduction in processing time. Many previous works have studied sketching, proposed sketching algorithms, and supported these algorithms with robust theoretical guarantees. For example, the Johnson-Lindenstrauss lemma is proposed by Johnson and Lindenstrauss [1984]: it shows that under a certain high-dimensional space, projecting points to a lower-dimensional subspace may preserve the pairwise distances between these points. This mathematical property becomes the foundation of the development of faster algorithms for tasks such as nearest neighbor search. In addition, as explained in Ailon and Chazelle [2006], the Fast Johnson-Lindenstrauss Transform (FJLT) introduces a specific family of structured random projections that can be applied to a matrix in input sparsity time.

More recently, sketching has been applied to many numerical linear algebra tasks, such as linear regression [Clarkson and Woodruff, 2013, Nelson and Nguyên, 2013], online optimization problems [Reddy et al., 2021], training neural networks [Song et al., 2024b, Xiao et al., 2018, Song et al., 2021b, Gao et al., 2024, Brand et al., 2021], reinforcement learning [Wang et al., 2020, Xu et al., 2023], tensor decomposition [Song, 2019, Song et al., 2019, Deng et al., 2023d], relational database [Qin et al., 2022], low-rank approximation [Boutsidis and Woodruff, 2014, Makarychev et al., 2020, Meng and Mahoney, 2013, Andoni et al., 2018, Song et al., 2017], distributed problems [Boutsidis et al., 2016, Woodruff and Zhong, 2016], weighted low rank approximation [Razenshteyn et al., 2016, Gu et al., 2024, Song et al., 2025], CP decomposition [Ma and Solomonik, 2021], regression inspired by softmax [Li et al., 2023c, Gao et al., 2025c, Sinha et al., 2023, Deng et al., 2023a], and Kronecker product regression [Reddy et al., 2022].

3 TECHNIQUE OVERVIEW

In this section, we introduce the primary technique employed in this paper. This serves as a summary of our theoretical analysis, which is deferred to the Appendix due to space limitations.

Specifically, in Section 3.1, we present the key mathematical properties used to analyze the attention optimization problem, as defined in Definition 1.2. In Section 3.2, we describe the techniques for constructing and analyzing the essential properties of our main algorithm (see Algorithm 1).

3.1 THEORETICAL ANALYSIS

Big Picture In this section, we provide an overview of the key techniques used in our theoretical analysis. Our analysis of this multivariate loss function relies on a novel technique that leverages support vector machines (SVM) to reformulate the loss function:

$$\|D(X)^{-1}\exp(A_1 X A_2^\top)A_3 Y - B\|_F^2$$

into the form of inner products and Kronecker product

$$\sum_{j_0=1}^{n}\sum_{i_0=1}^{d}(\langle\langle \exp(\mathsf{A}_{j_0}x), \mathbf{1}_n\rangle^{-1}$$

$$\cdot \exp(\mathsf{A}_{j_0} x), A_3 Y_{*,i_0} \rangle - b_{j_0,i_0})^2. \quad (1)$$

We define

- $u(x)_{j_0} := \exp(\mathsf{A}_{j_0} x)$,
- $\alpha(x)_{j_0} := \langle \exp(\mathsf{A}_{j_0} x), \mathbf{1}_n \rangle$,
- $f(x)_{j_0} := \alpha(x)_{j_0}^{-1} u(x)_{j_0}$,
- $h(Y)_{i_0} := A_3 Y_{*,i_0}$, and
- $c(x,y)_{j_0,i_0} := \langle f(x)_{j_0}, h(y)_{i_0} \rangle - b_{j_0,i_0}$.

to decompose Eq. (1) into small parts and compute their gradient and Hessian respectively. Unlike prior works that focus on single-variable loss functions [Gao et al., 2025b, Deng et al., 2023b, Song et al., 2024a, Deng et al., 2023a, Gao et al., 2023a,c,b], our multivariate loss function has a more complex Hessian matrix: $H = \begin{bmatrix} H_{x,x} & H_{x,y} \\ H_{y,x} & H_{y,y} \end{bmatrix}$. We first present how we decompose the Hessian into blocks (X, Y). Then, we show that the diagonal sub Hessian matrices $H_{x,x}, H_{y,y} \in \mathbb{R}^{d^2 \times d^2}$ are positive semi-definite and provide an upper bound on the spectral norm of the off-diagonal sub Hessian matrices $H_{x,y}, H_{y,x} \in \mathbb{R}^{d^2 \times d^2}$. Next, we demonstrate that the full Hessian matrix $H \in \mathbb{R}^{2d^2 \times 2d^2}$, consisting of the sub matrices $H_{x,x}$, $H_{x,y}$, $H_{y,x}$, and $H_{y,y}$, is also positive semi-definite. Finally, we introduce techniques for proving that the Hessian is Lipschitz.

Problem Reformulation Using SVM The initial works [Deng et al., 2023a, Gao et al., 2025c, Song et al., 2024a] on attention regression problems consider the simplest ℓ_2 norm, such as $\min_{x \in \mathbb{R}^d} \| \langle \exp(Ax), \mathbf{1}_n \rangle^{-1} \exp(Ax) - c \|_2^2$ (**Part 1** of Definition 1.3), which corresponds to a single row of the full attention matrix. Inspired by the tensor trick from Diao et al. [2018, 2019],

$$\mathrm{vec}(A_1 X A_2^\top) = (A_1 \otimes A_2) \, \mathrm{vec}(X) \in \mathbb{R}^{n^2},$$

later works [Gao et al., 2023c,b] consider a slightly more complicated version of the **Part 1** equation, namely the Frobenius norm of the whole matrix, such as $\min_{X \in \mathbb{R}^{d \times d}} \| D(X)^{-1} \exp(A_1 X A_2^\top) - C \|_F^2$ (**Part 2** of Definition 1.3). In particular, instead of using a single rescaling factor (**Part 1**), we now have n rescaling factors (**Part 2**). We split $\exp((A_1 \otimes A_2) \, \mathrm{vec}(X)) \in \mathbb{R}^{n^2}$ into n chunks, each of size n, and apply the same rescaling factor within each chunk.

Remark 3.1. *For a matrix $\mathsf{A} = A_1 \otimes A_2 \in \mathbb{R}^{n^2 \times d^2}$, we can split it into n blocks, where the first block $\mathsf{A}_1 \in \mathbb{R}^{n \times d^2}$ contains the first n rows of A, the second block $\mathsf{A}_2 \in \mathbb{R}^{n \times d^2}$ contains the next n rows of A, and so on. The j_0-th block $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ contains the rows from $(j_0 - 1)n + 1$ to $j_0 n$ of A, and the n-th block $\mathsf{A}_n \in \mathbb{R}^{n \times d^2}$ contains the rows from $(n-1)n + 1$ to n^2 of A.*

Note that while the tensor trick is necessary for considering matrix norm regression, it is not sufficient to account for the value matrix $A_3 Y$ in the attention optimization problem (Definition 1.2). Therefore, we take a step further by incorporating both the SVM and the tensor trick to reformulate the entire equation of the attention optimization problem. The standard SVM objective function [Joachims, 2006, Chang and Lin, 2001, Gu et al., 2025, Tarzanagh et al., 2023] in optimization can be viewed as the product of a summation over a batch of inner products. Inspired by this, we define n functions $f(x)_{j_0} = \langle \exp(\mathsf{A}_{j_0} x), \mathbf{1}_n \rangle^{-1} \exp(\mathsf{A}_{j_0} x) \in \mathbb{R}^n$ (see Definition A.10) for each $j_0 \in [n]$ and d functions $h(Y)_{i_0} = A_3 Y_{*,i_0} \in \mathbb{R}^n$ (see Definition A.11), where $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ is one $n \times d^2$ block from A. Here, x is the vectorization of X, and y is the vectorization of Y. Then the objective function in Definition 1.2, $\| D(X)^{-1} \exp(A_1 X A_2^\top) A_3 Y - B \|_F^2$, can be turned into

$$\sum_{j_0=1}^{n} \sum_{i_0=1}^{d} (\langle f(x)_{j_0}, h(Y)_{i_0} \rangle - b_{j_0,i_0})^2 \quad (2)$$

where b_{j_0,i_0} is the entry of matrix $B \in \mathbb{R}^{n \times d}$. We call this formulation SVM-inspired formulation.

Split Hessian Into Blocks (X, Y) In the fast approximation and convergence guarantee of the training process for the attention matrix, the PSD property is a key focus in Section C. Unlike single or multiple softmax regression or their variants [Deng et al., 2023a, Gao et al., 2023b, 2025c], both the weights X and Y (as defined in Definition 1.2) need to be considered, which significantly increases the complexity of the analysis. Therefore, our Hessian matrix discussed in Section C has the following format

$$H = \begin{bmatrix} H_{x,x} & H_{x,y} \\ H_{y,x} & H_{y,y} \end{bmatrix}$$

To establish the positive semi-definite property, we will examine the properties of the matrix above individually.

Positive Semi-Definite For Hessian $H_{x,x}$, $H_{y,y}$ The positive semi-definite of $H_{x,x}, H_{y,y}$ constitutes a crucial initial step in the proof outlined in Lemma C.1. These Hessian are discussed in detail in Section F and Section G. However, proving the PSD property for $H_{x,x}$ and $H_{y,y}$ in the context of the attention optimization problem is non-trivial. The challenges arise from the complex structure of the attention mechanism and the presence of the exponential function in the loss formulation (Definition 1.2).

To tackle these challenges, we dive deep into the structure of $H_{x,x}$ and $H_{y,y}$ (see Section F and Section G for details). We express these matrices in terms of the constituent functions of the attention mechanism, such as the exponential function, the softmax function, and the key-query-value transformations. This fine-grained representation allows us to analyze the PSD property at a granular level. Another key

insight in our analysis is the role of the regularization term (see details in Section A.6) in the loss function. By carefully choosing the regularization weight, we can ensure that it dominates any potentially negative contributions from the complex attention terms. This is a delicate balancing act, as the regularization weight needs to be large enough to enforce the PSD property, but not so large that it overwhelms the attention signal [Li et al., 2023b, Deng et al., 2023a].

Leveraging this insight, we derive lower bounds on the regularization weight that guarantee the PSD property for $H_{x,x}$ and $H_{y,y}$ (Lemma G.1 and Lemma F.1 respectively). These bounds are expressed in terms of the spectral norms of the attention matrices and the minimum singular values of the key-query-value transformations. By ensuring that the regularization weight exceeds these bounds, we can provably establish the PSD property: there exists a real number $l > 0$ such that

$$H(x) = H_{x,x} \succeq l \cdot I_{d^2} \text{ and } H(y) = H_{y,y} \succeq l \cdot I_{d^2}.$$

Upper Bounds for the Spectral Norm of $H_{x,y}$, $H_{y,x}$ $H_{x,y}$ and $H_{y,x}$ blocks capture the intricate interaction between the weights X and Y in the attention mechanism. Bounding their influence is crucial for ensuring the overall positive semi-definite (PSD) property of the Hessian and the convergence of our optimization algorithm. To establish the spectral upper bound of $H_{x,y}$, we can decompose $H_{x,y}$ into $\{G_i\}_{i=1}^4$ as described in Lemma I.10. Another important technique in our analysis is the use of the boundedness properties of the attention functions. We show that the exponential function and the softmax function, when applied to bounded inputs, produce outputs with controlled spectral norms. This allows us to propagate the boundedness through the complex matrix expressions in $H_{x,y}$.

Leveraging these insights, we derive a spectral upper bound for each component in Lemma I.10, namely $\max_{i\in[n]} \|G_i\| \leq R^2$, where R is a constant that depends on the spectral norms of the attention matrices. Using these component-wise bounds, we then derive a tight spectral upper bound for the full off-diagonal block $H_{x,y}$, $\|H(x,y)\| \leq nd \cdot 10R^2$ Given this upper bound, our final focus in the proof of the positive semi-definite property (PSD) will be as follows.

PSD for Hessian H The challenge in establishing the PSD property for H lies in the complex interplay between its constituent blocks: $H_{x,x}$, $H_{x,y}$, $H_{y,x}$, and $H_{y,y}$. Each of these blocks has its own intricate structure, involving the attention matrices, the exponential function, and the softmax normalization. Moreover, the off-diagonal blocks $H_{x,y}$ and $H_{y,x}$ introduce cross-term interactions that can potentially disrupt the PSD property.

To tackle this challenge, we employ a carefully orchestrated analysis that leverages the properties of the individual blocks and their interrelationships. Our strategy is to show that the PSD property of the diagonal blocks $H_{x,x}$ and $H_{y,y}$ is strong enough to compensate for any potentially negative contributions from the off-diagonal blocks.

With the PSD property of the diagonal blocks and the spectral bounds on the off-diagonal blocks in hand, we then embark on the final step of proving the PSD property for the full Hessian H through using

$$\begin{aligned} &\begin{bmatrix} u^\top & v^\top \end{bmatrix} H \begin{bmatrix} u \\ v \end{bmatrix} \\ &= u^\top H_{x,x} u + v^\top H_{y,y} v + u^\top H_{x,y} v + v^\top H_{y,x} u, \end{aligned}$$

for any arbitrary $u, v \in \mathbb{R}^{d^2}$.

Consequently, based on the positive semi-definite property of the diagonal matrix, the computation of the off-diagonal part of the matrix does not affect the positivity of the entire matrix, thereby establishing a positive semi-definite. With $\alpha_1, \alpha_2, \alpha_3$ as the bound of the matrix above respectively in Lemma C.1, we have the following result

$$H \succeq \min\{\alpha_1 - \alpha_3, \alpha_2 - \alpha_3\} \cdot I_{2d^2}$$

Given the relationship of $\{a_i\}_{i=1}^3$ as discussed above, the positive semi-definite property of the Hessian matrix is established.

Lipschitz Property for Hessian The Lipschitz property of the Hessian is determined by the upper bound and Lipschitz property of the basic functions that constitute the Hessian matrix H. Since H has three parts $H_{x,x}$, $H_{x,y}$ and $H_{y,y}$. In Section G, due to $H(y)$ is independent of y, the Lipschitz property can be easily established. For details of others, we refer the readers to read Section I.

To compute the Lipschitz continuity of $H_{x,x}$, we begin by providing a brief explanation. In our proof, we first establish upper bounds for the functions $u(x)$, $c(x)$, and $f(x)$ in Lemma E.4, which together form the matrix $H_{x,x}$ (as detailed in Section A.3). Importantly, we ensure that these basic functions possess the Lipschitz property in Lemma E.5. Using the foundational components mentioned above, we can decompose $H_{x,x}$ into 4 distinct parts denoted as $\{G_k\}_{k=1}^4$. We will leverage the Lipschitz property of the basic functions above and a method introduced below. The following task is extensively involved in the Lipschitz proof (for each G_k), we want to bound $|\prod_{i=1}^t \beta_i(x) - \prod_{i=1}^t \beta_i(\widetilde{x})|$, which has an upper bound as:

$$\sum_{j=0}^{t-1} |\prod_{i=0}^{j} \beta_i(\widetilde{x}) \prod_{i'=j+1}^{t} \beta_{i'}(x) - \prod_{i=1}^{j+1} \beta_i(\widetilde{x}) \prod_{i'=j+2}^{t+1} \beta_{i'}(x)|$$

where assume that $\beta_0(x) = 1$ and $\beta_{t+1}(x) = 1$ for convenience. We will then proceed to establish the Lipschitz continuity of $H_{x,x}$

$$\sum_{k=1}^{K} \|G_k(x,y) - G_k(\widetilde{x},\widetilde{y})\| \leq n^{1.5} \exp(20R^2)(\|x-\widetilde{x}\|_2 + \|y-\widetilde{y}\|_2)$$

3.2 ALGORITHM

Algorithm 1 Our Algorithm

procedure TRAININGALGORITHM(A_1, A_2, A_3) ▷ Theorem 1.4
  Let $x(0), y(0) \in \mathbb{R}^{d^2}$ denote initialization point.
  for $t = 0 \to T-1$ **do**
    /*Forward*/
    Compute $h(y(t)) \in \mathbb{R}^{n \times d}$ ▷ $\mathcal{T}_{\mathrm{mat}}(n,d,d)$ time
    Compute $f(x(t)) \in \mathbb{R}^{n \times n}$ ▷ $\mathcal{T}_{\mathrm{mat}}(n,d,n)$ time
    Compute $c(x(t), y(t)) \in \mathbb{R}^{n \times d}$ (based on $f(x(t)), h(y(t))$) ▷ $\mathcal{T}_{\mathrm{mat}}(n,d,d)$ time
    /*Gradient*/
    Compute $g(x(t))$ based on Lemma B.4 ▷ $\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$ time
    Compute $g(y(t))$ based on Lemma B.5 ▷ $\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$ time
    /*Hessian*/
    Compute $\widetilde{H}$ via TensorSRHT ▷ $\widetilde{O}(nd + d^{2\omega})$
    /*Update*/
    $\begin{bmatrix} x(t+1) \\ y(t+1) \end{bmatrix} \leftarrow \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} - \begin{bmatrix} g(x(t)) \\ g(y(t)) \end{bmatrix} \widetilde{H}^{-1}$ ▷ $O(d^{2\omega})$
  end for
  return $\begin{bmatrix} x(T) \\ y(T) \end{bmatrix}$
end procedure

In this section, we present the techniques for constructing and analyzing the properties of our algorithm (see Algorithm 1). First, we present our technique for simplifying the computation of the attention matrix. Then, we display the techniques for the gradient and Hessian computation. After that, we delve into the primary contribution of our work: TensorSRHT fast approximation for Hessian. Finally, we combine the running time of all of the previous parts (forward function, gradient, Hessian, inverse of approximate Hessian) and conclude the total running time of our algorithm (see Algorithm 1).

Forward Computation To simplify the computation of the attention matrix, we can decompose the computation process into three components: f, c, and h as defined in Section A.2. The forward computation can then be completed in $O(\mathcal{T}_{\mathrm{mat}}(n,d,d) + \mathcal{T}_{\mathrm{mat}}(n,n,d))$ time, as stated in Lemma B.3.

Gradient Computation We can compute the gradient in Section B as follows:

$$\frac{\mathrm{d}L(x,y)}{\mathrm{d}x} = \mathrm{vec}(A_1^\top p(x,y) A_2),$$

for some matrix $p(x,y) \in \mathbb{R}^{n \times n}$. Here $A_1^\top p(x,y) A_2$ can be computed in $\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(d,n,d)$ time. Similarly,

$$\frac{\mathrm{d}L(x,y)}{\mathrm{d}y} = \mathrm{vec}(A_3^\top \widetilde{q}(x,y)),$$

which also takes $\mathcal{T}_{\mathrm{mat}}(n,n,d) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$ time. We will now establish the overall running time for gradient computation. By utilizing the results from Lemma B.4 and Lemma B.5, we can efficiently compute the gradients of $g(x(t))$ and $g(y(t))$ in $\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$ time.

Straightforward Hessian Computation Computing the Hessian in straightforward way would take $\mathcal{T}_{\mathrm{mat}}(d^2, n^2, d^2)$ time, because we need to explicitly write down $\mathsf{A}^\top \mathsf{A} \in \mathbb{R}^{d^2 \times d^2}$ where $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$. This is too slow, we use sketching ideas to speed up this running time. Using sketching matrices to speed up the Hessian computation has been extensively studied in convex and non-convex optimization [Jiang et al., 2021, Lee et al., 2019, Song and Yu, 2021, Gu and Song, 2022, Gu et al., 2025, Qin et al., 2023b].

TensorSRHT Fast Approximation for Hessian Now, let's delve into the key contribution of this paper. Given that $\mathsf{A} = A_1 \otimes A_2 \in \mathbb{R}^{n^2 \times d^2}$, the time complexity of regression becomes prohibitively expensive. Our contribution aims to execute a fast approximation to significantly reduce the time complexity when using the Newton Method. We construct our TensorSRHT sketching matrix $S \in \mathbb{R}^{m \times n^2}$ by

$$S = \frac{1}{\sqrt{m}} P \cdot (QD_1 \otimes QD_2),$$

where $P \in \{0,1\}^{m \times n^2}$ contains only one 1 at a random coordinate, Q is a $n \times n$ Hadamard matrix, and D_1, D_2 are two $n \times n$ independent diagonal matrices with diagonals that are each independently set to be a Rademacher random variable (uniform in $\{-1,1\}$). We choose $m = O(\epsilon^{-2} d^2 \log^3(nd/\epsilon\delta)) \ll n^2$, where $\epsilon > 0$ is the accuracy parameter and $\delta \in (0,1)$ is the failure probability, so $S\mathsf{A} \in \mathbb{R}^{m \times d^2}$ is a much smaller matrix compared with $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$. Therefore, using $S\mathsf{A}$, we can construct a sparse Hessian. This reduces the time from $\mathcal{T}_{\mathrm{mat}}(d^2, n^2, d^2)$ down to $\widetilde{O}(nd) + \mathcal{T}_{\mathrm{mat}}(d^2, d^2, d^2)$[2]. Additionally, Ahle et al. [2020], Song et al. [2021a] show that with $m = O(\epsilon^{-2} d^2 \log^3(nd/\epsilon\delta))$, the TensorSRHT sketching matrix S is an oblivious subspace embedding, which may further implies that with high probability $(1-\delta)$, the sketched Hessian $\widetilde{H}$ approximates the true Hessian H with bounded error in terms of ϵ.

[2]We consider the regime $n \gg d$ in the paper which is the most common setting in practice because n is the length of the document, and d is feature dimension.

Overall Time Building upon the aforementioned properties, we can apply the Newton Method in Section K to establish convergence for the regression problem. In Summary, we know that

- Computing forward function $\mathcal{T}_{\mathrm{mat}}(n,n,d) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$ time (Lemma B.3)
- Computing gradient takes $\mathcal{T}_{\mathrm{mat}}(n,n,d)+\mathcal{T}_{\mathrm{mat}}(n,d,d)$ time (Lemma B.4 and Lemma B.5)
- Compute Hessian takes $\widetilde{O}(nd) + \mathcal{T}_{\mathrm{mat}}(d^2,d^2,d^2)$ (Lemma J.6)
- Compute g times inverse of approximate Hessian, this can be done in $\mathcal{T}_{\mathrm{mat}}(d^2,d^2,d^2) = d^{2\omega}$

The total time can be expressed as $\widetilde{O}(\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(n,d,d) + d^{2\omega})\log(1/\epsilon)$, for $\omega \approx 2.37$.

4 DISCUSSION

Attention Formulation. In this paper, our attention formulation in Definition 1.1 exactly matches the softmax attention in the traditional notation system Vaswani et al. [2017], with only some basic notational differences. Specifically, recalling Definition 1.1, we compute the query-key attention matrix as $D^{-1}\underbrace{\exp(X_\ell Q K^\top X_\ell^\top)}_{:=A}$, where $D^{-1}A$ recovers the computation $\mathrm{Softmax}(\frac{\widetilde{Q}\widetilde{K}^\top}{\sqrt{d}})$ (with $\widetilde{Q} := X_\ell W_Q$, $\widetilde{K} := X_\ell W_K$) in Vaswani et al. [2017].

The key difference is that we use Q and K to denote W_Q and W_K, and we use A to denote the numerator part of the softmax computation in each row, while D^{-1} normalizes each row. This means that our theoretical result is highly practical, with perfect alignment to the Transformers used in real LLMs.

Generalization to Multi-Layer Attention. Our main result in Theorem 1.4 can be easily generalized to the multilayer case. To show this, we first consider the recursive attention computation in Definition 1.1:

$$X_{\ell+1} \leftarrow D^{-1}\exp(X_\ell Q K^\top X_\ell^\top) X_\ell V,$$

where each layer computes its output based on the previous layer's input and the weight matrices.

In this paper, our result states that given any arbitrary X_ℓ we treat the weights $QK^\top$ and V as variables, and we can output a good approximation of $X_{\ell+1}$ denoted as (see Definition 1.2). In another work Deng et al. [2023c], they treat the input X_ℓ as a variable and study the training. Since our formulation and algorithm treat X_ℓ as an input and do not assume anything specific about its origin, and we can directly combine our result with attention training in Deng et al. [2023c], our results apply to any layer in the network. Therefore, our methods naturally extend to multi-layer attention by applying them iteratively at each layer.

Justification of Assumptions. In this work, our goal is to design an efficient algorithm that can be applied to a broader range of modern transformer architectures. Consequently, our method does not rely on strict assumptions, requiring only assumptions on good initialization points of x_0 and y_0 (see Definition K.1) and on the regularization term $\|(W \otimes I)(A_1 \otimes A_2)x\|_2^2 + \|WA_3 y\|_F^2$ in Definition A.14.

Both assumptions can be easily satisfied in practice. Specifically, the first assumption can be met by spending additional effort in selecting a suitable initialization point, while the second is a standard practice in attention optimization Gao et al. [2025c], Li et al. [2023b] and widely accepted in the broader field of optimization. These assumptions are also weaker than those in previous works, as we do not rely on conditions such as $d = O(\log n)$, $d = o(\log^2 n)$, or bounded entry assumptions as in Alman and Song [2023], Zandieh et al. [2023], nor do we overly simplify the problem as in Song et al. [2024a], Gao et al. [2025c], Deng et al. [2023a].

5 CONCLUSION

In this paper, we make several important contributions to optimizing attention mechanisms in LLMs by providing the first complete analysis of an unsimplified single-layer attention optimization problem. Unlike previous work that simplified the problem by fixing certain components, our work treats all weight matrices Q, K, V as variables, offering a more comprehensive theoretical understanding. We introduce a novel approach that combines tensor tricks and SVM-inspired formulation to reformulate the attention optimization problem in a more tractable way. This reformulation allows us to develop new theoretical insights while maintaining the full complexity of the attention mechanism. Our main technical achievement is developing an algorithm that can solve the attention optimization problem up to ϵ accuracy in $\widetilde{O}((\mathcal{T}_{\mathrm{mat}}(n,n,d)+\mathcal{T}_{\mathrm{mat}}(n,d,d)+d^{2\omega})\log(1/\epsilon))$ time, where $\mathcal{T}_{\mathrm{mat}}$ represents matrix multiplication time, n is the sequence length, d is the embedding dimension, and $\omega \approx 2.37$ is the matrix multiplication exponent. These guarantees are established through careful analysis of the positive semi-definite properties of the Hessian matrix, Lipschitz continuity of the Hessian, and the application of TensorSRHT techniques for fast approximation.

In conclusion, we provide theoretical insights into attention optimization and present a concrete algorithm with provable guarantees. While the immediate practical applications may be limited by the single-layer constraint, the analytical techniques and theoretical framework developed here could serve as building blocks for future work on more complex attention architectures.

Acknowledgements

The authors would like to thank the anonymous reviewer of UAI 2025 for their highly insightful suggestions.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, et al. Benchmarking Arabic AI with large language models. In *EACL*, 2024.

Thomas D Ahle, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *SODA*, 2020.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. MEGA: Multilingual evaluation of generative AI. In *EMNLP*, 2023.

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *STOC*, 2006.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *ICML*, 2019a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the convergence rate of training recurrent neural networks. In *NeurIPS*, 2019b.

Josh Alman and Zhao Song. Fast attention requires bounded entries. In *NeurIPS*, 2023.

Josh Alman and Zhao Song. The fine-grained complexity of gradient computation for training large language models. In *NeurIPS*, 2024a.

Josh Alman and Zhao Song. How to capture higher-order correlations? generalizing matrix softmax attention to kronecker computation. In *ICLR*, 2024b.

Josh Alman and Zhao Song. Fast rope attention: Combining the polynomial method and fast fourier transform. *arXiv preprint arXiv:2505.11892*, 2025a.

Josh Alman and Zhao Song. Only large weights (and not skip connections) can prevent the perils of rank collapse. *arXiv preprint arXiv:2505.16284*, 2025b.

Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *SODA*, 2021.

Josh Alman, Zhao Song, Ruizhe Zhang, and Danyang Zhuo. Bypass exponential time preprocessing: Fast neural network training via weight-data correlation preprocessing. In *NeurIPS*, 2024.

Zaid Alyafeai, Maged S Alshaibani, Badr AlKhamissi, Hamzah Luqman, Ebrahim Alareqi, and Ali Fadel. Taqyim: Evaluating Arabic NLP tasks using ChatGPT models. *arXiv preprint arXiv:2306.16322*, 2023.

Ehsan Amid and Manfred K Warmuth. Winnowing with gradient descent. In *COLT*, 2020a.

Ehsan Amid and Manfred KK Warmuth. Reparameterizing mirror descent as gradient descent. In *NeurIPS*, 2020b.

Alexandr Andoni, Chengyu Lin, Ying Sheng, Peilin Zhong, and Ruiqi Zhong. Subspace embedding and linear regression with orlicz norm. In *ICML*, 2018.

Daman Arora, Himanshu Gaurav Singh, and Mausam. Have LLMs advanced enough? a challenging problem solving benchmark for large language models. In *EMNLP*, 2023.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *ICML*, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *NeurIPS*, 2019b.

Navid Azizan, Sahin Lale, and Babak Hassibi. Stochastic mirror descent on overparameterized nonlinear models. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *AACL*, 2023.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, and Ben He. ChatGPT is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. In *COLING*, 2024.

Song Bian, Zhao Song, and Junze Yin. Federated empirical risk minimization via second-order method. *arXiv preprint arXiv:2305.17482*, 2023.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *COLT*, 2020.

Christos Boutsidis and David P Woodruff. Optimal CUR matrix decompositions. In *STOC*, 2014.

Christos Boutsidis, David P Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *STOC*, 2016.

Jan den van Brand and Zhao Song. A $\sqrt{n}$ passes streaming algorithm for solving bipartite matching exactly. *Manuscript*, 2023.

Jan van den Brand. A deterministic linear program solver in current matrix multiplication time. In *SODA*, 2020.

Jan van den Brand, Binghui Peng, Zhao Song, and Omri Weinstein. Training (overparametrized) neural networks in near-linear time. In *ITCS*, 2021.

Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. In *ICML*, 2024.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*. Springer Science & Business Media, 1997.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *ICLR*, 2023.

HanQin Cai, Yuchen Lou, Daniel McKenzie, and Wotao Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In *ICML*, 2021.

Tianle Cai, Ruiqi Gao, Jikai Hou, Siyu Chen, Dong Wang, Di He, Zhihua Zhang, and Liwei Wang. Gram-gauss-newton method: Learning overparameterized neural networks for regression problems. *arXiv preprint arXiv:1905.11675*, 2019.

Yang Cao and Zhao Song. Sorsa: Singular values and orthonormal regularized singular vectors adaptation of large language models. *arXiv preprint arXiv:2409.00055*, 2025.

Yang Cao, Xiaoyu Li, and Zhao Song. Grams: Gradient descent with adaptive momentum scaling. *arXiv preprint arXiv:2412.17107*, 2024.

Yang Cao, Bo Chen, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, Zhao Song, and Mingda Wan. Force matching with relativistic constraints: A physics-inspired approach to stable and efficient generative modeling. *arXiv preprint arXiv:2502.08150*, 2025a.

Yang Cao, Zhao Song, and Chiwun Yang. Video latent flow matching: Optimal polynomial projections for video interpolation and extrapolation. *arXiv preprint arXiv:2502.00500*, 2025b.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. Assessing cross-cultural alignment between ChatGPT and human societies: An empirical study. *arXiv preprint arXiv:2303.17466*, 2023.

Yuefan Cao, Xuyang Guo, Jiayan Huo, Yingyu Liang, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Zhen Zhuang. Text-to-image diffusion models cannot count, and prompt refinement cannot help. *arXiv preprint arXiv:2503.06884*, 2025c.

Cayque Monteiro Castro Nascimento and André Silva Pimentel. Do large language models understand chemistry? a conversation with ChatGPT. *Journal of Chemical Information and Modeling*, 2023.

Chih-Chung Chang and Chih-Jen Lin. Training v-support vector classifiers: theory and algorithms. *Neural computation*, 2001.

Beidi Chen, Zichang Liu, Binghui Peng, Zhaozhuo Xu, Jonathan Lingjie Li, Tri Dao, Zhao Song, Anshumali Shrivastava, and Christopher Re. Mongoose: A learnable lsh framework for efficient neural network training. In *ICLR*, 2021.

Bo Chen, Xiaoyu Li, Yingyu Liang, Jiangxuan Long, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for RoPE-based transformer architecture. *arXiv preprint arXiv:2411.07602*, 2024a.

Bo Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Bypassing the exponential dependency: Looped transformers efficiently learn in-context by multi-step gradient descent. In *AISTATS*, 2025a.

Bo Chen, Zhenmei Shi, Zhao Song, and Jiahao Zhang. Provable failure of language models in learning majority boolean logic via gradient descent. *arXiv preprint arXiv:2504.04702*, 2025b.

Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv preprint arXiv:2406.05370*, 2024b.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *IJCNLP*, 2023.

Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. The computational limits of state-space models and mamba via the lens of circuit complexity. In *CPAL*, 2025c.

Yifang Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Fundamental limits of visual autoregressive transformers: Universal approximation abilities. In *ICML*, 2025d.

Joseph Chervenak, Harry Lieman, Miranda Blanco-Breindel, and Sangita Jindal. The promise and peril of using a large language model to obtain clinical information: ChatGPT performs strongly as a fertility counseling tool with limitations. *Fertility and Sterility*, 2023.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. Instructeval: Towards holistic evaluation of instruction-tuned large language models. *arXiv preprint arXiv:2306.04757*, 2023.

Minje Choi, Jiaxin Pei, Sagar Kumar, Chang Shu, and David Jurgens. Do LLMs understand social knowledge? evaluating the sociability of large language models with socket benchmark. In *EMNLP*, 2023.

Matthias Christandl, François Le Gall, Vladimir Lysikov, and Jeroen Zuiddam. Barriers for rectangular matrix multiplication. *Computational complexity*, 2025.

Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. In *STOC*, 2013.

Michael B Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *SODA*, 2016.

Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *STOC*, 2019.

Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 1982.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Alex Damian, Tengyu Ma, and Jason D Lee. Label noise SGD provably prefers flat global minimizers. In *NeurIPS*, 2021.

Xuan-Quy Dao and Ngoc-Bich Le. Investigating the effectiveness of ChatGPT in mathematical reasoning and problem solving: Evidence from the vietnamese national high school graduation examination. *arXiv preprint arXiv:2306.06331*, 2023.

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

Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. *arXiv preprint arXiv:2304.04397*, 2023b.

Yichuan Deng, Zhao Song, Shenghao Xie, and Chiwun Yang. Unmasking transformers: A theoretical approach to data recovery via attention weights. *arXiv preprint arXiv:2310.12462*, 2023c.

Yichuan Deng, Zhao Song, and Junze Yin. Faster robust tensor power method for arbitrary order. *arXiv preprint arXiv:2306.00406*, 2023d.

Yichuan Deng, Zhao Song, Lichen Zhang, and Ruizhe Zhang. Efficient algorithm for solving hyperbolic programs. *arXiv preprint arXiv:2306.07587*, 2023e.

Yichuan Deng, Zhihang Li, Sridhar Mahadevan, and Zhao Song. Zero-th order algorithm for softmax attention optimization. *Big Data*, 2024.

Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. How ready are pre-trained abstractive models and LLMs for legal case judgement summarization? *arXiv preprint arXiv:2306.01248*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Huaian Diao, Zhao Song, Wen Sun, and David Woodruff. Sketching for kronecker product regression and p-splines. In *AISTATS*, 2018.

Huaian Diao, Rajesh Jayaram, Zhao Song, Wen Sun, and David Woodruff. Optimal sketching for kronecker product regression and low rank approximation. In *NeurIPS*, 2019.

Ye Dong, Wen-jie Lu, Yancheng Zheng, Haoqi Wu, Derun Zhao, Jin Tan, Zhicong Huang, Cheng Hong, Tao Wei, and Wenguang Cheng. Puma: Secure inference of llama-7b in five minutes. *arXiv preprint arXiv:2307.12533*, 2023.

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

Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *FOCS*, 2023.

Emilio Ferrara. Should ChatGPT be biased? challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*, 2023.

Michael C Frank. Baby steps in evaluating the capacities of large language models. *Nature Reviews Psychology*, 2023.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *ICML*, 2023.

François Le Gall and Florent Urrutia. Improved rectangular matrix multiplication using powers of the coppersmith-winograd tensor. In *SODA*, 2018.

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

Yeqi Gao, Zhao Song, and Shenghao Xie. In-context learning for attention scheme: from single softmax regression to multiple softmax regression via a tensor trick. *arXiv preprint arXiv:2307.02419*, 2023b.

Yeqi Gao, Zhao Song, and Junze Yin. Gradientcoin: A peer-to-peer decentralized large language models. *arXiv preprint arXiv:2308.10502*, 2023c.

Yeqi Gao, Lianke Qin, Zhao Song, and Yitan Wang. A sublinear adversarial training algorithm. In *ICLR*, 2024.

Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in llm based on tensor and svm trick, and solving it in matrix multiplication time. In *UAI*, 2025a.

Yeqi Gao, Zhao Song, Xin Yang, Ruizhe Zhang, and Yufa Zhou. Fast quantum algorithm for attention computation. In *QIP*, 2025b.

Yeqi Gao, Zhao Song, and Junze Yin. An iterative algorithm for rescaled hyperbolic functions regression. In *AISTATS*, 2025c.

Yuzhou Gu and Zhao Song. A faster small treewidth sdp solver. *arXiv preprint arXiv:2211.06033*, 2022.

Yuzhou Gu, Zhao Song, Junze Yin, and Lichen Zhang. Low rank matrix completion via robust alternating minimization in nearly linear time. In *ICLR*, 2024.

Yuzhou Gu, Zhao Song, and Lichen Zhang. Faster algorithms for structured linear and kernel support vector machines. In *ICLR*, 2025.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *ICML*, 2018.

Taicheng Guo, Kehan Guo, Zhengwen Liang, Zhichun Guo, Nitesh V Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. In *NeurIPS*, 2023.

Xuyang Guo, Zekai Huang, Jiayan Huo, Yingyu Liang, Zhenmei Shi, Zhao Song, and Jiahao Zhang. Can you count to nine? a human evaluation benchmark for counting limits in modern text-to-video models. *arXiv preprint arXiv:2504.04051*, 2025a.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025b.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vtextbench: A human evaluation benchmark for textual control in video generation models. *arXiv preprint arXiv:2505.04946*, 2025c.

Thilo Hagendorff and Sarah Fabi. Human-like intuitive behavior and reasoning biases emerged in language models–and disappeared in chatgpt. *Nature computational science*, 2023.

Jeff Z HaoChen, Colin Wei, Jason Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. In *COLT*, 2021.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *NeurIPS*, 2023.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *NAACL*, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Jerry Yao-Chieh Hu, Weimin Wu, Zhuoru Li, Sophia Pi, , Zhao Song, and Han Liu. On statistical rates and provably efficient criteria of latent diffusion transformers (DiTs). In *NeurIPS*, 2024.

Jerry Yao-Chieh Hu, Hude Liu, Hong-Yu Chen, Weimin Wu, and Han Liu. Universal approximation with softmax attention. *arXiv preprint arXiv:2504.15956*, 2025a.

Jerry Yao-Chieh Hu, Maojiang Su, En-Jui Kuo, Zhao Song, and Han Liu. Computational limits of low-rank adaptation (lora) fine-tuning for transformer models. In *ICLR*, 2025b.

Jerry Yao-Chieh Hu, Wei-Po Wang, Ammar Gilani, Chenyang Li, Zhao Song, and Han Liu. Fundamental limits of prompt tuning transformers: Universality, capacity and efficiency. In *ICLR*, 2025c.

Jerry Yao-Chieh Hu, Weimin Wu, Yi-Chen Lee, Yu-Chao Huang, Minshuo Chen, and Han Liu. On statistical rates of conditional diffusion transformers: Approximation, estimation and minimax optimality. In *ICLR*, 2025d.

Jerry Yao-Chieh Hu, Xiwen Zhang, Maojiang Su, Zhao Song, and Han Liu. Minimalist softmax attention provably learns constrained boolean functions. *arXiv preprint arXiv:2505.19531*, 2025e.

Baihe Huang, Xiaoxiao Li, Zhao Song, and Xin Yang. Fl-ntk: A neural tangent kernel-based framework for federated learning analysis. In *ICML*, 2021.

Baihe Huang, Shunhua Jiang, Zhao Song, Runzhou Tao, and Ruizhe Zhang. Solving SDP faster: A robust IPM framework and efficient implementation. In *FOCS*, 2022.

Sophie Huiberts, Yin Tat Lee, and Xinzhi Zhang. Upper and lower bounds on the smoothed complexity of the simplex method. In *STOC*, 2023.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *COLT*, 2019.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. In *ICLR*, 2020a.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. In *NeurIPS*, 2020b.

Ziwei Ji and Matus Telgarsky. Characterizing the implicit bias via a primal-dual analysis. In *ALT*, 2021.

Ziwei Ji, Miroslav Dudik, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *COLT*, 2020.

Ziwei Ji, Nathan Srebro, and Matus Telgarsky. Fast margin maximization via dual acceleration. In *ICML*, 2021.

Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *FOCS*, 2020a.

Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *STOC*, 2020b.

Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. A faster algorithm for solving general lps. In *STOC*, 2021.

Thorsten Joachims. Training linear svms in linear time. In *KDD*, 2006.

Douglas Johnson, Rachel Goodman, J Patrinely, Cosby Stone, Eli Zimmerman, Rebecca Donald, Sam Chang, Sean Berkowitz, Avni Finn, Eiman Jahangir, et al. Assessing the accuracy and reliability of ai-generated medical responses: an evaluation of the chat-gpt model. *Research square*, 2023.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 1984.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. *arXiv preprint arXiv:2403.03100*, 2024.

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

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

Yekun Ke, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Circuit complexity bounds for visual autoregressive model. *arXiv preprint arXiv:2501.04299*, 2025.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. In *ICLR*, 2025.

Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. In *NeurIPS*, 2021.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020.

Anastasis Kratsios, Behnoosh Zamanlooy, Tianlin Liu, and Ivan Dokmanić. Universal approximation under constraints is possible with transformers. In *ICLR*, 2022.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. In *EMNLP*, 2023.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *ACL*, 2023.

François Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC*, 2014.

François Le Gall. Faster rectangular matrix multiplication by combination loss analysis. In *SODA*, 2024.

Jason D Lee, Ruoqi Shen, Zhao Song, Mengdi Wang, et al. Generalized leverage score sampling for neural networks. In *NeurIPS*, 2020.

Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *FOCS*, 2015.

Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *COLT*, 2019.

Chenyang Li, Yingyu Liang, Zhenmei Shi, Zhao Song, and Tianyi Zhou. Fourier circuits in neural networks and transformers: A case study of modular arithmetic with multiple inputs. In *AISTATS*, 2025a.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

Xiaoyu Li, Yingyu Liang, Zhenmei Shi, Zhao Song, Wei Wang, and Jiahao Zhang. On the computational capability of graph neural networks: A circuit complexity bound perspective. *arXiv preprint arXiv:2501.06444*, 2025b.

Xinzhe Li, Ming Liu, Shang Gao, and Wray Buntine. A survey on out-of-distribution evaluation of neural nlp models. In *IJCAI*, 2023a.

Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *NeurIPS*, 2018.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *NeurIPS*, 2019.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. How do transformers learn topic structure: Towards a mechanistic understanding. In *ICML*, 2023b.

Zhihang Li, Zhao Song, and Tianyi Zhou. Solving regularized exp, cosh and sinh regression problems. *arXiv preprint arXiv:2303.15725*, 2023c.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss?–a mathematical framework. In *ICLR*, 2022.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *ICLR*, 2024.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, et al. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. In *ICLR*, 2023d.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *The Annals of Statistics*, 2020.

Yingyu Liang, Jiangxuan Long, Zhenmei Shi, Zhao Song, and Yufa Zhou. Beyond linear approximations: A novel pruning approach for attention matrix. In *ICLR*, 2025.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, 2024.

Dongqi Liu and Vera Demberg. ChatGPT vs human-authored text: Insights into controllable text summarization and sentence style transfer. *arXiv preprint arXiv:2306.07799*, 2023.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *ICLR*, 2024.

Hude Liu, Jerry Yao-Chieh Hu, Zhao Song, and Han Liu. Attention mechanism, max-affine partition, and universal approximation. *arXiv preprint arXiv:2504.19901*, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *NeurIPS*, 2023a.

S. Cliff Liu, Zhao Song, Hengjie Zhang, Lichen Zhang, and Tianyi Zhou. Space-efficient interior point method, with applications to linear programming and maximum weight bipartite matching. In *ICALP*, 2023b.

Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In *NeurIPS*, 2013.

Linjian Ma and Edgar Solomonik. Fast and accurate randomized algorithms for low-rank tensor decompositions. In *NeurIPS*, 2021.

Konstantin Makarychev, Aravind Reddy, and Liren Shan. Improved guarantees for k-means++ and k-means++ parallel. In *NeurIPS*, 2020.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *NeurIPS*, 2023.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022.

Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *STOC*, 2013.

Edward Moroshko, Blake E Woodworth, Suriya Gunasekar, Jason D Lee, Nati Srebro, and Daniel Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. In *NeurIPS*, 2020.

Alexander Munteanu, Simon Omlor, Zhao Song, and David Woodruff. Bounding the width of neural networks via coupled initialization a worst case analysis. In *ICML*, 2022.

Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Pedro Henrique Pamplona Savarese, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. In *AISTATS*, 2019.

John J Nay, David Karamardian, Sarah B Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H Choi, and Jungo Kasai. Large language models as tax attorneys: A case study in legal capabilities emergence. *arXiv preprint arXiv:2306.07075*, 2023.

Jelani Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *FOCS*, 2013.

OpenAI. Optimizing language models for dialogue, 2022.

OpenAI. GPT-4 technical report, 2023.

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

Rasmus Pagh. Compressed matrix multiplication. *TOCT*, 2013.

Vishal Pallagani, Bharath Muppasani, Keerthiram Murugesan, Francesca Rossi, Biplav Srivastava, Lior Horesh, Francesco Fabiano, and Andrea Loreggia. Understanding the capabilities of large language models for automated planning. *arXiv preprint arXiv:2305.16151*, 2023.

Qian Qian and Xiaoyuan Qian. The implicit bias of adagrad on separable data. In *NeurIPS*, 2019.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is ChatGPT a general-purpose natural language processing task solver? In *EMNLP*, 2023a.

Lianke Qin, Rajesh Jayaram, Elaine Shi, Zhao Song, Danyang Zhuo, and Shumo Chu. Adore: Differentially oblivious relational database operators. In *VLDB*, 2022.

Lianke Qin, Zhao Song, Lichen Zhang, and Danyang Zhuo. An online and unified algorithm for projection matrix vector multiplication with application to empirical risk minimization. In *AISTATS*, 2023b.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Kovid Rathee. Meet google meena, 2020.

Ilya Razenshteyn, Zhao Song, and David P Woodruff. Weighted low rank approximations with provable guarantees. In *STOC*, 2016.

Aravind Reddy, Ryan A Rossi, Zhao Song, Anup Rao, Tung Mai, Nedim Lipka, Gang Wu, Eunyee Koh, and Nesreen Ahmed. Online map inference and learning for nonsymmetric determinantal point processes. *arXiv preprint arXiv:2111.14674*, 2021.

Aravind Reddy, Zhao Song, and Lichen Zhang. Dynamic tensor product regression. In *NeurIPS*, 2022.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of BERT. In *NeurIPS*, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *FOCS*, 2006.

Ritwik Sinha, Zhao Song, and Tianyi Zhou. A mathematical abstraction for balancing the trade-off between creativity and reality in large language models. *arXiv preprint arXiv:2306.02295*, 2023.

Charlie Snell, Ruiqi Zhong, Dan Klein, and Jacob Steinhardt. Approximating how single head attention learns. *arXiv preprint arXiv:2103.07601*, 2021.

Zhao Song. *Matrix theory: optimization, concentration, and algorithms*. PhD thesis, The University of Texas at Austin, 2019.

Zhao Song and Zheng Yu. Oblivious sketching-based central path method for solving linear programming problems. In *ICML*, 2021.

Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise ℓ_1-norm error. In *STOC*, 2017.

Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *SODA*, 2019.

Zhao Song, David Woodruff, Zheng Yu, and Lichen Zhang. Fast sketching of polynomial kernels of polynomial degree. In *ICML*, 2021a.

Zhao Song, Shuo Yang, and Ruizhe Zhang. Does preprocessing help training over-parameterized neural networks? In *NeurIPS*, 2021b.

Zhao Song, Zhaozhuo Xu, and Lichen Zhang. Speeding up sparsification using inner product search data structures. *arXiv preprint arXiv:2204.03209*, 2022.

Zhao Song, Mingquan Ye, and Lichen Zhang. Streaming semidefinite programs: $o(\sqrt{n})$ passes, small space and fast runtime. *Manuscript*, 2023.

Zhao Song, Junze Yin, and Lichen Zhang. Solving attention kernel regression problem via pre-conditioner. In *AISTATS*, 2024a.

Zhao Song, Lichen Zhang, and Ruizhe Zhang. Training multi-layer over-parametrized neural network in subquadratic time. In *ITCS*, 2024b.

Zhao Song, Mingquan Ye, Junze Yin, and Lichen Zhang. Efficient alternating minimization with applications to weighted low rank approximation. In *ICLR*, 2025.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 2018.

Jared Spataro. Introducing microsoft 365 copilot – your copilot for work, 2023.

Giriprasad Sridhara, Ranjani H. G., and Sourav Mazumdar. ChatGPT: A study on its utility for ubiquitous software engineering tasks. *arXiv preprint arXiv:2305.16837*, 2023.

Haoyuan Sun, Kwangjun Ahn, Christos Thrampoulidis, and Navid Azizan. Mirror descent maximizes generalized margin and can be implemented efficiently. In *NeurIPS*, 2022.

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023.

Guiyao Tie, Xueyang Zhou, Tianhe Gu, Ruihang Zhang, Chaoran Hu, Sizhe Zhang, Mengqu Sun, Yan Zhang, Pan Zhou, and Lichao Sun. Mmmr: Benchmarking massive multi-modal reasoning tasks. *arXiv preprint arXiv:2505.16459*, 2025.

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

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

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. In *NeurIPS*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Bohan Wang, Qi Meng, Wei Chen, and Tie-Yan Liu. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In *ICML*, 2021.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*, 2023a.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. Document-level machine translation with large language models. In *EMNLP*, 2023b.

Ruosong Wang, Peilin Zhong, Simon S Du, Russ R Salakhutdinov, and Lin Yang. Planning with general objective functions: Going beyond total rewards. In *NeurIPS*, 2020.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. In *EMNLP*, 2022.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. Cmath: Can your language model pass chinese elementary school math test? *arXiv preprint arXiv:2306.16636*, 2023.

Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *STOC*, 2012.

Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *SODA*, 2024.

David P Woodruff and Peilin Zhong. Distributed low rank approximation of implicit functions of a matrix. In *ICDE*, 2016.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in over-parametrized models. In *COLT*, 2020.

Weimin Wu, Teng-Yun Hsiao, Jerry Yao-Chieh Hu, Wenxin Zhang, and Han Liu. In-context learning as conditioned associative memory retrieval. In *ICML*, 2025a.

Weimin Wu, Maojiang Su, Jerry Yao-Chieh Hu, Zhao Song, and Han Liu. In-context deep learning via transformer models. In *ICML*, 2025b.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*, 2023.

Chang Xiao, Peilin Zhong, and Changxi Zheng. Bourgan: Generative networks with metric embeddings. In *NeurIPS*, 2018.

Shuo Xie, Jiahao Qiu, Ankita Pasad, Li Du, Qing Qu, and Hongyuan Mei. Hidden state variability of pretrained language models can guide computation reduction for transfer learning. In *EMNLP*, 2022.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *IEEE Transactions on Knowledge and Data Engineering*, 2025a.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*, 2025b.

Zhaozhuo Xu, Zhao Song, and Anshumali Shrivastava. A tale of two efficient value iteration algorithms for solving linear mdps with large action space. In *AISTATS*, 2023.

Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. Path attention: Position encoding via accumulating householder transformations. *arXiv preprint arXiv:2505.16381*, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. In *ICLR*, 2021.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. In *ICML*, 2023.

Eric Zelikman, Qian Huang, Percy Liang, Nick Haber, and Noah D Goodman. Just one byte (per gradient): A note on low-bandwidth decentralized language model finetuning using shared randomness. *arXiv preprint arXiv:2306.10015*, 2023.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? In *NeurIPS*, 2020a.

Lichen Zhang. Speeding up optimizations via data structures: Faster search, sample and maintenance. Master's thesis, Carnegie Mellon University, 2022.

Ruizhe Zhang and Xinzhi Zhang. A hyperbolic extension of kadison-singer type results. In *ICALP*, 2023.

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

Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. In *NeurIPS*, 2023.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. Sentiment analysis in the era of large language models: A reality check. In *NAACL*, 2024.

Yi Zhang, Orestis Plevrakis, Simon S Du, Xingguo Li, Zhao Song, and Sanjeev Arora. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. In *NeurIPS*, 2020b.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, and Min Lin. On evaluating adversarial robustness of large vision-language models. In *NeurIPS*, 2023.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In *NeurIPS*, 2019.

Difan Zou, Jingfeng Wu, Vladimir Braverman, Quanquan Gu, Dean P Foster, and Sham Kakade. The benefits of implicit regularization from sgd in least squares problems. In *NeurIPS*, 2021.

A Fast Optimization View: Reformulating Single Layer Attention in LLM Based on Tensor and SVM Trick, and Solving It in Matrix Multiplication Time (Supplementary Material)

Yeqi Gao[1] **Zhao Song**[2,*] **Weixin Wang**[3,†] **Junze Yin**[4,‡]

[1]University of Washington, [2]University of California, Berkeley
[3]Johns Hopkins University, [4]Boston University
[*]magic.linuxkde@gmail.com, [†]weixinw1@uci.edu,[‡]junze@bu.edu

Roadmap. In Section A, we present the basic notations we use, some mathematical facts, and helpful definitions that support the following proof. In Section B, we compute the gradients of the helpful functions defined earlier. In Section C, we define the Hessian for further discussion. In Section D, we compute the Hessian matrix with respect to X. In Section E, we demonstrate that the Hessian for X is Lipschitz. In Section F, we show that the Hessian matrix with respect to X is positive semidefinite (PSD). In Section G, we compute the Hessian matrix with respect to Y and show that it is Lipschitz and positive semidefinite (PSD). In Section H, we compute the Hessian matrix with respect to both X and Y. In Section I, we demonstrate that the Hessian matrix with respect to both X and Y is Lipschitz. In Section J, we introduce some tensor sketch techniques to obtain fast approximations of the Hessian. In Section K, we introduce the Newton step.

A PRELIMINARIES

In Section A.1, we present the basic mathematical properties of vectors, norms and matrices. In section A.2, we provide a definition of $L(X, Y)$. In Section A.3, we define a series of helpful functions with respect to X. In section A.4, we define a series of helpful functions with respect to Y. In Section A.5, we define a series of helpful functions with respect to both X and Y. In Section A.6, we define the regularization function. In Section A.7, we introduce facts related to fast matrix multiplication.

Notation Now we define the basic notations we use in this paper.

First, we define the notations related to the sets. We use $\mathbb{N}$ to denote the set of positive integers, namely $\mathbb{N} := \{1, 2, 3, \dots\}$. Let n and d be in $\mathbb{N}$. We define $[n] := \{1, 2, \dots, n\}$. We use $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n \times d}$ to denote the set containing all real numbers, all n-dimensional vectors, and $n \times d$ matrices, whose entries are all in $\mathbb{R}$. We use $\mathbb{R}_+$ to denote the set containing all positive real numbers.

Then, we define the notations related to vectors. Let $x, y \in \mathbb{R}^d$. For all $i \in [d]$, we define $x_i \in \mathbb{R}$ as the i-th entry of x. We define $\langle \cdot, \cdot \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as $\langle x, y \rangle := \sum_{i=1}^d x_i y_i$, which is called the inner product between x and y. We define $x \circ y \in \mathbb{R}^d$ as $(x \circ y)_i := x_i \cdot y_i$, for all $i \in [d]$. For all $p \in \{1, 2, \infty\}$, we define $\|x\|_p := (\sum_{i \in [d]} |x_i|^p)^{1/p}$, which is the ℓ_p norm of x. We use $\mathbf{1}_d$ and $\mathbf{0}_d$ to denote the d-dimensional vectors whose entries are all 1's and 0's, respectively.

After that, we define the notations related to matrices. Let $A \in \mathbb{R}^{n \times d}$. For all $i \in [n]$ and $j \in [d]$, we use $A_{i,j} \in \mathbb{R}$ to denote the entry of A at i-th row and j-th column, use $A_{i,*} \in \mathbb{R}^d$ and $A_{*,j} \in \mathbb{R}^n$ to denote vectors, where $(A_{i,*})_j = A_{i,j} = (A_{*,j})_i$. We use $A^\top \in \mathbb{R}^{d \times n}$ to denote the transpose of the matrix A, where $A^\top_{i,j} = A_{j,i}$. For $X \in \mathbb{R}^{d \times d}$, we define $x = \mathrm{vec}(X) \in \mathbb{R}^{d^2}$ as $X_{i,j} = \mathrm{vec}(X)_{(i-1) \times d + j}$. For $x \in \mathbb{R}^d$, we define $\mathrm{diag}(x) \in \mathbb{R}^{d \times d}$ as $\mathrm{diag}(x)_{i,i} = x_i$, for all $i \in [d]$ and other entries of $\mathrm{diag}(x)$ are all 0's. $\|A\|_F \in \mathbb{R}$ and $\|A\| \in \mathbb{R}$ denote the Frobenius norm and the spectral norm of $A \in \mathbb{R}^{n \times d}$, respectively, where $\|A\|_F := \sqrt{\sum_{i \in [n]} \sum_{j \in [d]} |A_{i,j}|^2}$ and $\|A\| := \max_{x \in \mathbb{R}^d} \|Ax\|_2 / \|x\|_2$. Let $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$. For each $j_1 \in [n]$, we use $\mathsf{A}_{j_1} \in \mathbb{R}^{n \times d^2}$ to denote one $n \times d^2$ block from $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$. Let $C, D \in \mathbb{R}^{d \times d}$ be symmetric matrices, $C \succeq D$ if for all $y \in \mathbb{R}^d$, $y^\top C y \geq y^\top D y$. C is said to be a positive semidefinite (PSD) matrix if $y^\top C y \geq 0$. We use I_d to denote the $d \times d$ identity matrix. $\mathrm{nnz}(A)$ represents the number of entries in the matrix A that are not equal to

zero. $\mathbf{0}_{n\times n} \in \mathbb{R}^{n\times n}$ is a matrix, where for all $i, j \in [n]$, $(\mathbf{0}_{n\times n})_{i,j} = 0$.

Let n_1, n_2, d_1, d_2 be positive integers. Let $A \in \mathbb{R}^{n_1\times d_1}$ and $B \in \mathbb{R}^{n_2\times d_2}$. We define the Kronecker product between matrices A and B, denoted $A \otimes B \in \mathbb{R}^{n_1 n_2 \times d_1 d_2}$, as $(A \otimes B)_{(i_1-1)n_2+i_2,(j_1-1)d_2+j_2}$ is equal to $A_{i_1,j_1}B_{i_2,j_2}$, where $i_1 \in [n_1], j_1 \in [d_1], i_2 \in [n_2], j_2 \in [d_2]$. $\mathrm{mat} : \mathbb{R}^{n^2} \to \mathbb{R}^{n\times n}$ is defined by $X_{i,j} = \mathrm{mat}(x)_{i,j} := x_{(i-1)\cdot n+j}$, and $\mathrm{vec} = \mathrm{mat}^{-1}$.

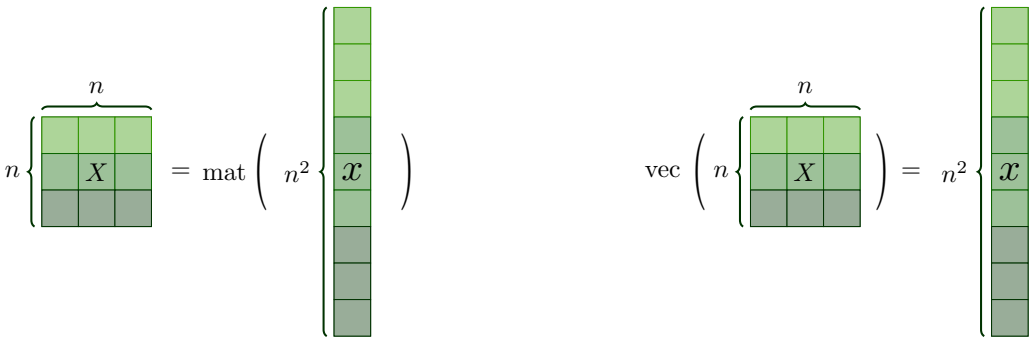

Figure 2: The visualization of the functions $\mathrm{mat} : \mathbb{R}^{n^2} \to \mathbb{R}^{n\times n}$ and $\mathrm{vec} = \mathrm{mat}^{-1} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n^2}$. We have $x \in \mathbb{R}^{n^2}$ and $X \in \mathbb{R}^{n\times n}$. In this figure, we give an example of $n = 3$. In the left figure, by the function mat, the first three entries of the vector x are mapped to $X_{1,1}$, $X_{1,2}$, and $X_{1,3}$ respectively, the second three entries of the vector x are mapped to $X_{2,1}$, $X_{2,2}$, and $X_{2,3}$ respectively, and the third three entries of the vector x are mapped to $X_{3,1}$, $X_{3,2}$, and $X_{3,3}$ respectively. For the right figure, every entry in X is mapped to x by vec in the reverse pattern of mat.

A.1 BASIC FACTS

In this section, we will introduce the basic mathematical facts.

Fact A.1. *Let $a, b \in \mathbb{R}$.*

For all vectors $u, v, w \in \mathbb{R}^n$, we have

- $\langle u, v\rangle = \langle u \circ v, \mathbf{1}_n\rangle = u^\top \mathrm{diag}(v)\mathbf{1}_n$
- $\langle u \circ v, w\rangle = \langle u \circ w, v\rangle$
- $\langle u \circ v, w\rangle = \langle u \circ v \circ w, \mathbf{1}_n\rangle = u^\top \mathrm{diag}(v)w$
- $\langle u \circ v \circ w \circ z, \mathbf{1}_n\rangle = u^\top \mathrm{diag}(v \circ w)z$
- $u \circ v = v \circ u = \mathrm{diag}(u) \cdot v = \mathrm{diag}(v) \cdot u$
- $u^\top(v \circ w) = v^\top(u \circ w) = w^\top(u \circ v) = u^\top \mathrm{diag}(v)w = v^\top \mathrm{diag}(u)w = w^\top \mathrm{diag}(u)v$
- $\mathrm{diag}(u)^\top = \mathrm{diag}(u)$
- $\mathrm{diag}(u) \cdot \mathrm{diag}(v) \cdot \mathbf{1}_n = \mathrm{diag}(u)v$
- $\mathrm{diag}(u \circ v) = \mathrm{diag}(u)\,\mathrm{diag}(v)$
- $\mathrm{diag}(u) + \mathrm{diag}(v) = \mathrm{diag}(u + v)$
- $\langle u, v\rangle = \langle v, u\rangle$
- $\langle u, v\rangle = u^\top v = v^\top u$
- $a\langle w, v\rangle + b\langle u, v\rangle = \langle aw + bu, v\rangle = \langle v, aw + bu\rangle = a\langle v, w\rangle + b\langle v, u\rangle$.

Fact A.2. *Let $R > 0$ be a real number.*

For vectors $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, we have

- $\|x \circ y\|_2 \leq \|x\|_\infty \cdot \|y\|_2$
- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$

- $\|\exp(x)\|_\infty \leq \exp(\|x\|_2)$
- $\|x+y\|_2 \leq \|x\|_2 + \|y\|_2$
- $\|\alpha x\|_2 \leq |\alpha| \cdot \|x\|_2$
- *For any* $\|x\|_2, \|y\|_2 \leq R$*, we have* $\|\exp(x) - \exp(y)\|_2 \leq \exp(R) \cdot \|x-y\|_2$

Fact A.3. *For any matrices* $X, Y \in \mathbb{R}^{n \times n}$ *and for any vector* $x \in \mathbb{R}^n$*, we have*

- $\|X^\top\| = \|X\|$
- $\|X\| \geq \|Y\| - \|X - Y\|$
- $\|X + Y\| \leq \|X\| + \|Y\|$
- $\|X \cdot Y\| \leq \|X\| \cdot \|Y\|$
- *If* $X \preceq \alpha \cdot Y$*, then* $\|X\| \leq \alpha \cdot \|Y\|$*, for* X *and* Y *being PSD matrices and* $\alpha > 0$*.*
- $\|Yx\|_2 \leq \|Y\| \cdot \|x\|_2$

Fact A.4. *For any vectors* $u, v \in \mathbb{R}^n$*, we have*

- *Part 1.* $uu^\top \preceq \|u\|_2^2 \cdot I_n$
- *Part 2.* $\mathrm{diag}(u) \preceq \|u\|_2 \cdot I_n$
- *Part 3.* $\mathrm{diag}(u \circ u) \preceq \|u\|_2^2 \cdot I_n$
- *Part 4.* $uv^\top + vu^\top \preceq uu^\top + vv^\top$
- *Part 5.* $uv^\top + vu^\top \succeq -(uu^\top + vv^\top)$
- *Part 6.* $(v \circ u)(v \circ u)^\top \preceq \|v\|_\infty^2 uu^\top$
- *Part 7.* $\mathrm{diag}(u \circ v) \preceq \|u\|_2 \|v\|_2 \cdot I_n$

Fact A.5. *Let* $g, f : \mathbb{R}^d \to \mathbb{R}^n$ *and* $q : \mathbb{R}^d \to \mathbb{R}$*.*

Let $x \in \mathbb{R}^d$ *be an arbitrary vector.*

Let $a \in \mathbb{R}$ *be an arbitrary real number.*

Then, we have

- $\frac{\mathrm{d}q(x)^a}{\mathrm{d}x} = a \cdot q(x)^{a-1} \cdot \frac{\mathrm{d}q(x)}{\mathrm{d}x}$
- $\frac{\mathrm{d}\|f(x)\|_2^2}{\mathrm{d}t} = 2\langle f(x), \frac{\mathrm{d}f(x)}{\mathrm{d}t} \rangle$
- $\frac{\mathrm{d}\langle f(x), g(x) \rangle}{\mathrm{d}t} = \langle \frac{\mathrm{d}f(x)}{\mathrm{d}t}, g(x) \rangle + \langle f(x), \frac{\mathrm{d}g(x)}{\mathrm{d}t} \rangle$
- $\frac{\mathrm{d}(g(x) \circ f(x))}{\mathrm{d}t} = \frac{\mathrm{d}g(x)}{\mathrm{d}t} \circ f(x) + g(x) \circ \frac{\mathrm{d}f(x)}{\mathrm{d}t}$ *(product rule for Hadamard product)*

A.2 GENERAL DEFINITIONS

In this section, we introduce some general definitions.

Definition A.6 (Index summary)**.** *We use* i *to denote indices in* $[d^2]$ *range, and* j *to denote indices in* $[n^2]$ *range.*

We use i_0, i_1, i_2 *to denote indices in* $[d]$*, and* j_0, j_1, j_2 *to denote indices in* $[n]$*.*

Definition A.7. *If the following conditions hold*

- *Let* $A_1 \in \mathbb{R}^{n \times d}$*.*
- *Let* $A_2 \in \mathbb{R}^{n \times d}$*.*
- *Let* $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$ *denote the Kronecker product between* A_1, A_2
  - *For each* $j_0 \in [n]$*, we use* $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ *to be one* $n \times d^2$ *block from* $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$ *(see Remark 3.1).*
- *Let* $A_3 \in \mathbb{R}^{n \times d}$*.*

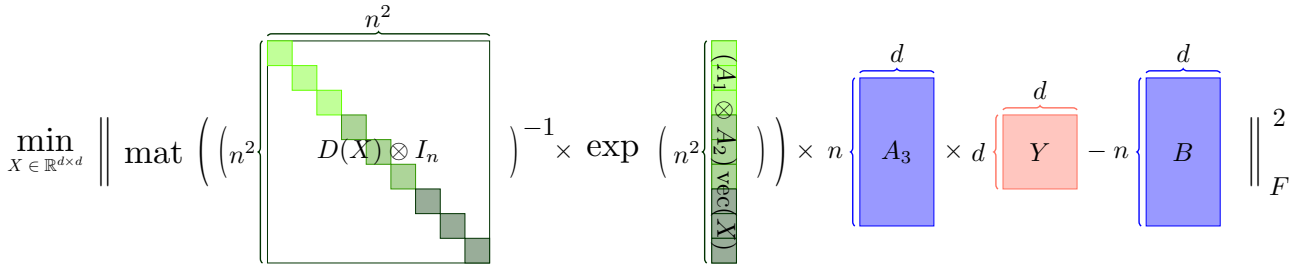

Figure 3: The visualization of a variation of Definition 1.2. Let $A_1, A_2, A_3, B \in \mathbb{R}^{n \times d}$, $X \in \mathbb{R}^{d \times d}$, $D(X) \in \mathbb{R}^{n \times n}$ (see Figure 1 and Definition 1.2), and $\mathsf{A} = A_1 \otimes A_2 \in \mathbb{R}^{n^2 \times d^2}$. $\mathrm{mat} : \mathbb{R}^{n^2} \to \mathbb{R}^{n \times n}$ is defined by $X_{i,j} = \mathrm{mat}(x)_{i,j} := x_{(i-1)\cdot n + j}$, and $\mathrm{vec} = \mathrm{mat}^{-1}$. We first get that $(D(X) \otimes I_n)^{-1} \in \mathbb{R}^{n^2 \times n^2}$ and multiply A with $\mathrm{vec}(X)$. Then, we multiply $(D(X) \otimes I_n)^{-1} \in \mathbb{R}^{n^2 \times n^2}$ with $\mathsf{A} \cdot \mathrm{vec}(X) \in \mathbb{R}^{n^2}$, which gives us a vector in $\mathbb{R}^{n^2}$. We use mat to transform that into a matrix in $\mathbb{R}^{n \times n}$. After that, we multiply this matrix with $A_3 Y \in \mathbb{R}^{n \times d}$. Finally, we compute the minimum of the Frobenius norm of $\mathrm{mat}((D(X) \otimes I_n)^{-1} \cdot \exp(\mathsf{A}\,\mathrm{vec}(X)))A_3 Y - B$. In this figure, we give an example when $n = 3$: in the matrix $D(X) \otimes I_n$, the three light green squares (and their nearby white area) make up the first chunk, the three middle green squares (and their nearby white area) make up the second chunk, and the three dark green squares (and their nearby white area) make up the third chunk. The blue rectangles represent the matrices in $\mathbb{R}^{n \times d}$. The red rectangle represents the matrix in $\mathbb{R}^{d \times d}$.

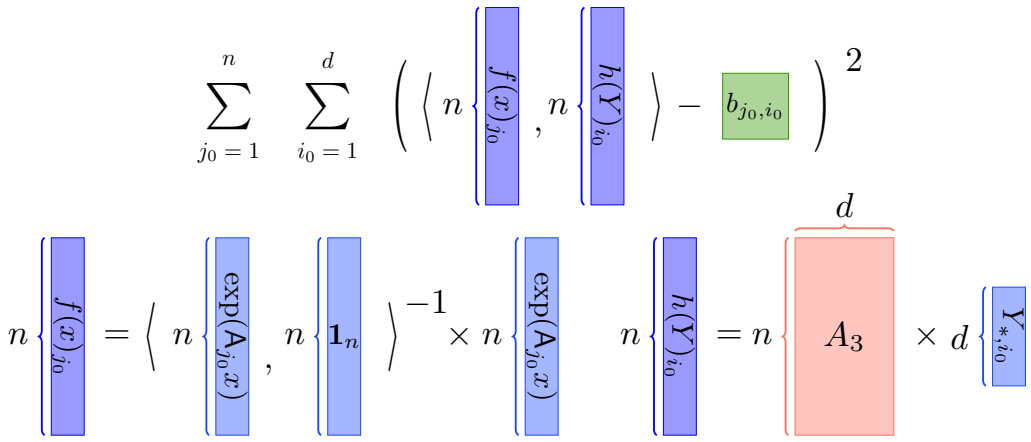

Figure 4: The visualization of Eq. (2). Let $A_1, A_2, A_3, B \in \mathbb{R}^{n \times d}$ and $X, Y \in \mathbb{R}^{d \times d}$. We have $\mathsf{A} = A_1 \otimes A_2 \in \mathbb{R}^{n^2 \times d^2}$ and $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ is the j_0-th block of A. $x = \mathrm{vec}(X) \in \mathbb{R}^{d^2}$. First, we use the definition of $f(x)_{j_0} \in \mathbb{R}^n$ (see Definition A.10) and $h(Y)_{i_0} \in \mathbb{R}^n$ (see Definition A.11) to compute them. Then, we find their inner produce and subtract the entry of B at j_0-th row and i_0-column from the inner produce. Finally, we compute the square of this difference and add all of them from $i_0 = 1$ to $i_0 = d$ and from $j_0 = 1$ to $j_0 = n$. In this figure, we use blue rectangles to represent vectors, where the dark blue represents $f(x)_{j_0}$ and $h(Y)_{i_0}$, and the light blue represents the terms used to compute $f(x)_{j_0}$ and $h(Y)_{i_0}$. The green square represents the scalar. The red rectangle represents the matrix.

- *Let $B \in \mathbb{R}^{n \times d}$ and b_{j_0,i_0} denote the (j_0, i_0)-th entry in $B \in \mathbb{R}^{n \times d}$ for each $j_0 \in [n]$ and $i_0 \in [d]$.*
- *Let $X \in \mathbb{R}^{d \times d}$.*

Our final goal is to study the loss function, defined as:

$$L(X,Y) := 0.5 \cdot \| \underbrace{D(X)^{-1}}_{n \times n} \underbrace{\exp(A_1 X A_2^\top)}_{n \times n} \underbrace{A_3}_{n \times d} \underbrace{Y}_{d \times d} - \underbrace{B}_{n \times d} \|_F^2$$

where

- *$D(X) \in \mathbb{R}^{n \times n}$ is defined as $D(X) := \mathrm{diag}(\exp(A_1 X A_2^\top)\mathbf{1}_n)$ and*
- *for each $j_0 \in [n]$, $D(X)_{j_0} \in \mathbb{R}$ is $\langle \exp(\mathsf{A}_{j_0}\, x), \mathbf{1}_n \rangle$, $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ is the j_0-th block of $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$, and $x \in \mathbb{R}^{d^2}$ is the vectorization of $X \in \mathbb{R}^{d \times d}$*

Further, for each $j_0 \in [n], i_0 \in [d]$, we define $L(X,Y)_{j_0,i_0}$ as follows:

$$L(X,Y)_{j_0,i_0} := 0.5(\langle \langle \exp(\mathsf{A}_{j_0} x), \mathbf{1}_n \rangle^{-1} \exp(\mathsf{A}_{j_0} x), A_3 Y_{*,i_0} \rangle - b_{j_0,i_0})^2$$

Using tensor-trick in Gao et al. [2023b,c], we can see that

$$L(X,Y) = \sum_{j_0=1}^{n} \sum_{i_0=1}^{d} L(X,Y)_{j_0,i_0}.$$

A.3 HELPFUL DEFINITIONS WITH RESPECT TO X

Now, we introduce a few helpful definitions related to $X \in \mathbb{R}^{d \times d}$.

Definition A.8. *Let $\mathsf{A} = A_1 \otimes A_2 \in \mathbb{R}^{n^2 \times d^2}$, where $A_1, A_2 \in \mathbb{R}^{n \times d}$, and $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ be one $n \times d^2$ block from A.*

We define $u(x)_{j_0} : \mathbb{R}^{d^2} \to \mathbb{R}^n$ as follows:

$$u(x)_{j_0} := \underbrace{\exp(\mathsf{A}_{j_0} x)}_{n \times 1}.$$

Definition A.9. *Let $\mathsf{A} = A_1 \otimes A_2 \in \mathbb{R}^{n^2 \times d^2}$, where $A_1, A_2 \in \mathbb{R}^{n \times d}$, and $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d^2}$ be one $n \times d^2$ block from A.*

We define $\alpha(x)_{j_0} : \mathbb{R}^{d^2} \to \mathbb{R}$ as:

$$\alpha(x)_{j_0} := \langle \underbrace{\exp(\mathsf{A}_{j_0} x)}_{n \times 1}, \underbrace{\mathbf{1}_n}_{n \times 1} \rangle.$$

Definition A.10. *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as in Definition A.9.*

Let $u(x)_{j_0} \in \mathbb{R}^n$ be defined as in Definition A.8.

We define $f(x)_{j_0} : \mathbb{R}^{d^2} \to \mathbb{R}^n$

$$f(x)_{j_0} := \underbrace{\alpha(x)_{j_0}^{-1}}_{\text{scalar}} \underbrace{u(x)_{j_0}}_{n \times 1}.$$

A.4 A HELPFUL DEFINITION WITH RESPECT TO Y

In this section, we introduce a helpful definition related to $Y \in \mathbb{R}^{d \times d}$.

Definition A.11. *For each $i_0 \in [d]$, we define $h()_{i_0} : \mathbb{R}^{d \times d} \to \mathbb{R}^n$ as:*

$$h(Y)_{i_0} := \underbrace{A_3}_{n \times d} \underbrace{Y_{*,i_0}}_{d \times 1}.$$

A.5 HELPFUL DEFINITIONS WITH RESPECT TO BOTH X AND Y

In this section, we introduce some helpful definitions related to both $X \in \mathbb{R}^{d \times d}$ and $Y \in \mathbb{R}^{d \times d}$.

Definition A.12. *We define $c(x,y)_{j_0,i_0} : \mathbb{R}^{d^2} \times \mathbb{R}^{d^2} \to \mathbb{R}$ as follows:*

$$c(x,y)_{j_0,i_0} := \langle f(x)_{j_0}, h(y)_{i_0} \rangle - b_{j_0,i_0}.$$

Furthermore, we define $c(x,:)_{j_0,i_0}$ as follows

$$c(x,:)_{j_0,i_0} := \langle f(x)_{j_0}, v \rangle - b_{j_0,i_0}$$

for some fixed vector $v \in \mathbb{R}^n$ which doesn't depend on x and also doesn't depend on y.

Similarly, we also define $c(:,y)_{j_0,i_0}$ as follows

$$c(:,y)_{j_0,i_0} := \langle v, h(y)_{i_0} \rangle - b_{j_0,i_0}$$

for some fixed vector $v \in \mathbb{R}^n$ which doesn't depend on x and also doesn't depend on y.

Definition A.13. *We define*

$$L(x,:)_{j_0,i_0} := 0.5c(x,:)^2_{j_0,i_0}$$

and

$$L(:,y)_{j_0,i_0} := 0.5c(:,y)^2_{j_0,i_0}$$

and

$$L(x,y)_{j_0,i_0} := 0.5c(x,y)^2_{j_0,i_0}$$

A.6 REGULARIZATION

In this section, we define the regularization loss we use.

Definition A.14. *Let $W \in \mathbb{R}^{n \times n}$ denote a positive diagonal matrix. We use the following regularization loss*

$$\|(W \otimes I)(A_1 \otimes A_2)x\|_2^2 + \|WA_3y\|_F^2$$

Note that $\|WA_3y\|_F^2 = \sum_{i_0=1}^d \|WA_3y_{i_0}\|_2^2$.

Adding this regularization term to the loss function $L(X,Y)$ (see Definition A.7), we can ensure the positive definiteness of this loss function (see Lemma G.1 and Lemma F.1).

A.7 FAST MATRIX MULTIPLICATION

We use $\mathcal{T}_{\mathrm{mat}}(a,b,c)$ to denote the time of multiplying an $a \times b$ matrix with another $b \times c$ matrix. Fast matrix multiplication Coppersmith [1982], Williams [2012], Le Gall [2014], Gall and Urrutia [2018], Christandl et al. [2025], Alman and Williams [2021], Duan et al. [2023], Le Gall [2024], Williams et al. [2024] is a fundamental tool in theoretical computer science.

Fact A.15. $O(\mathcal{T}_{\mathrm{mat}}(a,b,c)) = O(\mathcal{T}_{\mathrm{mat}}(b,a,c)) = O(\mathcal{T}_{\mathrm{mat}}(a,c,b))$.

For $k \in \mathbb{R}_+$, we define $\omega(k) \in \mathbb{R}_+$ to be the value such that $\forall n \in \mathbb{N}$, $\mathcal{T}_{\mathrm{mat}}(n,n,n^k) = O(n^{\omega(k)})$.

For convenience, we define three special values of $\omega(k)$. We define ω to be the fast matrix multiplication exponent, i.e., $\omega := \omega(1)$. We define $\alpha \in \mathbb{R}_+$ to be the dual exponent of matrix multiplication, i.e., $\omega(\alpha) = 2$. We define $\beta := \omega(2)$.

The following fact can be found in Lemma 3.6 of Jiang et al. [2020a], also see Bürgisser et al. [1997].

Fact A.16 (Convexity of $\omega(k)$)**.** *The function $\omega(k)$ is convex.*

B GRADIENT

In Section B.1, we show the gradient with respect to variables x. In Section B.2, we prove the gradient with respect to variables y. In Section B.3, we compute running time of c, f, h. In Section B.4, we reformulate the gradient with respect to X to compute time complexity. In Section B.5, we reformulate the gradient with respect to Y to compute time complexity.

B.1 GRADIENT FOR x

In this section, we compute the gradient for x. Most of the following gradient computations can be found in Gao et al. [2023b,c].

Lemma B.1 (Gradient with respect to x)**.** *If the following conditions hold*

- *For each $i \in [d^2]$, let $\mathsf{A}_{j_0,i} \in \mathbb{R}^n$ denote the i-th column for $\mathsf{A}_{j_0} \in \mathbb{R}^{n \times d}$*
- *Let $u(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.8*

- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $L(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.13*

Then, for each $i \in [d^2]$*, for each* $j_0 \in [n]$*, we have*

- **Part 1.**

$$\frac{\mathrm{d}u(x)_{j_0}}{\mathrm{d}x_i} = u(x)_{j_0} \circ \mathsf{A}_{j_0,i}$$

- **Part 2.**

$$\frac{\mathrm{d}\alpha(x)_{j_0}}{\mathrm{d}x_i} = \langle u(x)_{j_0} \circ \mathsf{A}_{j_0,i}, \mathbf{1}_n \rangle$$

- **Part 3.**

$$\frac{\mathrm{d}f(x)_{j_0}}{\mathrm{d}x_i} = f(x)_{j_0} \circ \mathsf{A}_{j_0,i} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle$$

- **Part 4.** *For a fixed vector* $v \in \mathbb{R}^n$ *(which doesn't depend on* x*), we have*

$$\frac{\mathrm{d}\langle f(x)_{j_0}, v \rangle}{\mathrm{d}x_i} = \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle - \langle f(x)_{j_0}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle$$

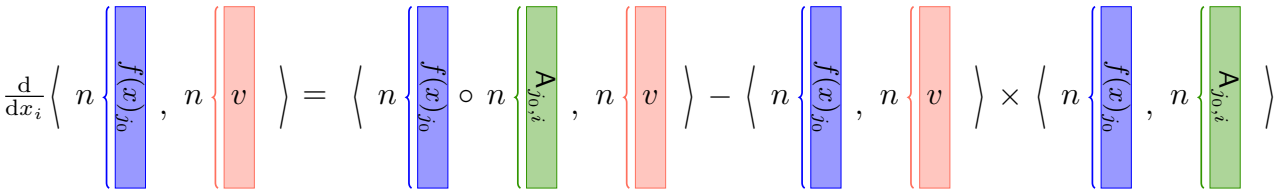

Figure 5: The visualization of Part 4 of Lemma B.1. We are given $f(x)_{j_0}, v, \mathsf{A}_{j_0,i} \in \mathbb{R}^n$. The left-hand side of the equation is the derivative of the inner product of $f(x)_{j_0}$ and v with respect to $x_i \in \mathbb{R}$. For the right-hand side, we have three steps. Step 1: we compute the Hadamard product of $f(x)_{j_0}$ and $\mathsf{A}_{j_0,i}$. Step 2: We find the inner product of this Hadamard product and v. Step 3: We subtract the product of two inner products, one is of $f(x)_{j_0}$ and v and the other is of $f(x)_{j_0}$ and $\mathsf{A}_{j_0,i}$, from the result of step 2. The purple rectangles represent the vector $f(x)_{j_0}$. The red rectangles represent the vector v. The green rectangles represent the vector $\mathsf{A}_{j_0,i}$.

- **Part 5.** *For each* $i_0 \in [d]$

$$\frac{\mathrm{d}c(x,:)_{j_0,i_0}}{\mathrm{d}x_i} = \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle - \langle f(x)_{j_0}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle$$

- **Part 6.**

$$\frac{\mathrm{d}L(x,:)_{j_0,i_0}}{\mathrm{d}x_i} = c(x,:)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle - \langle f(x)_{j_0}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle)$$

- **Part 7.** *(for hessian diagonal term)*

$$\frac{\mathrm{d}\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle}{\mathrm{d}x_i} = \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v \rangle - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle$$

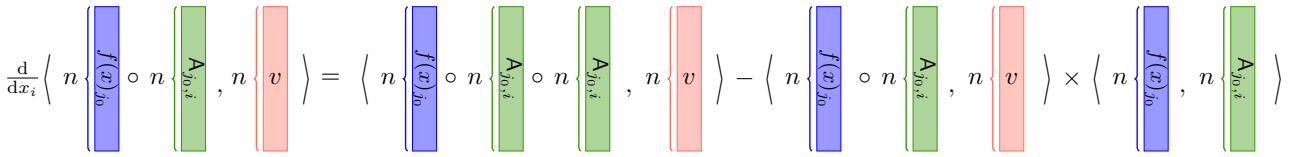

Figure 6: The visualization of Part 7 of Lemma B.1. We are given $f(x)_{j_0}, v, \mathsf{A}_{j_0,i} \in \mathbb{R}^n$. First, we compute the Hadamard product between $f(x)_{j_0}$ and $\mathsf{A}_{j_0,i}$. The left-hand side of the equation is the derivative of the inner product of this Hadamard product and v with respect to $x_i \in \mathbb{R}$. For the right-hand side, we have four steps. Step 1: We compute the inner product of the Hadamard product of $f(x)_{j_0}, \mathsf{A}_{j_0,i}, \mathsf{A}_{j_0,i}$ and v. Step 2: We compute the inner product of the Hadamard product of $f(x)_{j_0}, \mathsf{A}_{j_0,i}$ and v. Step 3: We compute the inner product between $f(x)_{j_0}$ and $\mathsf{A}_{j_0,i}$. Step 4: We subtract the product of steps 2 and 3 from step 1. The purple rectangles represent the vector $f(x)_{j_0}$. The red rectangles represent the vector v. The green rectangles represent the vector $\mathsf{A}_{j_0,i}$.

- **Part 8.** *(for hessian off-diagonal term)*

$$\frac{\mathrm{d}\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle}{\mathrm{d}x_l} = \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle$$

- **Part 9** *(for hessian diagonal term, this can be obtained by using Part 4 as a black-box)*

$$\frac{\mathrm{d}\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle}{\mathrm{d}x_i} = \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}\rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle$$

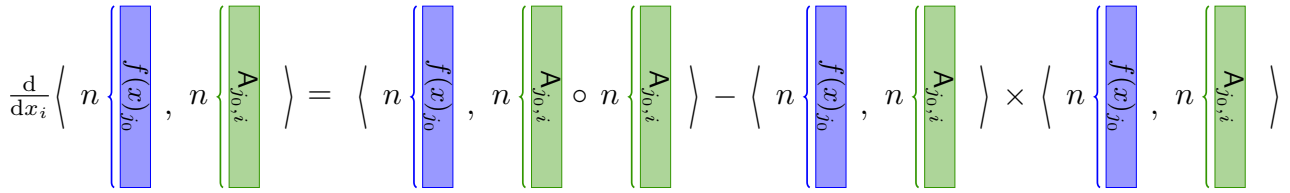

Figure 7: The visualization of Part 9 of Lemma B.1. We are given $f(x)_{j_0}, \mathsf{A}_{j_0,i} \in \mathbb{R}^n$. The left-hand side of the equation is the derivative of the inner product of $f(x)_{j_0}$ and $\mathsf{A}_{j_0,i}$ with respect to $x_i \in \mathbb{R}$. For the right-hand side, we have three steps. Step 1: we compute the Hadamard product of $\mathsf{A}_{j_0,i}$ and $\mathsf{A}_{j_0,i}$. Step 2: We find the inner product of $f(x)_{j_0}$ and this Hadamard product. Step 3: We subtract the square of inner product of $f(x)_{j_0}$ and $\mathsf{A}_{j_0,i}$ from the result of step 2. The purple rectangles represent the vector $f(x)_{j_0}$. The green rectangles represent the vector $\mathsf{A}_{j_0,i}$.

- **Part 10** *(for hessian off-diagonal term, this can be obtained by using Part 4 as a black-box)*

$$\frac{\mathrm{d}\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle}{\mathrm{d}x_l} = \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}\rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle$$

Proof. **Proof of Part 1.** See Part 4 of Proof of Lemma 5.18 in Gao et al. [2023b] (Page 14).

Proof of Part 2. See Part 5 of Proof of Lemma 5.18 in Gao et al. [2023b] (Page 14).

Proof of Part 3. See Part 9 of Proof of Lemma 5.18 in Gao et al. [2023b] (page 15).

Proof of Part 4. See Part 14 of Proof of Lemma 5.18 in Gao et al. [2023b] (page 15).

Proof of Part 5.

Note that by Definition A.12, we have

$$c(x,:)_{j_0,i_0} := \langle f(x)_{j_0}, v\rangle - b_{j_0,i_0} \tag{3}$$

Therefore, we have

$$\frac{\mathrm{d}c(x,:)_{j_0,i_0}}{\mathrm{d}x_i} = \frac{\mathrm{d}(\langle f(x)_{j_0}, v\rangle - b_{j_0,i_0})}{\mathrm{d}x_i}$$

$$
\begin{aligned}
&= \frac{\mathrm{d}\langle f(x)_{j_0}, v\rangle}{\mathrm{d}x_i} \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle,
\end{aligned}
$$

where the first step comes from Eq. (3), the second step follows from $\frac{\mathrm{d}b_{j_0,i_0}}{\mathrm{d}x_i} = 0$, and the third step is due to **Part 4**.

Proof of Part 6. Noted that by Definition A.13, we have

$$
L(x,:)_{j_0,i_0} = 0.5c(x,:)^2_{j_0,i_0} \tag{4}
$$

Therefore, we have

$$
\begin{aligned}
\frac{\mathrm{d}L(x,:)_{j_0,i_0}}{\mathrm{d}x_i} &= \frac{\mathrm{d}(0.5c(x,:)^2_{j_0,i_0})}{\mathrm{d}x_i} \\
&= c(x,:)_{j_0,i_0} \frac{\mathrm{d}c(x,:)}{\mathrm{d}x_i} \\
&= c(x,:)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle),
\end{aligned}
$$

where the first step is due to Eq. (4), the second step is because of chain rule of derivative, the last step comes from **Part 5**.

Proof of Part 7.

We have

$$
\begin{aligned}
\frac{\mathrm{d}\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle}{\mathrm{d}x_i} &= \langle \frac{\mathrm{d}(f(x)_{j_0} \circ \mathsf{A}_{j_0,i})}{\mathrm{d}x_i}, v\rangle \\
&= \langle \frac{\mathrm{d}f(x)_{j_0}}{\mathrm{d}x_i} \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle (f(x)_{j_0} \circ \mathsf{A}_{j_0,i} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle
\end{aligned}
$$

where the first step is due to Fact A.5, the second step comes from Fact A.5, the third step is because of **Part 4**, the fourth step is owing to simple algebra, the fifth step follows from Fact A.1, and the last step comes from Fact A.1.

Proof of Part 8.

We have

$$
\begin{aligned}
\frac{\mathrm{d}\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle}{\mathrm{d}x_l} &= \langle \frac{\mathrm{d}(f(x)_{j_0} \circ \mathsf{A}_{j_0,i})}{\mathrm{d}x_l}, v\rangle \\
&= \langle \frac{\mathrm{d}f(x)_{j_0}}{\mathrm{d}x_l} \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle (f(x)_{j_0} \circ \mathsf{A}_{j_0,l} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle) \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle - \langle f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle \circ \mathsf{A}_{j_0,i}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle
\end{aligned}
$$

where the first step comes from Fact A.5, the second step is because of Fact A.5, the third step follows from **Part 4**, the fourth step is due to simple algebra, the fifth step is owing to Fact A.1, and the last step comes from Fact A.1.

Proof of Part 9.

We have

$$
\frac{\mathrm{d}\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle}{\mathrm{d}x_i} = \langle \frac{\mathrm{d}f(x)_{j_0}}{\mathrm{d}x_i}, \mathsf{A}_{j_0,i}\rangle
$$

$$
\begin{aligned}
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle, \mathsf{A}_{j_0,i} \rangle \\
&= \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i} \rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle
\end{aligned}
$$

where the first step is due to Fact A.5, the second step comes from **Part 4**, and the last step is because of Fact A.1.

Proof of Part 10. We have

$$
\begin{aligned}
\frac{\mathrm{d}\langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle}{\mathrm{d}x_l} &= \langle \frac{\mathrm{d}f(x)_{j_0}}{\mathrm{d}x_l}, \mathsf{A}_{j_0,i} \rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l} - f(x)_{j_0} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l} \rangle, \mathsf{A}_{j_0,i} \rangle \\
&= \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l} \rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l} \rangle
\end{aligned}
$$

where the first step comes from Fact A.5, the second step is owing to **Part 4**, and the last step is due to Fact A.1. □

B.2 GRADIENT WITH RESPECT TO y

In this section, we compute the gradient with respect to y.

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

- *Let $v \in \mathbb{R}^n$ which doesn't depend on x and also doesn't depend on y.*
- *Let $c(:, y)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12.*
- *Let $L(:, y)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.13.*
- *Let $h(y_{i_0}) := \underbrace{A_3}_{n \times d} \underbrace{y_{i_0}}_{d \times 1}$.*
- *Let $h(y_{i_0}) = h(y)_{i_0}$ for convenient*
- *Let $A_{3,*,i_2} \in \mathbb{R}^n$ denote the i_2-th column of matrix $A_3 \in \mathbb{R}^{n \times d}$ for each $i_2 \in [d]$*

Then, we have

- **Part 1.** *If $i_1 = i_0$*

$$
\frac{\mathrm{d}h(y_{i_0})}{\mathrm{d}y_{i_1,i_2}} = A_{3,*,i_2}
$$

- **Part 2.** *If $i_1 \neq i_0$*

$$
\frac{\mathrm{d}h(y_{i_0})}{\mathrm{d}y_{i_1,i_2}} = \mathbf{0}_n
$$

- **Part 3.** *If $i_1 = i_0$*

$$
\frac{\mathrm{d}\langle v, h(y)_{i_0} \rangle}{\mathrm{d}y_{i_1,i_2}} = \langle v, A_{3,*,i_2} \rangle
$$

- **Part 4.** *If $i_1 \neq i_0$*

$$
\frac{\mathrm{d}\langle v, h(y)_{i_0} \rangle}{\mathrm{d}y_{i_1,i_2}} = 0
$$

- **Part 5.** *If $i_1 = i_0$*

$$
\frac{\mathrm{d}c(:, y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = \langle v, A_{3,*,i_2} \rangle
$$

- **Part 6.** *If $i_1 \neq i_0$*

$$
\frac{\mathrm{d}c(:, y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = 0
$$

- **Part 7.** *If* $i_1 = i_0$

$$\frac{\mathrm{d}L(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = c(:,y)_{j_0,i_0}\langle v, A_{3,*,i_2}\rangle$$

- **Part 8.** *If* $i_1 \neq i_0$

$$\frac{\mathrm{d}L(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = 0$$

Proof. **Proof of Part 1.**

$$\begin{aligned}\frac{\mathrm{d}h(y_{i_0})}{\mathrm{d}y_{i_1,i_2}} &= \frac{\mathrm{d}A_3 y_{i_0}}{\mathrm{d}y_{i_1,i_2}} \\ &= A_{3,*,i_2}\end{aligned}$$

where the first step is due to the definition of $h(y_{i_0})$ (see the Lemma statement), and the last step comes from that for $i \neq i_2,\ \frac{\mathrm{d}}{\mathrm{d}y_{i_2}} f(y_i) = 0$.

Proof of Part 2.

$$\frac{\mathrm{d}h(y_{i_0})}{\mathrm{d}y_{i_1,i_2}} = \mathbf{0}_n$$

where the first step is due to $i_1 \neq i_2$.

Proof of Part 3.

$$\begin{aligned}\frac{\mathrm{d}\langle v, h(y)_{i_0}\rangle}{\mathrm{d}y_{i_1,i_2}} &= \langle v, \frac{\mathrm{d}h(y_{i_0})}{\mathrm{d}y_{i_1,i_2}}\rangle \\ &= \langle v, A_{3,*,i_2}\rangle\end{aligned}$$

where the first step comes from Fact A.5, the second step is due to the result of **Part 1**.

Proof of Part 4.

$$\begin{aligned}\frac{\mathrm{d}\langle v, h(y)_{i_0}\rangle}{\mathrm{d}y_{i_1,i_2}} &= \langle v, \frac{\mathrm{d}h(y_{i_0})}{\mathrm{d}y_{i_1,i_2}}\rangle \\ &= 0\end{aligned}$$

where the first step is becaues of Fact A.5, the second step comes from the result of **Part 2**.

Proof of Part 5.

$$\begin{aligned}\frac{\mathrm{d}c(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} &= \frac{\mathrm{d}\langle v, h(y)_{i_0}\rangle - b_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} \\ &= \frac{\mathrm{d}\langle v, h(y)_{i_0}\rangle}{\mathrm{d}y_{i_1,i_2}} \\ &= \langle v, A_{3,*,i_2}\rangle\end{aligned}$$

where the first step comes from the Definition A.12, the second step is because of $\frac{\mathrm{d}b_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = 0$, and the last step is due to **Part 3**.

Proof of Part 6.

$$\begin{aligned}\frac{\mathrm{d}c(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} &= \frac{\mathrm{d}\langle v, h(y)_{i_0}\rangle - b_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} \\ &= \frac{\mathrm{d}\langle v, h(y)_{i_0}\rangle}{\mathrm{d}y_{i_1,i_2}}\end{aligned}$$

$$= 0$$

where the first step is due to the Definition A.12, the second step comes from $\frac{\mathrm{d}b_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = 0$, and the last step is owing to **Part 4**.

Proof of Part 7.

$$\begin{aligned}\frac{\mathrm{d}L(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} &= \frac{\mathrm{d}0.5c(:,y)_{j_0,i_0}^2}{\mathrm{d}y_{i_1,i_2}} \\ &= c(:,y)_{j_0,i_0} \cdot \frac{\mathrm{d}c(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} \\ &= c(:,y)_{j_0,i_0} \langle v, A_{3,*,i_2} \rangle\end{aligned}$$

where the first step is due to the Definition A.13, the second step comes from the chain rule of derivative, and the last step is owing to **Part 5**.

Proof of Part 8.

$$\begin{aligned}\frac{\mathrm{d}L(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} &= \frac{\mathrm{d}0.5c(:,y)_{j_0,i_0}^2}{\mathrm{d}y_{i_1,i_2}} \\ &= c(:,y)_{j_0,i_0} \cdot \frac{\mathrm{d}c(:,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} \\ &= 0\end{aligned}$$

where the first step is because of the Definition A.13, the second step is due to the chain rule of derivative, and the last step comes from **Part 6**. □

B.3 COMPUTATION OF c, f, h

In this section, we explain how to compute $c(x,y), f(x), h(y)$.

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

- *For each $j_0 \in [n]$, $i_0 \in [d]$, let $c(x,y)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12. (We can view $c(x,y)$ as an $n \times d$ matrix)*
- *For each $j_0 \in [n]$, let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10. (We can view $f(x)$ as an $n \times n$ matrix)*
- *For each $i_0 \in [d]$, let $h(y)_{i_0} \in \mathbb{R}^n$ be defined as Definition A.11. (We can view $h(y)$ as $n \times d$ matrix)*
- *Let $A_3 \in \mathbb{R}^{n \times d}$*
- *We can view y as an $d \times d$ matrix*

Then, we can compute f, h, c in $O(\mathcal{T}_{\mathrm{mat}}(n,d,d) + \mathcal{T}_{\mathrm{mat}}(n,n,d))$ time.

Proof. By definition A.11, we have

$$\underbrace{h(y)}_{n \times d} = \underbrace{A_3}_{n \times d} \underbrace{y}_{d \times d}. \tag{5}$$

First $h(y) \in \mathbb{R}^{n \times d}$ can be viewed as multiplying $n \times d$ matrix (A_3) and $d \times d$ matrix (y), this can be computed in $\mathcal{T}_{\mathrm{mat}}(n,d,d)$.

We also have

$$\underbrace{f(x)}_{n \times n} = \underbrace{D(X)^{-1}}_{n \times n} \exp(\underbrace{A_1}_{n \times d} \underbrace{X}_{d \times d} \underbrace{A_2^\top}_{d \times n}), \quad \text{and} \quad D(X) = \mathrm{diag}(\exp(A_1 X A_2^\top) \mathbf{1}_n) \tag{6}$$

Then the computation of $f(x) \in \mathbb{R}^{n \times n}$ can be done in $\mathcal{T}_{\mathrm{mat}}(n,n,d) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$.

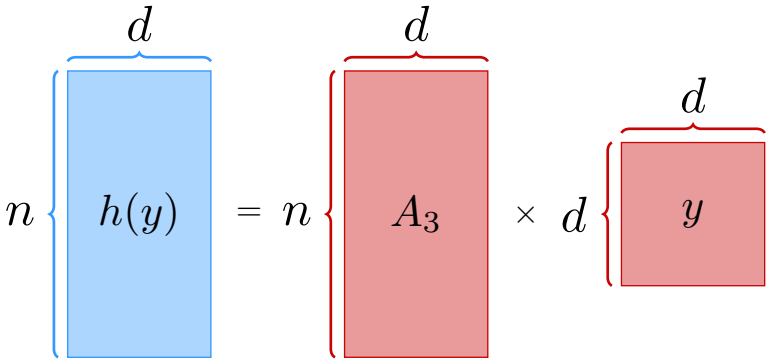

Figure 8: The visualization of Eq. (5). We have $A_3 \in \mathbb{R}^{n \times d}$. $h : \mathbb{R}^{d \times d} \to \mathbb{R}^{n \times d}$ is a function, which maps the matrix $y \in \mathbb{R}^{d \times d}$ to $h(y)$ by multiplying A_3 and y. The red rectangles represent matrices which are the factors, and the blue rectangle represents the matrix which is the product.

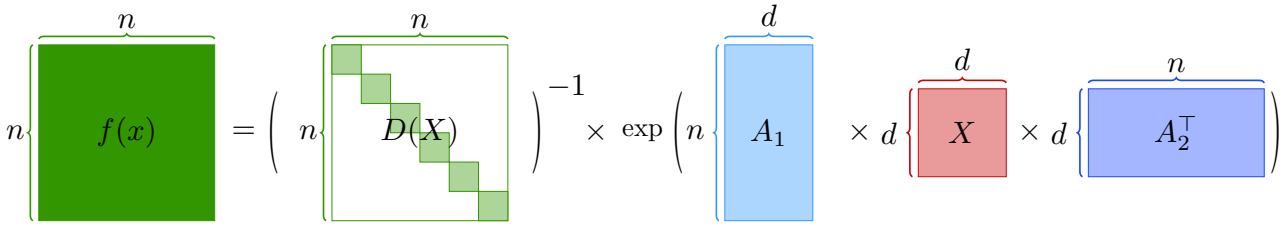

Figure 9: The visualization of Eq. (6). We have $A_1, A_2 \in \mathbb{R}^{n \times d}$, $X \in \mathbb{R}^{d \times d}$, and $D(X) \in \mathbb{R}^{n \times n}$ (see Definition 1.2 and Figure 1). First, we find the inverse of the matrix $D(X)$ and compute $\exp(A_1 X A_2^\top) \in \mathbb{R}^{n \times n}$, as shown in Figure 1. Then, we multiply $D(X)^{-1}$ and $\exp(A_1 X A_2^\top)$ to get $f(x) \in \mathbb{R}^{n \times n}$. The green squares represent the square matrices in $\mathbb{R}^{n \times n}$. The blue rectangles represent the matrices in $\mathbb{R}^{n \times d}$ (the dark blue denotes the transpose of the matrix in $\mathbb{R}^{n \times d}$). The red square represents the square matrices in $\mathbb{R}^{d \times d}$.

Given that

$$\underbrace{c(x,y)}_{n \times d} = \underbrace{f(x)}_{n \times n} \underbrace{h(y)}_{n \times d} - \underbrace{B}_{n \times d} \tag{7}$$

Then c can be done in $\mathcal{T}_{\mathrm{mat}}(n, n, d)$.

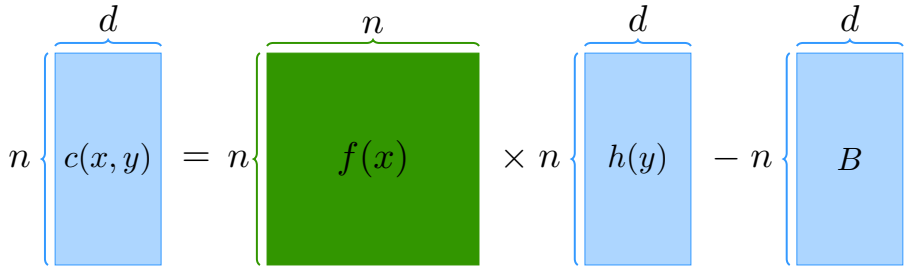

Figure 10: The visualization of Eq. (7). Let $f(x) \in \mathbb{R}^{n \times n}$ (see Figure 9) and $h(y) \in \mathbb{R}^{n \times d}$ (see Figure 8). We have $B \in \mathbb{R}^{n \times d}$. We multiply $f(x)$ with $h(y)$ and subtract B from their product to get $c(x, y) \in \mathbb{R}^{n \times d}$. The green square represents the square matrices in $\mathbb{R}^{n \times n}$. The blue rectangles represent the matrix in $\mathbb{R}^{n \times d}$.

□

B.4 REFORMULATING GRADIENT (x) IN MATRIX VIEW

In this section, we reformulate the gradient x in the matrix's view.

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

- $\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x_i} = c(x,y)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0} \rangle - \langle f(x)_{j_0}, h(y)_{i_0} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle)$
- *Let* $c(x,y) \in \mathbb{R}^{n \times d}$
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$
- *Let* $v = h(y)_{i_0} \in \mathbb{R}^n$
- *Let* $\frac{\mathrm{d}L(x,y)}{\mathrm{d}x} = \sum_{j_0=1}^n \sum_{i_0=1}^d \frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x}$
- *Let*

$$q(x,y)_{j_0} = \sum_{i_0=1}^d c(x,y)_{j_0,i_0} h(y)_{i_0}$$

then, we have

- **Part 1.**

$$\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x} = \underbrace{c(x,y)_{j_0,i_0}}_{\text{scalar}} \cdot \underbrace{\mathsf{A}_{j_0}^\top}_{d^2 \times n} \underbrace{(\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}^\top)}_{n \times n} \underbrace{h(y)_{i_0}}_{n \times 1}$$

- **Part 2.** *Suppose* $c(x,y), \mathsf{A}, f(x), h(y)$ *are given, then* $\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x}$ *can be computed in* $O(nd^2)$ *time.*
- **Part 3.**

$$\frac{\mathrm{d}L(x,y)}{\mathrm{d}x} = \sum_{j_0=1}^n \underbrace{\mathsf{A}_{j_0}^\top}_{d^2 \times n} \underbrace{(\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}^\top)}_{n \times n} \underbrace{q(x,y)_{j_0}}_{n \times 1}$$

- **Part 4.** *Suppose* $c(x,y), \mathsf{A}, f(x), h(y)$ *are given, then* $\frac{\mathrm{d}L(x,y)}{\mathrm{d}x} \in \mathbb{R}^{d^2}$ *can be computed in* $\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$ *time*

Proof. **Proof of Part 1.**

From the Lemma statement, we have

$$\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x_i} = c(x,y)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0} \rangle - \langle f(x)_{j_0}, h(y)_{i_0} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle) \tag{8}$$

Note that by Fact A.1, we have

$$\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0} \rangle = \mathsf{A}_{j_0,i}^\top \mathrm{diag}(f(x)_{j_0}) h(y)_{i_0}$$

and

$$\langle f(x)_{j_0}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle = \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top h(y)_{i_0}$$

Therefore, Eq. (8) becomes

$$\begin{aligned} \frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x_i} &= c(x,y)_{j_0,i_0} \cdot (\mathsf{A}_{j_0,i}^\top \mathrm{diag}(f(x)_{j_0}) h(y)_{i_0} - \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top h(y)_{i_0}) \\ &= c(x,y)_{j_0,i_0} \cdot \mathsf{A}_{j_0,i}^\top (\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}^\top) h(y)_{i_0}, \end{aligned}$$

where the second step follows from simple algebra.

Thus, we complete the proof.

Proof of Part 2.

We first compute $(\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}^\top) h(y)_{i_0}$, this can be done in $O(n)$ time.

Then we can compute the rest, it takes $O(nd^2)$ time.

Proof of Part 3 and Part 4.

Firstly, we can compute $q(x,y)_{j_0} \in \mathbb{R}^n$.

Recall from the Lemma statement, we have

$$q(x,y)_{j_0} = \sum_{i_0=1}^d c(x,y)_{j_0,i_0} h(y)_{i_0}. \tag{9}$$

Let $q(x,y)_{j_0} \in \mathbb{R}^n$ denote the j_0-th column of $q(x,y)$.

Then we have

$$q(x,y) = \underbrace{h(y)}_{n \times d} \underbrace{c(x,y)^\top}_{d \times n}$$

This takes $\mathcal{T}_{\mathrm{mat}}(n,d,n)$ time.

Then, we compute

$$p(x,y)_{j_0} = (\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}) q(x,y)_{j_0}. \tag{10}$$

This takes $O(n^2)$ time in total.

We can show that

$$\begin{aligned}
& \frac{\mathrm{d}L(x,y)}{\mathrm{d}x} \\
= & \sum_{j_0=1}^n \sum_{i_0=1}^d \frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}x} \\
= & \sum_{j_0=1}^n \sum_{i_0=1}^d \underbrace{c(x,y)_{j_0,i_0}}_{\mathrm{scalar}} \cdot \underbrace{\mathsf{A}_{j_0}^\top}_{d^2 \times n} \underbrace{(\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}^\top)}_{n \times n} \underbrace{h(y)_{i_0}}_{n \times 1} \\
= & \sum_{j_0=1}^n \mathsf{A}_{j_0}^\top (\mathrm{diag}(f(x)_{j_0}) - f(x)_{j_0} f(x)_{j_0}) q(x,y)_{j_0} \\
= & \sum_{j_0=1}^n \mathsf{A}_{j_0}^\top p(x,y)_{j_0} \\
= & \mathrm{vec}(A_1^\top p(x,y) A_2)
\end{aligned}$$

where the first step is based on Definition A.7, the second step is because of **Part 1**, the third step is due to Eq. (9), the fourth step follows from Eq. (10), and the last step due to tensor-trick.

Note that $A_1^\top p(x,y) A_2$ can be computed in $\mathcal{T}_{\mathrm{mat}}(n,d,n) + \mathcal{T}_{\mathrm{mat}}(d,n,d)$ time. □

B.5 REFORMULATING GRADIENT (y) IN MATRIX VIEW

In this section, we reformulate the gradient y in the matrix's view.

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

- *if* $i_1 = i_0$, $\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = c(x,y)_{j_0,i_0}\langle f(x)_{j_0}, A_{3,*,i_2}\rangle$
- *if* $i_1 \neq i_0$, $\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}y_{i_1,i_2}} = 0$
- *Let* $\frac{\mathrm{d}L(x,y)}{\mathrm{d}y_{i_0,i_2}} = \sum_{j_0=1}^n c(x,y)_{j_0,i_0}\langle f(x)_{j_0}, A_{3,*,i_2}\rangle$
- *Let* $\widetilde{q}(x,y)_{i_0} = \sum_{j_0=1}^n f(x)_{j_0} c(x,y)_{j_0,i_0}$

Then we have

- **Part 1.**

$$\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}y_{i_0,i_2}} = \underbrace{A_{3,*,i_2}^\top}_{1\times n}\underbrace{f(x)_{j_0}}_{n\times 1}\underbrace{c(x,y)_{j_0,i_0}}_{\text{scalar}}$$

- **Part 2.**

$$\frac{\mathrm{d}L(x,y)}{\mathrm{d}y_{i_0,i_2}} = \underbrace{A_{3,*,i_2}^\top}_{1\times n}\underbrace{\widetilde{q}(x,y)_{i_0}}_{n\times 1}$$

- **Part 3.**

$$\frac{\mathrm{d}L(x,y)}{\mathrm{d}y} = \mathrm{vec}(\underbrace{A_3^\top}_{d\times n}\underbrace{\widetilde{q}(x,y)}_{n\times d})$$

- **Part 4**. *Computing* $\frac{\mathrm{d}L(x,y)}{\mathrm{d}y}$ *takes* $\mathcal{T}_{\mathrm{mat}}(n,n,d) + \mathcal{T}_{\mathrm{mat}}(n,d,d)$

Proof. **Proof of Part 1.**

$$\begin{aligned}\frac{\mathrm{d}L(x,y)_{j_0,i_0}}{\mathrm{d}y_{i_0,i_2}} &= c(x,y)_{j_0,i_0}\langle f(x)_{j_0}, A_{3,*,i_2}\rangle \\ &= A_{3,*,i_2}^\top f(x)_{j_0} c(x,y)_{j_0,i_0}\end{aligned}$$

where the first step comes from the assumption from the Lemma statement and the second step is based on Fact A.1.

Proof of Part 2.

$$\begin{aligned}\frac{\mathrm{d}L(x,y)}{\mathrm{d}y_{i_0,i_2}} &= \sum_{j_0=1}^n c(x,y)_{j_0,i_0}\langle f(x)_{j_0}, A_{3,*,i_2}\rangle \\ &= \sum_{j_0=1}^n A_{3,*,i_2}^\top f(x)_{j_0} c(x,y)_{j_0,i_0} \\ &= A_{3,*,i_2}^\top \widetilde{q}(x,y)_{i_0}\end{aligned}$$

where the first step is due to the assumption from the Lemma statement, the second step is because of Fact A.1, and the last step comes from the definition of $\widetilde{q}(x,y)_{i_0}$ (see from the Lemma statement).

Proof of Part 3.

$$\frac{\mathrm{d}L(x,y)}{\mathrm{d}y} = \mathrm{vec}(A_3^\top \widetilde{q}(x,y))$$

where the first step comes from tensor trick based on **Part 2**.

Proof of Part 4. Computing $\widetilde{q}(x,y) \in \mathbb{R}^{n\times d}$ takes $\mathcal{T}_{\mathrm{mat}}(n,n,d)$ time.

Computing $A_3^\top \widetilde{q}(x,y)$ takes $\mathcal{T}_{\mathrm{mat}}(n,d,d)$ time.

□

C HESSIAN

In this section, we provide more details related to Hessian.

Finally the hessian $H \in \mathbb{R}^{2d^2 \times 2d^2}$ which can be written as

$$H = \begin{bmatrix} H_{x,x} & H_{x,y} \\ H_{y,x} & H_{y,y} \end{bmatrix}$$

where

- $H_{x,x} \in \mathbb{R}^{d^2 \times d^2}$ is $\frac{\mathrm{d}^2 L}{\mathrm{d}x \mathrm{d}x}$ (see details in Section D)
- $H_{x,y}$, $H_{y,x} \in \mathbb{R}^{d^2 \times d^2}$ is $\frac{\mathrm{d}^2 L}{\mathrm{d}x \mathrm{d}y}$ (see details in Section H)
- $H_{y,y} \in \mathbb{R}^{d^2 \times d^2}$ is $\frac{\mathrm{d}^2 L}{\mathrm{d}y \mathrm{d}y}$ (see details in Section G)
  - We can view $H_{y,y} = \begin{bmatrix} H_{y,y,1,1} & 0 & 0 & \cdots & 0 \\ 0 & H_{y,y,2,2} & 0 & \cdots & 0 \\ 0 & 0 & H_{y,y,3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & H_{y,y,d,d} \end{bmatrix}$
  - where $H_{y,y,i_0,i_0} = \sum_{j_0=1}^n \frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0,*} \mathrm{d}y_{i_0,*}} \in \mathbb{R}^{d \times d}$ for each $i_0 \in [d]$

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

- $H_{x,x} \succeq \alpha_1 I_{d^2}$
- $H_{y,y} \succeq \alpha_2 I_{d^2}$
- $\|H_{x,y}\| \leq \alpha_3$
- $\|H_{y,x}\| \leq \alpha_3$
- *Let* $\alpha_1 \geq \alpha_3 > 0$, $\alpha_2 \geq \alpha_3 > 0$

Then we have

$$H \succeq \min\{\alpha_1 - \alpha_3, \alpha_2 - \alpha_3\} \cdot I_{2d^2}$$

Proof. Let $u, v \in \mathbb{R}^{d^2}$, then we have

$$\begin{aligned} \begin{bmatrix} u^\top & v^\top \end{bmatrix} H \begin{bmatrix} u \\ v \end{bmatrix} &= u^\top H_{x,x} u + v^\top H_{y,y} v + u^\top H_{x,y} v + v^\top H_{y,x} u \\ &\geq \|u\|_2^2 \cdot \alpha_1 + \|v\|_2^2 \cdot \alpha_2 + u^\top H_{x,y} v + v^\top H_{y,x} u \\ &\geq \|u\|_2^2 \cdot \alpha_1 + \|v\|_2^2 \cdot \alpha_2 - \|u\|_2 \|v\|_2 (\|H_{x,y}\| + \|H_{y,x}\|) \\ &\geq \|u\|_2^2 \cdot \alpha_1 + \|v\|_2^2 \cdot \alpha_2 - \|u\|_2 \|v\|_2 2\alpha_3 \\ &\geq \|u\|_2^2 \cdot \alpha_1 + \|v\|_2^2 \cdot \alpha_2 - (\|u\|_2^2 + \|v\|_2^2)\alpha_3 \\ &\geq (\|u\|_2^2 + \|v\|_2^2) \cdot \min\{\alpha_1 - \alpha_3, \alpha_2 - \alpha_3\} \end{aligned}$$

where the first step is based on the expansion of H, the second step is due to $H_{x,x} \succeq \alpha_1 I_{d^2}, H_{y,y} \succeq \alpha_2 I_{d^2}$, the third step comes from Fact A.2 and Fact A.3 , the fourth step is because of $\|H_{x,y}\| \leq \alpha_3, \|H_{y,x}\| \leq \alpha_3$, the fifth step is owing to $2\|u\|_2 \|v\|_2 \leq \|u\|_2^2 + \|v\|_2^2$, and the last step is based on the simple algebra.

Thus, it implies

$$H \succeq \min\{\alpha_1 - \alpha_3, \alpha_2 - \alpha_3\} \cdot I_{2d^2}$$

□

D HESSIAN FOR X

In Section D.1, we compute the Hessian matrix with respect to x. In Section D.2, we present a helpful lemma to simplify the Hessian. In Section D.3, we define $B(x)$, representing the Hessian.

D.1 HESSIAN

Now, we start to compute the Hessian matrix with respect to x.

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

- *Let* $\gamma(x)_{j_0} := \langle f(x)_{j_0}, v \rangle$ *(We define this notation for easy of writing proofs.)*

Then we have for each $i \in [d^2]$, $l \in [d^2]$

- *Part 1.* $i = l$ *Hessian diagonal term*

$$\begin{aligned}\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x_i \mathrm{d}x_i} = & (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle - \gamma_{j_0}(x) \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle)^2 \\ & + c(x,:)_{j_0,i_0} \cdot \\ & (\\ & + \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v \rangle (1 - \gamma_{j_0}(x)) \\ & - 2 \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle \\ & + 2 \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle^2 \cdot \gamma_{j_0}(x) \\ &)\end{aligned}$$

- *Part 2.* $i \neq l$ *Hessian off-diagonal term*

$$\begin{aligned}\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x_i \mathrm{d}x_l} = & (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle - \gamma_{j_0}(x) \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle) \\ & \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v \rangle - \gamma_{j_0}(x) \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l} \rangle) \\ & + c(x,:)_{j_0,i_0} \cdot \\ & (\\ & + \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v \rangle (1 - \langle f(x)_{j_0}, v \rangle)) \\ & - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l} \rangle - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle \\ & + 2 \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle \langle f(x)_{j_0}, \mathsf{A}_{j_0,l} \rangle \cdot \gamma_{j_0}(x) \\ &)\end{aligned}$$

Proof. **Proof of Part 1.**

At first, we have

$$\begin{aligned}& \frac{\mathrm{d}}{\mathrm{d}x_i} (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle - \langle f(x)_{j_0}, v \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle) \\ = & \underbrace{\frac{\mathrm{d}}{\mathrm{d}x_i} \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v \rangle}_{\text{Part 7 of Lemma B.1}} \\ & - \underbrace{(\frac{\mathrm{d}}{\mathrm{d}x_i} \langle f(x)_{j_0}, v \rangle)}_{\text{Part 4 of Lemma B.1}} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle \\ & - \underbrace{(\frac{\mathrm{d}}{\mathrm{d}x_i} \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle)}_{\text{Part 9 of Lemma B.1}} \cdot \langle f(x)_{j_0}, v \rangle\end{aligned}$$

$$
\begin{aligned}
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad - (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad - (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, \mathsf{A}_{j_0,i}\rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \cdot \langle f(x)_{j_0}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v\rangle \\
&\quad - 2\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad + 2\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle^2 \cdot \langle f(x)_{j_0}, v\rangle \\
&\quad - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0}, v\rangle
\end{aligned}
$$

where the first step is based on the product rule of derivative, the second step comes from **Part 4, Part 7, and Part 9** of Lemma B.1, and the last step is due to simple algebra.

Then we can show that

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}x_i}(\frac{\mathrm{d}}{\mathrm{d}x_i} L_{j_0,i_0}) \\
&= \frac{\mathrm{d}}{\mathrm{d}x_i}(c(x,:)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)) \\
&= (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)^2 \\
&\quad + c(x,:)_{j_0,i_0} \cdot \frac{\mathrm{d}}{\mathrm{d}x_i}(\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle),
\end{aligned}
$$

where the first step comes from **Part 6** of Lemma B.1 and the second step is due to **Part 5** of Lemma B.1.

Combining the above two equations, we complete the proof.

Proof of Part 2.

Firstly, we can show that

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}x_l}(\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \\
&= \underbrace{\frac{\mathrm{d}}{\mathrm{d}x_l}\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle}_{\text{Part 8 of Lemma B.1}} \\
&\quad - \underbrace{(\frac{\mathrm{d}}{\mathrm{d}x_l}\langle f(x)_{j_0}, v\rangle)}_{\text{Part 4 of Lemma B.1}} \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad - \underbrace{(\frac{\mathrm{d}}{\mathrm{d}x_l}\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)}_{\text{Part 10 of Lemma B.1}} \cdot \langle f(x)_{j_0}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad - (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle) \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad - (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, \mathsf{A}_{j_0,l}\rangle - \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle\langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle) \cdot \langle f(x)_{j_0}, v\rangle \\
&= \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle \\
&\quad - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
&\quad + 2\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle\langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle \cdot \langle f(x)_{j_0}, v\rangle \\
&\quad - \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle \cdot \langle f(x)_{j_0}, v\rangle
\end{aligned}
$$

where the first step is owing to the product rule of derivative, the second step is based on **Part 4, Part 8, and Part 10** of Lemma B.1, and the last step comes from simple algebra.

We have

$$
\frac{\mathrm{d}}{\mathrm{d}x_l}(\frac{\mathrm{d}}{\mathrm{d}x_i} L_{j_0,i_0})
$$

$$
\begin{aligned}
&= \frac{\mathrm{d}}{\mathrm{d}x_l}(c(x,:)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)) \\
&= (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \\
&\quad \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle) \\
&\quad + c(x,:)_{j_0,i_0} \cdot \frac{\mathrm{d}}{\mathrm{d}x_l}(\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle - \langle f(x)_{j_0}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)
\end{aligned}
$$

Combining the above two equations, we complete the proof. □

D.2 A HELPFUL LEMMA

In this section, we present a helpful Lemma.

Lemma D.2. *We have*

- *Part 1.*

$$
\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle = \underbrace{\mathsf{A}_{j_0,i}^\top}_{d^2 \times n} \underbrace{\mathrm{diag}(f(x)_{j_0} \circ v)}_{n \times n} \underbrace{\mathsf{A}_{j_0,l}}_{n \times d^2}
$$

- *Part 2.*

$$
\begin{aligned}
&\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle + \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
= \ & \mathsf{A}_{j_0,i}^\top (\underbrace{(f(x)_{j_0} \circ v)(f(x)_{j_0})^\top + f(x)_{j_0}(f(x)_{j_0} \circ v)^\top}_{\mathrm{rank}-2}) \mathsf{A}_{j_0,l}
\end{aligned}
$$

- *Part 3.*

$$
\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle = \mathsf{A}_{j_0,i}^\top \underbrace{(f(x)_{j_0} \circ v)(f(x)_{j_0} \circ v)^\top}_{\mathrm{rank}-1} \mathsf{A}_{j_0,l}
$$

- *Part 4.*

$$
\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle = \mathsf{A}_{j_0,i}^\top \underbrace{(f(x)_{j_0})(f(x)_{j_0})^\top}_{\mathrm{rank}-1} \mathsf{A}_{j_0,l}
$$

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

$$
\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i} \circ \mathsf{A}_{j_0,l}, v\rangle = \mathsf{A}_{j_0,i}^\top \mathrm{diag}(f(x)_{j_0} \circ v) \mathsf{A}_{j_0,l}
$$

where the first step follows from Fact A.1.

Proof of Part 2. We have

$$
\begin{aligned}
&\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle + \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\
= \ & \langle f(x)_{j_0} \circ v, \mathsf{A}_{j_0,i}\rangle \cdot f(x)_{j_0}^\top \mathsf{A}_{j_0,l} \\
& + \langle f(x)_{j_0} \circ v, \mathsf{A}_{j_0,l}\rangle \cdot \mathsf{A}_{j_0,i}^\top \cdot f(x)_{j_0} \\
= \ & \mathsf{A}_{j_0,i}^\top \cdot (f(x)_{j_0} \circ v)(f(x)_{j_0})^\top \mathsf{A}_{j_0,i} \\
& + \mathsf{A}_{j_0,i}^\top f(x)_{j_0}(f(x)_{j_0} \circ v)^\top \mathsf{A}_{j_0,l} \\
= \ & \mathsf{A}_{j_0,i}^\top ((f(x)_{j_0} \circ v)(f(x)_{j_0})^\top \\
& + f(x)_{j_0}(f(x)_{j_0} \circ v)^\top) \mathsf{A}_{j_0,l}
\end{aligned}
$$

where the first step follows from Fact A.1, the second step follows from Fact A.1, and the last step follows from the simple algebra.

Proof of Part 3. We have

$$\begin{aligned}\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, v\rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,l}, v\rangle &= \langle f(x)_{j_0} \circ v, \mathsf{A}_{j_0,i}\rangle \langle f(x)_{j_0} \circ v, \mathsf{A}_{j_0,l}\rangle \\ &= \mathsf{A}_{j_0,i}^\top (f(x)_{j_0} \circ v)(f(x)_{j_0} \circ v)^\top \mathsf{A}_{j_0,l}\end{aligned}$$

where the first step follows from Fact A.1, and the last step follows from Fact A.1.

Proof of Part 4. We have

$$\langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,l}\rangle = \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top \mathsf{A}_{j_0,l}$$

where the first step follows from Fact A.1. □

D.3 DEFINING $B(x)$

In this section, we formally define $B(x)$.

Definition D.3. *If the following conditions hold*

- *Let* $\gamma_{j_0}(x) = \langle f(x)_{j_0}, v\rangle$

We define $B(x) \in \mathbb{R}^{n\times n}$ *as follows*

$$\begin{aligned}B(x) :=& B_{\mathrm{diag}}^1 \\ &+ B_{\mathrm{rank}}^1 + B_{\mathrm{rank}}^2 + B_{\mathrm{rank}}^3\end{aligned}$$

where

- $B_{\mathrm{diag}}^1 := (1 - \gamma_{j_0}(x)) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$

and

- $B_{\mathrm{rank}}^1 := -(2\gamma_{j_0}(x) + c(x,:)_{j_0,i_0}) \cdot ((f(x)_{j_0} \circ v) f(x)_{j_0}^\top + f(x)_{j_0}(f(x)_{j_0} \circ v)^\top)$
- $B_{\mathrm{rank}}^2 := (2\gamma_{j_0}(x) c(x,:)_{j_0,i_0} + \gamma_{j_0}(x)^2) \cdot f(x)_{j_0} f(x)_{j_0}^\top$
- $B_{\mathrm{rank}}^3 := (f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$

Lemma D.4. *Let* $B(x)$ *be defined as Definition D.3, then we have*

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}x} = \underbrace{\mathsf{A}_{j_0}^\top}_{d^2\times n} \underbrace{B(x)}_{n\times n} \underbrace{\mathsf{A}_{j_0}}_{n\times d^2}$$

Proof. The proof follows by combining Lemma D.1 and Lemma D.2. □

E LIPSCHITZ PROPERTY OF $H_{x,x}$

In Section E.1, we present the main results of the Lipschitz property of $H_{x,x}$. In Section E.2, we summarize the results from following steps 1-9. In Section E.3, we compute the upper bound of basic functions for the following proof. In Section E.4, we compute the Lipschitz Property of basic functions for the following proof. In Section E.5, we analyze the first step of Lipschitz function $c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$. In Section E.6, we analyze the second step of Lipschitz function $-\gamma_{j_0}(x) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$. In Section E.7, we analyze the third step of Lipschitz function $-2\gamma_{j_0}(x) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$. In Section E.8, we analyze the fourth step of Lipschitz function $-c(x,:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$. In Section E.9, we analyze the fifth step of Lipschitz function $-2\gamma_{j_0}(x) \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$. In Section E.10, we analyze the sixth step of Lipschitz function $-c(x,:)_{j_0,i_0}) \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$. In Section E.11, we analyze the seventh step of Lipschitz function $2\gamma_{j_0}(x) c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top$. In Section E.12, we analyze the eighth step of Lipschitz function $\gamma_{j_0}(x)^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top$. In Section E.13, we analyze the nineth step of Lipschitz function $(f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$.

E.1 MAIN RESULT

In this section, we present the main result of the Lipschitz property.

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

- *Let* $H_{j_0,i_0} = \frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}x} : \mathbb{R}^{d^2} \to \mathbb{R}^{d^2 \times d^2}$
- *Let* $H = \sum_{j_0=1}^n \sum_{i_0=1}^d H_{j_0,i_0}$ *(because* $L = \sum_{j_0=1}^n \sum_{i_0=1}^d L_{j_0,i_0}$*)*
- *Let* $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$ *and* $u(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.8*
- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$
- $\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$
- *Let* $R \geq 4$
- *Let* $M := \exp(O(R^2 + \log(nd)))$

Then, we have for all $x, \widetilde{x} \in \mathbb{R}^{d^2}$

- *Part 1. For each* $j_0 \in [n]$, $i_0 \in [d]$

$$\|H_{j_0,i_0}(x) - H_{j_0,i_0}(\widetilde{x})\| \leq M \cdot \|x - \widetilde{x}\|_2$$

- *Part 2.*

$$\|H(x) - H(\widetilde{x})\| \leq M \cdot \|x - \widetilde{x}\|_2$$

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

$$\begin{aligned} \|H_{j_0,i_0}(x) - H_{j_0,i_0}(\widetilde{x})\| \leq & \sum_{k=1}^9 \| \mathsf{A}_{j_0}^\top \| \cdot \|G_k(x) - G_k(\widetilde{x})\| \cdot \| \mathsf{A}_{j_0} \| \\ \leq & 9R^2 \cdot n^{1.5} \exp(20R^2) \\ \leq & n^{1.5} \exp(30R^2) \end{aligned}$$

where the first step follows from definition of $H_{j_0,i_0}(x)$, the second step follows from Lemma E.2, and last step follows from simple algebra.

Proof of Part 2.

Then, we have

$$\begin{aligned} \|H(x) - H(\widetilde{x})\| \leq & \sum_{j_0=1}^n \sum_{i_0=1}^d \|H_{j_0,i_0}(x) - H_{j_0,i_0}(\widetilde{x})\| \\ \leq & nd \cdot n^{1.5} \exp(30R^2) \end{aligned}$$

where the first step follows from triangle inequality and $H = \sum_{j_0=1}^n \sum_{i_0=1}^d H_{j_0,i_0}$, and the second step follows from **Part 1.** □

E.2 SUMMARY OF NINE STEPS

In this section, we provide a summary of the nine-step calculation of Lipschitz for different matrix functions.

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

- $G_1(x) = c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$
- $G_2(x) = -\gamma_{j_0}(x) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$
- $G_3(x) = -2\gamma_{j_0}(x) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$
- $G_4(x) = -c(x,:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$
- $G_5(x) = -2\gamma_{j_0}(x) \cdot f(x)_{j_0} (f(x)_{j_0} \circ v)^\top$ *(The proof of this is identical to G_3)*
- $G_6(x) = -c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} (f(x)_{j_0} \circ v)^\top$ *(The proof of this is identical to G_4)*
- $G_7(x) = 2\gamma_{j_0}(x) c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top$
- $G_8(x) = \gamma_{j_0}(x)^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top$
- $G_9(x) = (f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$

Then, we have

$$\max_{k \in [9]} \|G_k(x) - G_k(\widetilde{x})\| \leq n^{1.5} \exp(20R^2).$$

Proof. The proof follows from Lemma E.7, Lemma E.8, Lemma E.9, Lemma E.10, Lemma E.11, Lemma E.12, Lemma E.13, Lemma E.14, and Lemma E.15. □

E.3 A CORE TOOL: UPPER BOUND FOR SEVERAL BASIC FUNCTIONS

In this section, we analyze the upper bound of several basic functions.

Lemma E.3 (Lemma 8.9 in Deng et al. [2023a] page 44 and Lemma 6.2 in Gao et al. [2023b] page 20)**.** *Provided that the subsequent requirements are satisfied*

- *Let* $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$ *satisfy* $\max_{j_0 \in [n]} \| \mathsf{A}_{j_0} \| \leq R$
- *Let* $x \in \mathbb{R}^{d^2}$ *satisfy that* $\|x\|_2 \leq R$
- *We define* $u(x)$ *as Definition A.8*
- *Let* β *be the greatest lower bound of* $\langle u(x)_{j_0}, \mathbf{1}_n \rangle$

Then we have

$$\beta \geq \exp(-R^2).$$

Lemma E.4 (Basic Functions Upper Bound)**.** *If the following conditions hold,*

- *Let* $u(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.8*
- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$
- *Let* β *be the greatest lower bound of* $\langle u(x)_{j_0}, \mathbf{1}_n \rangle$
- $\|A_1\|, \|A_2\|, \|A_3\| \leq R$
- $\| \mathsf{A}_{j_0} \| \leq R$
- $\|x\|_2 \leq R$

- $|b_{j_0,i_0}| \leq R$
- *Let* $R \geq 4$
- $\|v\|_2 \leq R^2$

Then we have: for all $x \in \mathbb{R}^{d^2}$

- *Part 1.* $\|u(x)_{j_0}\|_2 \leq \sqrt{n} \cdot \exp(R^2)$
- *Part 2.* $|\alpha(x)_{j_0}| \leq n \exp(R^2)$
- *Part 3.* $|\alpha(x)_{j_0}|^{-1} \leq \exp(R^2)$
- *Part 4.* $\|f(x)_{j_0}\|_2 \leq 1$
- *Part 5.* $|\gamma(x)_{j_0}| \leq R^2$
- *Part 6.* $|c(x,:)_{j_0,i_0}| \leq 2R^2$

Proof. We present our proof as follows.

Proof of Part 1. We have

$$\begin{aligned}
\|u(x)_{j_0}\|_2 &= \|\exp(\mathsf{A}_{j_0} x)\|_2 \\
&\leq \sqrt{n} \cdot \|\exp(\mathsf{A}_{j_0} x)\|_\infty \\
&\leq \sqrt{n} \cdot \exp(\|\mathsf{A}_{j_0} x\|_2) \\
&\leq \sqrt{n} \cdot \exp(R^2)
\end{aligned}$$

where the first step follows from Definition A.8, the second step is based on Fact A.2, the third step follows from Fact A.2, and the fourth step is because of $\|\mathsf{A}_{j_0}\| \leq R$ and $\|x\|_2 \leq R$ (see from the Lemma statement).

Proof of Part 2. We have

$$\begin{aligned}
|\alpha(x)_{j_0}| &= |\langle u(x)_{j_0}, \mathbf{1}_n \rangle| \\
&\leq \sqrt{n} \cdot \|u(x)_{j_0}\|_2 \\
&\leq \sqrt{n} \cdot \sqrt{n} \cdot \exp(R^2) \\
&= n \exp(R^2)
\end{aligned}$$

where the first step is due to Definition A.9, the second is based on Fact A.2, the third step follows from **Part 1**. and the forth step follows from simple algebra.

Proof of Part 3.

We have

$$\begin{aligned}
|\alpha^{-1}(x)_{j_0}| &= \frac{1}{\langle u(x)_{j_0}, \mathbf{1}_n \rangle} \\
&\leq \frac{1}{\beta} \\
&\leq \exp(R^2)
\end{aligned}$$

where the first step is because of Definition A.9, the second step follows from the definition of β and the third step is due to Lemma E.3.

Proof of Part 4. We have

$$\begin{aligned}
\|f(x)_{j_0}\|_2 &\leq \|f(x)_{j_0}\|_1 \\
&= 1
\end{aligned}$$

where the first step follows from Fact A.2, the second step is due to Definition A.10

Proof of Part 5. We have

$$\begin{aligned}|\gamma(x)_{j_0}| &= |\langle f(x)_{j_0}, v\rangle| \\ &\leq \|f(x)_{j_0}\|_2 \cdot \|v\|_2 \\ &\leq 1 \cdot R^2 \\ &= R^2\end{aligned}$$

where the first step follows from the definition of $\gamma(x)_{j_0}$ (see from the Lemma statement), the second step follows from Cauchy–Schwarz inequality, the third step follows from **Part 2** and the upper bound for the ℓ_2 norm of v (from the Lemma statement), and the last step follows from simple algebra.

Proof of Part 6. We have

$$\begin{aligned}|c(x,:)_{j_0,i_0}| &= |\langle f(x)_{j_0}, v\rangle - b_{j_0,i_0}| \\ &\leq |\gamma_{j_0}(x) - b_{j_0,i_0}| \\ &\leq |\gamma_{j_0}(x)| + |b_{j_0,i_0}| \\ &\leq R^2 + R \\ &\leq 2R^2\end{aligned}$$

where the first step is based on Definition A.12, the second step is because of the definition of $\gamma_{j_0}(x)$, the third step follows from triangle inequality, the fourth step is based on **Part 6** and $|b_{j_0,i_0}| \leq R$ (see from the Lemma statement), and the last step follows from $R \geq 1$. □

E.4 A CORE TOOL: LIPSCHITZ PROPERTY FOR SEVERAL BASIC FUNCTIONS

In this section, we analyze the Lipschitz property of several basic functions.

Lemma E.5 (Basic Functions Lipschitz Property)**.** *If the following conditions hold,*

- $\|v\|_2 \leq R^2$
- $\|\mathsf{A}_{j_0}\| \leq R$
- *Let β be the greatest lower bound of $\langle u(x)_{j_0}, \mathbf{1}_n\rangle$*
- *Let $\beta^{-1} \leq \exp(R^2)$*
- *Let $R \geq 4$*
- *Let $\|x\|_2 \leq R$ and $\|\widetilde{x}\|_2 \leq R$.*

Then, we have: for all $x, \widetilde{x} \in \mathbb{R}^{d^2}$

- *Part 1.* $\|u(x)_{j_0} - u(\widetilde{x})_{j_0}\|_2 \leq \sqrt{n}\exp(2R^2) \cdot \|x - \widetilde{x}\|_2$
- *Part 2.* $|\alpha(x)^{-1} - \alpha^{-1}(\widetilde{x})| \leq n\exp(4R^2) \cdot \|x - \widetilde{x}\|_2$
- *Part 3.* $\|f(x)_{j_0} - f(\widetilde{x})_{j_0}\|_2 \leq n^{1.5}R\exp(6R^2) \cdot \|x - \widetilde{x}\|_2$
- *Part 4.* $|\gamma(x)_{j_0} - \gamma(\widetilde{x})_{j_0}| \leq n^{1.5}\exp(7R^2) \cdot \|x - \widetilde{x}\|_2$
- *Part 5.* $|c(x,:)_{j_0,i_0} - c(\widetilde{x},:)_{j_0,i_0}| \leq n^{1.5}\exp(7R^2) \cdot \|x - \widetilde{x}\|_2$

Proof. **Proof of Part 1.**

We have

$$\begin{aligned}\|u(x)_{j_0} - u(\widetilde{x})_{j_0}\|_2 &= \|\exp(\mathsf{A}_{j_0}\, x) - \exp(\mathsf{A}_{j_0}\, \widetilde{x})\|_2 \\ &\leq \exp(\|\mathsf{A}_{j_0}\, x\|_2) \cdot \|\mathsf{A}_{j_0}(x - \widetilde{x})\|_2 \\ &\leq \sqrt{n}\exp(R^2) \cdot \|\mathsf{A}_{j_0}(x - \widetilde{x})\|_2 \\ &\leq \sqrt{n}\exp(R^2) \cdot \|\mathsf{A}_{j_0}\| \cdot \|x - \widetilde{x}\|_2\end{aligned}$$

$$\leq \sqrt{n}R\exp(R^2)\cdot\|x-\widetilde{x}\|_2,$$

where the first step is due to Definition A.8, the second step is because of Fact A.2, the third step is based on Fact A.2, the fourth step follows from Fact A.3, and fifth step is due to $\|\mathsf{A}_{j_0}\|\leq R$.

Proof of Part 2

We have

$$\begin{aligned}
|\alpha(x)_{j_0}^{-1}-\alpha(\widetilde{x})_{j_0}^{-1}| &\leq \alpha(x)^{-1}\alpha(\widetilde{x})^{-1}\cdot|\alpha(x)-\alpha(\widetilde{x})| \\
&\leq \beta^{-2}\cdot|\alpha(x)-\alpha(\widetilde{x})| \\
&\leq \beta^{-2}\cdot|\langle u(x)_{j_0},\mathbf{1}_n\rangle-\langle u(\widetilde{x})_{j_0},\mathbf{1}_n\rangle| \\
&\leq \beta^{-2}\cdot\sqrt{n}\|u(x)_{j_0}-u(\widetilde{x})_{j_0}\|_2 \\
&\leq 2\beta^{-2}\cdot nR\exp(R^2)\|x-\widetilde{x}\|_2 \\
&\leq n\exp(4R^2)\cdot\|x-\widetilde{x}\|_2
\end{aligned}$$

where the first step is due to simple algebra, the second step is due to $\beta\geq\langle u(x)_{j_0},\mathbf{1}_n\rangle$, the third step follows from Definition of $\alpha(x)$ (see Definition A.9), the fourth step is based on Fact A.1 and Fact A.2, the fifth step is because of **Part 1**, and the sixth step follows from $R>4$ and $\beta^{-1}\leq\exp(R^2)$.

Proof of Part 3.

We have

$$\begin{aligned}
\|f(x)_{j_0}-f(\widetilde{x})_{j_0}\|_2 &= \|\alpha(x)_{j_0}^{-1}u(x)_{j_0}-\alpha(\widetilde{x})_{j_0}^{-1}u(\widetilde{x})_{j_0}\|_2 \\
&\leq \|\alpha(x)_{j_0}^{-1}u(x)_{j_0}-\alpha(\widetilde{x})_{j_0}^{-1}u(x)_{j_0}\|_2+\|\alpha(\widetilde{x})_{j_0}^{-1}u(x)_{j_0}-\alpha(\widetilde{x})_{j_0}^{-1}u(\widetilde{x})_{j_0}\|_2 \\
&= |\alpha(x)_{j_0}^{-1}-\alpha(\widetilde{x})_{j_0}^{-1}|\cdot\|u(x)_{j_0}\|_2+|\alpha(\widetilde{x})_{j_0}^{-1}|\cdot\|u(x)_{j_0}-u(\widetilde{x})_{j_0}\|_2 \\
&\leq n^{1.5}\exp(6R^2)\cdot\|x-\widetilde{x}\|_2
\end{aligned}$$

where the first step is due to Definition A.10, the second step is based on triangle inequality, the third step follows from Fact A.2, the fourth follows from combination of **Part 1**, **Part 2** and Lemma E.4.

Proof of Part 4.

We have

$$\begin{aligned}
|\gamma_{j_0}(x)-\gamma_{j_0}(\widetilde{x})| &= |\langle f(x)_{j_0},v\rangle-\langle f(\widetilde{x})_{j_0},v\rangle| \\
&\leq |\langle f(x)_{j_0}-f(\widetilde{x})_{j_0},v\rangle| \\
&\leq \|v\|_2\cdot\|f(x)_{j_0}-f(\widetilde{x})\|_2 \\
&\leq n^{1.5}\exp(7R^2)\cdot\|x-\widetilde{x}\|_2
\end{aligned}$$

where the first step is based on the definition of $\gamma_{j_0}(x)$, the second is because of Fact A.1, the third step is due to Cauchy–Schwarz inequality, and the last step follows from **Part 3**, $\|v\|\leq R^2$ and $R\geq 4$.

Proof of Part 5.

We have

$$\begin{aligned}
|c(x,:)_{j_0,i_0}-c(\widetilde{x},:)_{j_0,i_0}| &= |\langle f(x)_{j_0},v\rangle-\langle f(\widetilde{x})_{j_0},v\rangle| \\
&\leq |\gamma_{j_0}(x)-\gamma_{j_0}(\widetilde{x})| \\
&\leq n^{1.5}\exp(7R^2)\cdot\|x-\widetilde{x}\|_2
\end{aligned}$$

where the first step follows from Definition A.12, the second step is based on the definition of $\gamma_{j_0}(x)$ and the last step follows from **Part 4**. □

For convenient, we define

Definition E.6. *We define R_0 as follows*

$$R_0 := n^{1.5}\exp(10R^2).$$

E.5 CALCULATION: STEP 1 LIPSCHITZ FOR MATRIX FUNCTION $c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$

In this section, we introduce our calculation of Lipschitz for $c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$.

Lemma E.7. *If the following conditions*

- *Let* $G_1(x) = c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$
- *Let* R_0 *be defined as Definition E.6*
- *Let* $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$ *and* $u(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.8*
- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$
- $\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\|\mathsf{A}_{j_0}\| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$
- *Let* $R \geq 4$

Then, we have

$$\|G_1(x) - G_1(\widetilde{x})\| \leq 10R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2$$

Proof. We define

$$\begin{aligned} G_{1,1} &= c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) \\ G_{1,2} &= c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(\widetilde{x})_{j_0} \circ v) \end{aligned}$$

we have

$$\begin{aligned} \|G_{1,1}\| &= \|c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)\| \\ &\leq |c(x,:)_{j_0,i_0} - c(\widetilde{x},:)_{j_0,i_0}| \cdot \|\mathrm{diag}(f(x)_{j_0} \circ v)\| \\ &\leq R^2 \cdot |c(x,:)_{j_0,i_0} - c(\widetilde{x},:)_{j_0,i_0}| \\ &\leq R^2 R_0 \cdot \|x - \widetilde{x}\|_2 \end{aligned}$$

where the first step is based on definition $G_{1,1}$, the second step is due to Fact A.3, the third step follows from Lemma E.4, and the fourth step is because of Lemma E.5.

Additionally, we have

$$\begin{aligned} \|G_{1,2}\| &= \|c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(\widetilde{x})_{j_0} \circ v)\| \\ &\leq |c(\widetilde{x},:)_{j_0,i_0}| \cdot \|v\|_2 \cdot \|\mathrm{diag}(f(x)_{j_0}) - \mathrm{diag}(f(\widetilde{x})_{j_0})\| \\ &\leq 2R^4 \cdot \|f(x)_{j_0} - f(\widetilde{x})_{j_0}\|_2 \\ &\leq 2R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2 \end{aligned}$$

where the first step is because of definition of $G_{1,2}$, the second step is due to Fact A.3, the third step follows from Lemma E.4, and the fourth step is because of Lemma E.5.

Combining the above two equations, we complete the proof. □

E.6 CALCULATION: STEP 2 LIPSCHITZ FOR MATRIX FUNCTION $-\gamma_{j_0}(x) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$

In this section, we introduce our calculation of Lipschitz for $-\gamma_{j_0}(x) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$.

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

- *Let* $G_2(x) = -\gamma_{j_0}(x) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$

- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$
- *Let* $R \geq 4$

Then, we have

$$\|G_2(x) - G_2(\widetilde{x})\| \leq 10R^4 \cdot R_0 \|x - \widetilde{x}\|_2$$

Proof. We define

$$\begin{aligned}
G_{2,1} = & -\gamma_{j_0}(x) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - (-\gamma_{j_0}(\widetilde{x})) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) \\
G_{2,2} = & -\gamma_{j_0}(\widetilde{x}) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - (-\gamma_{j_0}(\widetilde{x})) \cdot c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) \\
G_{2,3} = & -\gamma_{j_0}(\widetilde{x}) \cdot c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - (-\gamma_{j_0}(\widetilde{x})) \cdot c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(\widetilde{x})_{j_0} \circ v)
\end{aligned}$$

We have

$$\begin{aligned}
\|G_{2,1}\| = & \|(-\gamma_{j_0}(x)) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - (-\gamma_{j_0}(\widetilde{x})) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)\| \\
\leq & |\gamma_{j_0}(x) - \gamma_{j_0}(\widetilde{x})| \cdot |c(x,:)_{j_0,i_0}| \cdot \|\mathrm{diag}(f(x)_{j_0} \circ v)\| \\
\leq & 2R^4 \cdot \|\gamma_{j_0}(x) - \gamma_{j_0}(\widetilde{x})\| \\
\leq & 2R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2,
\end{aligned}$$

where the first step is because of definition of $G_{2,1}$, the second step is due to Fact A.3, the third step follows from Lemma E.4, and the fourth step is because of Lemma E.5.

Additionally, we have

$$\begin{aligned}
\|G_{2,2}\| = & \| -\gamma_{j_0}(\widetilde{x}) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - (-\gamma_{j_0}(\widetilde{x})) \cdot c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)\| \\
\leq & \|\gamma_{j_0}(\widetilde{x}) \cdot \mathrm{diag}(f(\widetilde{x})_{j_0} \circ v)\| \cdot \|c(x,:)_{j_0,i_0} - c(\widetilde{x},:)_{j_0,i_0}\| \\
\leq & R^4 \cdot |c(x,:)_{j_0,i_0} - c(\widetilde{x},:)_{j_0,i_0}| \\
\leq & R^4 R_0 \cdot \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step is because of definition of $G_{2,2}$, the second step is due to Fact A.3, the third step follows from Lemma E.4, and the fourth step is because of Lemma E.5.

Additionally, we have

$$\begin{aligned}
\|G_{2,3}\| = & \| -\gamma_{j_0}(\widetilde{x}) \cdot c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) - (-\gamma_{j_0}(\widetilde{x})) \cdot c(\widetilde{x},:)_{j_0,i_0} \cdot \mathrm{diag}(f(\widetilde{x})_{j_0} \circ v)\| \\
\leq & \|\gamma_{j_0}(\widetilde{x})\| \cdot \|c(\widetilde{x},:)_{j_0,i_0}\| \cdot \|c(x,:)_{j_0,i_0} - c(\widetilde{x},:)_{j_0,i_0}\| \\
\leq & 2R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step is because of definition of $G_{2,3}$, the second step is due to Fact A.3, the third step follows from Lemma E.4 and Lemma E.5.

Combining all the above equations finish the proof. □

E.7 CALCULATION: STEP 3 LIPSCHITZ FOR MATRIX FUNCTION $-2\gamma_{j_0}(x) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$

In this section, we introduce our calculation of Lipschitz for $-2\gamma_{j_0}(x) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$.

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

- *Let* $G_3(x) = -2\gamma_{j_0}(x) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$.

- *Let R_0 be defined in Definition E.6.*
- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,:)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*

Then, we have

$$\|G_3(x) - G_3(\widetilde{x})\| \leq 10R^4 \cdot R_0 \|x - \widetilde{x}\|_2$$

Proof. We define

$$\begin{aligned}
G_{3,1} = & -2\gamma_{j_0}(x) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top - (-2\gamma_{j_0}(\widetilde{x}) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top) \\
G_{3,2} = & -2\gamma_{j_0}(\widetilde{x}) \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top - (-2\gamma_{j_0}(\widetilde{x}) \cdot (f(\widetilde{x})_{j_0} \circ v) f(x)_{j_0}^\top) \\
G_{3,3} = & -2\gamma_{j_0}(\widetilde{x}) \cdot (f(\widetilde{x})_{j_0} \circ v) f(x)_{j_0}^\top - (-2\gamma_{j_0}(\widetilde{x}) \cdot (f(\widetilde{x})_{j_0} \circ v) f(\widetilde{x})_{j_0}^\top)
\end{aligned}$$

For $G_{3,1}$, we have

$$\begin{aligned}
\|G_{3,1}\| \leq & 2 \cdot |\gamma(x)_{j_0} - \gamma(\widetilde{x})_{j_0}| \cdot \|f(x)_{j_0} \circ v\|_2 \cdot \|f(x)_{j_0}\|_2 \\
\leq & 2R_0 \cdot R^2 \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step is based on Fact A.3 and the second step is due to Lemma E.4 and Lemma E.5.

Similarly, we have

$$\|G_{3,2}\| \leq 2R_0 \cdot R^4 \|x - \widetilde{x}\|_2$$

and

$$\|G_{3,3}\| \leq 2R_0 \cdot R^4 \|x - \widetilde{x}\|_2$$

□

E.8 CALCULATION: STEP 4 LIPSCHITZ FOR MATRIX FUNCTION $-c(x,:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$

In this section, we introduce our calculation of Lipschitz for $-c(x,:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$.

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

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,:)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let $G_4(x) = -c(x,:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top$*

Then, we have

$$\|G_4(x) - G_4(\widetilde{x})\| \leq 10R^4 \cdot R_0 \|x - \widetilde{x}\|_2$$

Proof. We define

$$\begin{aligned}
G_{4,1} = & - c(x,:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top - (-c(\widetilde{x},:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top) \\
G_{4,2} = & - c(\widetilde{x},:)_{j_0,i_0} \cdot (f(x)_{j_0} \circ v) f(x)_{j_0}^\top - (-c(\widetilde{x},:)_{j_0,i_0} \cdot (f(\widetilde{x})_{j_0} \circ v) f(x)_{j_0}^\top) \\
G_{4,3} = & - c(\widetilde{x},:)_{j_0,i_0} \cdot (f(\widetilde{x})_{j_0} \circ v) f(x)_{j_0}^\top - (-c(\widetilde{x},:)_{j_0,i_0} \cdot (f(\widetilde{x})_{j_0} \circ v) f(\widetilde{x})_{j_0}^\top)
\end{aligned}$$

For $G_{4,1}$, we have

$$\|G_{4,1}\| \leq R^2 \cdot R_0 \cdot \|x - \widetilde{x}\|_2$$

For $G_{4,2}$, we have

$$\|G_{4,2}\| \leq 2R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2$$

For $G_{4,3}$, we have

$$\|G_{4,3}\| \leq 2R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2$$

□

E.9 CALCULATION: STEP 5 LIPSCHITZ FOR MATRIX FUNCTION $-2\gamma_{j_0}(x) \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$

In this section, we introduce our calculation of Lipschitz for $-2\gamma_{j_0}(x) \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$.

Lemma E.11. *If the following conditions hold*

- *Let R_0 be defined as Definition E.6*
- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,:)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let $G_5(x) = -2\gamma_{j_0}(x) \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$*

Then, we have

$$\|G_5(x) - G_5(\widetilde{x})\| \leq 10R^4 \cdot R_0 \|x - \widetilde{x}\|_2$$

Proof. This proof is similar to the proof of Lemma E.9, so we omit it here. □

E.10 CALCULATION: STEP 6 LIPSCHITZ FOR MATRIX FUNCTION $-c(x,:)_{j_0,i_0} \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$

In this section, we introduce our calculation of Lipschitz for $-c(x,:)_{j_0,i_0} \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$.

Lemma E.12. *If the following conditions hold*

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,:)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*

- *Let* $G_6(x) = -c(x,:)_{j_0,i_0} \cdot f(x)_{j_0}(f(x)_{j_0} \circ v)^\top$

Then, we have

$$\|G_5(x) - G_5(\widetilde{x})\| \leq 10R^4 \cdot R_0 \|x - \widetilde{x}\|_2$$

Proof. This proof is similar to the proof of Lemma E.10, so we omit it here. □

E.11 CALCULATION: STEP 7 LIPSCHITZ FOR MATRIX FUNCTION $2\gamma_{j_0}(x)c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top$

In this section, we introduce our calculation of Lipschitz for $2\gamma_{j_0}(x)c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top$.

Lemma E.13. *If the following conditions hold*

- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,:)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$
- $\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$
- *Let* $R \geq 4$
- *Let* $G_7(x) = 2\gamma_{j_0}(x)c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top$

Then, we have

$$\|G_7(x) - G_7(\widetilde{x})\| \leq 10R^4 R_0 \|x - \widetilde{x}\|_2$$

Proof. We define

$$\begin{aligned}
G_{7,1} &= 2\gamma_{j_0}(x)c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top - 2\gamma_{j_0}(\widetilde{x})c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top \\
G_{7,2} &= 2\gamma_{j_0}(\widetilde{x})c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top - 2\gamma_{j_0}(\widetilde{x})c(\widetilde{x},:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top \\
G_{7,3} &= 2\gamma_{j_0}(\widetilde{x})c(\widetilde{x},:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top - 2\gamma_{j_0}(\widetilde{x})c(\widetilde{x},:)_{j_0,i_0} \cdot f(\widetilde{x})_{j_0} f(x)_{j_0}^\top \\
G_{7,4} &= 2\gamma_{j_0}(\widetilde{x})c(\widetilde{x},:)_{j_0,i_0} \cdot f(\widetilde{x})_{j_0} f(x)_{j_0}^\top - 2\gamma_{j_0}(\widetilde{x})c(\widetilde{x},:)_{j_0,i_0} \cdot f(\widetilde{x})_{j_0} f(\widetilde{x})_{j_0}^\top
\end{aligned}$$

For $G_{7,1}$, we have

$$\begin{aligned}
\|G_{7,1}\| &= \|2\gamma_{j_0}(x)c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top - 2\gamma_{j_0}(\widetilde{x})c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top\| \\
&\leq 2|\gamma_{j_0}(x) - \gamma_{j_0}(\widetilde{x})| \|c(x,:)_{j_0,i_0} \cdot f(x)_{j_0} f(x)_{j_0}^\top\| \\
&\leq 2R_0 \cdot |c(x,:)_{j_0,i_0}| \cdot \|f(x)_{j_0}\| \cdot \|f(x)_{j_0}^\top\| \|x - \widetilde{x}\|_2 \\
&\leq 2R_0 \cdot 2R^2 \cdot \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step is due to the definition of $G_{7,1}$, the second step is because of Fact A.3, the third step is based on **Part 4** of Lemma E.5 and Fact A.3, and the last step comes from **Part 4 and Part 6** of Lemma E.4.

Similarly, for $G_{7,2}$, we have

$$\|G_{7,2}\| \leq 2R_0 \cdot R^2 \cdot \|x - \widetilde{x}\|_2$$

For $G_{7,3}$, we have

$$\|G_{7,3}\| \leq 2R_0 \cdot 2R^4 \cdot \|x - \widetilde{x}\|_2$$

For $G_{7,4}$, we have

$$\|G_{7,4}\| \leq 2R_0 \cdot 2R^4 \cdot \|x - \widetilde{x}\|_2$$

□

E.12 CALCULATION: STEP 8 LIPSCHITZ FOR MATRIX FUNCTION $\gamma_{j_0}(x)^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top$

In this section, we introduce our calculation of Lipschitz for $\gamma_{j_0}(x)^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top$.

Lemma E.14. *If the following conditions hold*

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,:)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let $G_{8,1} = \gamma_{j_0}(x)^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top$*

Then, we have

$$\|G_8(x) - G_8(\widetilde{x})\| \leq 10 R^4 R_0 \|x - \widetilde{x}\|_2$$

Proof. We define

$$\begin{aligned}
G_{8,1} &= \gamma_{j_0}(x)\gamma_{j_0}(x) \cdot f(x)_{j_0} f(x)_{j_0}^\top - \gamma_{j_0}(\widetilde{x})\gamma_{j_0}(x) \cdot f(x)_{j_0} f(x)_{j_0}^\top \\
G_{8,2} &= \gamma_{j_0}(\widetilde{x})\gamma_{j_0}(x) \cdot f(x)_{j_0} f(x)_{j_0}^\top - \gamma_{j_0}(\widetilde{x})^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top \\
G_{8,3} &= \gamma_{j_0}(\widetilde{x})^2 \cdot f(x)_{j_0} f(x)_{j_0}^\top - \gamma_{j_0}(\widetilde{x})^2 \cdot f(\widetilde{x})_{j_0} f(x)_{j_0}^\top \\
G_{8,4} &= \gamma_{j_0}(\widetilde{x})^2 \cdot f(\widetilde{x})_{j_0} f(x)_{j_0}^\top - \gamma_{j_0}(\widetilde{x})^2 \cdot f(\widetilde{x})_{j_0} f(\widetilde{x})_{j_0}^\top
\end{aligned}$$

We can show that

$$\max_{i \in [4]} \|G_{8,i}\| \leq R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2$$

□

E.13 CALCULATION: STEP 9 LIPSCHITZ FOR MATRIX FUNCTION $(f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$

In this section, we introduce our calculation of Lipschitz for $(f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$.

Lemma E.15. *If the following conditions hold*

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,:)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let $G_9(x) = (f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$*

Then, we have

$$\|G_9(x) - G_9(\widetilde{x})\| \leq 10 R^4 R_0 \|x - \widetilde{x}\|_2$$

Proof. We define

$$G_{9,1} = (f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top - (f(\widetilde{x})_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$$
$$G_{9,2} = (f(\widetilde{x})_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top - (f(\widetilde{x})_{j_0} \circ v) \cdot (f(\widetilde{x})_{j_0} \circ v)^\top$$

We can show that

$$\max_{i \in [2]} \|G_{9,i}\| \leq R^4 \cdot R_0 \cdot \|x - \widetilde{x}\|_2$$

□

F HESSIAN FOR X IS PSD

In Section F.1, we present the main result of PSD bound for Hessian. In Section F.2, we show the PSD bound for $B(x)$. In this section, our focus will be on establishing the PSD bound for $H_{x,x}$. Throughout this section, we will use the symbol H to represent $H_{x,x}$ for the sake of simplicity.

F.1 MAIN RESULT

In this section, we introduce the main result of the PSD bound for Hessian.

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

- *Let* $j_0 \in [n]$
- *Let* $i_0 \in [d]$
- *Let* $H_{j_0,i_0} = \frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}x} \in \mathbb{R}^{d^2 \times d^2}$
- *Let* $B_{j_0,i_0}(x) \in \mathbb{R}^{n \times n}$ *be defined as Definition D.3.*
  - *Therefore,* $H_{j_0,i_0} = \mathsf{A}_{j_0}^\top B_{j_0,i_0}(x) \mathsf{A}_{j_0} \in \mathbb{R}^{d^2 \times d^2}$
- *Let* $\max_{j_0 \in [n]} \| \mathsf{A}_{j_0} \| \leq R$
- *Let* $\sigma_{\min}$ *be the smallest singular value. We define* $\sigma_{\min}(\mathsf{A}_{\min}) := \min_{j_0 \in [n]} \sigma_{\min}(\mathsf{A}_{j_0})$.
- *Let* $H = \sum_{j_0=1}^n \sum_{i_0=1}^d H_{j_0,i_0}$
- *Let* $H_{\mathrm{reg},j_0,i_0} = \mathsf{A}_{j_0}^\top (B_{j_0,i_0}(x) + W^2) \mathsf{A}_{j_0}$ *where* $W \in \mathbb{R}^{n \times n}$ *is a positive diagonal matrix.*
- *Let* $H_{\mathrm{reg}} = \sum_{j_0=1}^n \sum_{i_0=1}^d H_{\mathrm{reg},j_0,i_0}$
- *Let* $C_0 := 30R^8$ *(be a local parameter in this lemma)*
- *Let* $l > 0$ *(denote the strongly convex parameter for hessian)*

Then, we have

- **Part 1.** *For each* $j_0 \in [n]$*, for each* $i_0 \in [d]$

$$-C_0 I_n \preceq B_{j_0,i_0}(x) \preceq C_0 I_n$$

- **Part 2.** *For each* $j_0 \in [n]$*, for each* $i_0 \in [d]$

$$\|H_{j_0,i_0}(x)\| \leq C_0 R^2.$$

- **Part 3.** *For each* $j_0 \in [n]$*,* $i_0 \in [d]$*, if* $\min_{j_1 \in [n]} w_{j_1,j_1} \geq \frac{l}{\sigma_{\min}(\mathsf{A}_{j_0})^2} + C_0$*, then we have*

$$H_{\mathrm{reg},j_0,i_0}(x) \succeq l \cdot I_{d^2}$$

- **Part 4.** *For each $j_0 \in [n]$, $i_0 \in [d]$, if $\min_{j_1 \in [n]} w_{j_1,j_1} \geq \frac{l}{\sigma_{\min}(\mathsf{A}_{j_0})^2} + 100 \cdot C_0$, then we have*

$$1.1 \cdot (B(x)_{j_0,i_0} + W^2) \succeq W^2 \succeq 0.9 \cdot (B(x)_{j_0,i_0} + W^2)$$

and

$$1.1 H_{j_0,i_0} \succeq H_{\mathrm{reg},j_0,i_0} \succeq 0.9 H_{j_0,i_0}$$

- **Part 5.** *For each $j_0 \in [n]$, $i_0 \in [d]$, if $\min_{j_1 \in [n]} w_{j_1,j_1} \geq \frac{l}{nd\sigma_{\min}(\mathsf{A}_{\min})^2} + C_0$, then we have*

$$H_{\mathrm{reg}}(x) \succeq l \cdot I_{d^2}$$

- **Part 6.** *For each $j_0 \in [n]$, $i_0 \in [d]$, if $\min_{j_1 \in [n]} w_{j_1,j_1} \geq \frac{l}{nd\sigma_{\min}(\mathsf{A}_{\min})^2} + 100 \cdot C_0$, then we have*

$$1.1 H \succeq H_{\mathrm{reg}} \succeq 0.9 H$$

Proof. **Proof of Part 1.**

It directly follows from Lemma F.2.

Proof of Part 2. We have

$$\begin{aligned} \|H_{j_0,i_0}\| &= \| \mathsf{A}_{j_0}^\top B_{j_0,i_0}(x) \mathsf{A}_{j_0} \| \\ &\leq \| \mathsf{A}_{j_0} \|^2 \cdot \|B_{j_0,i_0}(x)\| \\ &\leq R^2 \cdot \|B_{j_0,i_0}(x)\| \\ &\leq 30 R^{10} \end{aligned}$$

where the first step follows from the $H_{j_0,i_0} = \mathsf{A}_{j_0}^\top B_{j_0,i_0}(x) \mathsf{A}_{j_0}$, the second step follows from Fact A.3, the third step follows from $\max_{j_0 \in [n]} \| \mathsf{A}_{j_0} \| \leq R$, and the last step follow from **Part 1**.

Proof of Part 3.

The proof is similar to Deng et al. [2023a].

Proof of Part 4.

The proof is similar to Deng et al. [2023a].

Proof of Part 5 and Part 6. It is because we can write H as summation of nd terms H_{j_0,i_0} for all $j_0 \in [d]$, $i_0 \in [d]$. □

F.2 PSD BOUND

In this section, we analyze the PSD bound for each of the B_{rank} and B_{diag}.

Lemma F.2. *If the following condition holds*

- $B_{\mathrm{diag}}^1 := (1 - \gamma_{j_0}(x)) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v)$
- $B_{\mathrm{rank}}^1 := -(2\gamma_{j_0}(x) + c(x,:)_{j_0,i_0}) \cdot ((f(x)_{j_0} \circ v) f(x)_{j_0}^\top + f(x)_{j_0} (f(x)_{j_0} \circ v)^\top)$
- $B_{\mathrm{rank}}^2 := (2\gamma_{j_0}(x) c(x,:)_{j_0,i_0} + \gamma_{j_0}(x)^2) \cdot f(x)_{j_0} f(x)_{j_0}^\top$
- $B_{\mathrm{rank}}^3 := (f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top$
- $|\gamma(x)_{j_0}| \leq R^2$
- $|c(x,:)_{j_0,i_0}| \leq 2R^2$
- $\|v\|_2 \leq R^2$

Then, we have

- *Part 1.*

$$-8R^6 \cdot I_n \preceq B^1_{\mathrm{diag}} \preceq 8R^6 \cdot I_n$$

- *Part 2.*

$$-16R^8 \cdot I_n \preceq B^1_{\mathrm{rank}} \preceq 16R^8 \cdot I_n$$

- *Part 3.*

$$-8R^4 \cdot I_n \preceq B^2_{\mathrm{rank}} \preceq 8R^4 \cdot I_n$$

- *Part 4.*

$$0 \cdot I_n \preceq B^3_{\mathrm{rank}} \preceq 8R^4 \cdot I_n$$

Proof. **Proof of Part 1.**

$$\begin{aligned} B^1_{\mathrm{diag}} &= (1 - \gamma_{j_0}(x)) \cdot c(x,:)_{j_0,i_0} \cdot \mathrm{diag}(f(x)_{j_0} \circ v) \\ &\preceq |1 - \gamma_{j_0}(x)||c(x,:)_{j_0,i_0}|\|f(x)_{j_0}\|_2 \|v\|_2 \\ &\preceq 8R^6 \cdot I_n \end{aligned}$$

where the first step follows from the definition of B^1_{diag}, the second step follows from Fact A.4, and the last step follows from Lemma E.4, $|\gamma(x)_{j_0}| \leq R^2,\ |c(x,:)_{j_0,i_0}| \leq 2R^2$, and $\|v\|_2 \leq R^2$.

Proof of Part 2.

$$\begin{aligned} B^1_{\mathrm{rank}} =& -(2\gamma_{j_0}(x) + c(x,:)_{j_0,i_0}) \cdot ((f(x)_{j_0} \circ v) f(x)_{j_0}^\top + f(x)_{j_0}(f(x)_{j_0} \circ v)^\top) \\ \succeq& -|2\gamma_{j_0}(x) + c(x,:)_{j_0,i_0}| \cdot ((f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top + f(x)_{j_0} f(x)_{j_0}^\top) \\ \succeq& -4R^2 \cdot (\|f(x)_{j_0} \circ v\|_2^2 + \|f(x)_{j_0}\|_2^2) I_n \\ \succeq& -4R^2 (\|f(x)_{j_0}\|_2^2 \|v\|_2^2 + \|f(x)_{j_0}\|_2^2) I_n \\ \succeq& -5R^4 \cdot I_n \end{aligned}$$

where the first step follows from the definition of B^1_{rank}, the second step follows from Fact A.4, the third step follows from $|\gamma(x)_{j_0}| \leq R^2,\ |c(x,:)_{j_0,i_0}| \leq 2R^2$ and Fact A.4, the fourth step follows from Fact A.1, and last step follows from $\|f(x)_{j_0}\|_2 \leq 1$ (see **Part 4** of Lemma E.4) and $\|v\|_2 \leq R^2$.

Proof of Part 3.

$$\begin{aligned} B^2_{\mathrm{rank}} &= (2\gamma_{j_0}(x) c(x,:)_{j_0,i_0} + \gamma_{j_0}(x)^2) \cdot f(x)_{j_0} f(x)_{j_0}^\top \\ &\preceq |2\gamma_{j_0}(x) c(x,:)_{j_0,i_0} + \gamma_{j_0}(x)^2| \|f(x)_{j_0}\|_2^2 \\ &\preceq 8R^4 \cdot I_n \end{aligned}$$

where the first step follows from definition of B^2_{rank}, the second step follows from Fact A.4, and the last step follows from $|\gamma(x)_{j_0}| \leq R^2,\ |c(x,:)_{j_0,i_0}| \leq 2R^2$ and Lemma E.4.

Proof of Part 4.

$$\begin{aligned} B^3_{\mathrm{rank}} &= (f(x)_{j_0} \circ v) \cdot (f(x)_{j_0} \circ v)^\top \\ &\preceq \|f(x)_{j_0} \circ v\|_2^2 \\ &\preceq \|f(x)_{j_0}\|_2^2 \|v\|_2^2 \\ &\preceq 8R^4 \cdot I_n \end{aligned}$$

where the first step follows from definition of B^3_{rank}, the second step follows from Fact A.4, the third step follows from Fact A.1, and the last step follows from $\|v\|_2 \leq R^2$ and Lemma E.4.

□

G HESSIAN FOR Y

In Section G.1, we present the hessian property with respect to Y. In Section G.2, we compute the Hessian matrix with respect to Y for one j_0, i_0.

G.1 HESSIAN PROPERTY

In this section, we analyze the Hessian properties.

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

- *Let* $B_{j_0}(x) = f(x)_{j_0} f(x)_{j_0}^\top \in \mathbb{R}^{n \times n}$ *(because of Lemma G.2)*
- *Let* $B(x) = \sum_{j_0=1}^n B_{j_0}(x)$
- *Let* $H_{j_0,i_0} = \frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_0} \mathrm{d} y_{i_0}} = A_3^\top B_{j_0}(x) A_3 \in \mathbb{R}^{d \times d}$
- *Let* $H_{i_0} \in \mathbb{R}^{d \times d}$ *be* $H_{i_0} = \frac{\mathrm{d}^2 L}{\mathrm{d} y_{i_0} \mathrm{d} y_{i_0}} = \sum_{j_0=1}^d H_{j_0,i_0}$
- *Let* $H_{\mathrm{reg},i_0} = A_3^\top (B(x) + W^2) A_3$ *where* $W \in \mathbb{R}^{n \times n}$ *is a positive diagonal matrix*
- *Let* $H(y) \in \mathbb{R}^{d^2 \times d^2}$ *be* $H(y) = \begin{bmatrix} H_1 & 0 & \cdots & 0 \\ 0 & H_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H_d \end{bmatrix}$

Then, we have

- **Part 1.**
$$0 \preceq B_{j_0}(x) \preceq I_n$$
- **Part 2.**
$$0 \preceq B(x) \preceq n \cdot I_n$$
- **Part 3.** *If* $\min_{j_1 \in [n]} w_{j_1,j_1}^2 \geq \frac{l}{\sigma_{\min}(A_3)^2}$
$$H_{\mathrm{reg},i_0} \succeq l \cdot I_d, \quad H(y) \succeq l \cdot I_{d^2}$$
- **Part 4.** *If* $\min_{j_1 \in [n]} w_{j_1,j_1}^2 \geq \frac{l}{\sigma_{\min}(A_3)^2} + 100n$
$$0.9(W^2 + B(x)) \preceq W^2 \preceq 1.1(W^2 + B(x))$$
- **Part 5.** *Lipschitz, Due to* $H(y)$ *is independent of* y*, then*
$$\|H(y) - H(\widetilde{y})\| \leq \|y - \widetilde{y}\|_2$$

Proof. For hessian closed-form, we can obtain them from Lemma G.2.

The proofs are straightforward, so we omit the details here. □

G.2 HESSIAN FOR ONE j_0, i_0

In this section, we analyze the Hessian for the matrix Y with one j_0, i_0.

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

- *We define a temporary notation here $v := f(x)_{j_0}$ (for simplicity we drop the index j_0 in the statement. Note that v could have different meaning in other sections.)*
- *Let $f(x)_{j_0}$ be defined as Definition A.10.*
- *Let $c(x,:)_{j_0,i_0}$ be defined as Definition A.12.*
- *Let $h(y)_{i_0}$ be defined as Definition A.11.*
- *Let L_{j_0,i_0} be defined as Definition A.10.*

Then, we have

- **Part 1.** *For $i_1 = i_2$, the diagonal case*

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0,i_1}\mathrm{d}y_{i_0,i_1}} = A_{3,*,i_1}^\top v v^\top A_{3,*,i_1}$$

- **Part 2.** *For $i_1 \neq i_2$, the off-diagonal case*

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0,i_1}\mathrm{d}y_{i_0,i_2}} = A_{3,*,i_1}^\top v v^\top A_{3,*,i_2}$$

- **Part 3.** *The $\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0}\mathrm{d}y_{i_0}} \in \mathbb{R}^{d\times d}$*

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0}\mathrm{d}y_{i_0}} = A_3^\top v v^\top A_3$$

Proof. **Proof of Part 1.**

$$\begin{aligned}
\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0,i_1}\mathrm{d}y_{i_0,i_1}} &= \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(\frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}} L_{j_0,i_0}) \\
&= \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(c(:,y)_{j_0,i_0}\langle v, A_{3,*,i_1}\rangle) \\
&= \langle v, A_{3,*,i_1}\rangle \cdot \langle v, A_{3,*,i_1}\rangle \\
&= A_{3,*,i_1}^\top v v^\top A_{3,*,i_1}
\end{aligned}$$

where the first step follows from simple algebra, the second step follows from Lemma B.2, the third step follows from Lemma B.2, and the last step follows from Fact A.1.

Proof of Part 2.

$$\begin{aligned}
\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}y_{i_0,i_2}\mathrm{d}y_{i_0,i_1}} &= \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_2}}(\frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}} L_{j_0,i_0}) \\
&= \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_2}}(c(:,y)_{j_0,i_0}\langle v, A_{3,*,i_1}\rangle) \\
&= \langle v, A_{3,*,i_2}\rangle \cdot \langle v, A_{3,*,i_1}\rangle \\
&= A_{3,*,i_1}^\top v v^\top A_{3,*,i_2}
\end{aligned}$$

where the first step follows from simple algebra, the second step follows from Lemma B.2, the third step follows from Lemma B.2, and the last step follows from Fact A.1.

Proof of Part 3.

It follows by combining above two parts directly. □

H HESSIAN FOR X AND Y

In Section H.1, we compute the Hessian matrix with respect to both X and Y. In Section H.2, we present several helpful lemmas for the following proof. In Section H.3, we create $B(x)$ for the further analysis.

H.1 COMPUTING HESSIAN

In this section, we compute the Hessian matrix for X and Y.

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

- *Let $f(x)_{j_0}$ be defined as Definition A.10.*
- *Let $c(x,y)_{j_0,i_0}$ be defined as Definition A.12.*
- *Let $h(y)_{i_0}$ be defined as Definition A.11.*
- *Let L_{j_0,i_0} be defined as Definition A.7.*

Then, we have

- *Part 1.*

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(\frac{\mathrm{d}}{\mathrm{d}x_i}L_{j_0,i_0}) = & \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0}\rangle \\ & - \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \langle f(x)_{j_0}, h(y)_{i_0}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\ & + c(x,y)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, A_{3,*,i_1}\rangle - \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)\end{aligned}$$

Proof. We can show

$$\begin{aligned}& \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(\frac{\mathrm{d}}{\mathrm{d}x_i}L_{j_0,i_0}) \\ = & \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(c(x,y)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0}\rangle - \langle f(x)_{j_0}, h(y)_{i_0}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)) \\ = & \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(c(x,y)_{j_0,i_0}) \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0}\rangle - \langle f(x)_{j_0}, h(y)_{i_0}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \\ & + (c(x,y)_{j_0,i_0}) \cdot \frac{\mathrm{d}}{\mathrm{d}y_{i_0,i_1}}(\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0}\rangle - \langle f(x)_{j_0}, h(y)_{i_0}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \\ = & \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0}\rangle - \langle f(x)_{j_0}, h(y)_{i_0}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \\ & + c(x,y)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, A_{3,*,i_1}\rangle - \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle) \\ = & \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0}\rangle \\ & - \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \langle f(x)_{j_0}, h(y)_{i_0}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle \\ & + c(x,y)_{j_0,i_0} \cdot (\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, A_{3,*,i_1}\rangle - \langle f(x)_{j_0}, A_{3,*,i_1}\rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i}\rangle)\end{aligned}$$

where the first step is due to **Part 6** of Lemma B.1, the second step comes from the product rule of derivative, the third step is based on Lemma G.2, and the last step follows from simple algebra.

Thus, we complete the proof. □

H.2 A HELPFUL LEMMA

In this section, we provide a helpful Lemma.

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

- *Let $f(x)_{j_0}$ be defined in Definition A.10.*

- *Let* $\mathsf{A} \in \mathbb{R}^{n^2 \times d^2}$ *be defined in Definition A.8.*
- *Let* $c(x,y)_{j_0,i_0}$ *be defined as Definition A.12.*
- *Let* $h(y)_{i_0}$ *be defined as Definition A.11.*
- *Let* L_{j_0,i_0} *be defined as Definition A.7.*

Then, we have

- **Part 1.**

$$\langle f(x)_{j_0}, A_{3,*,i_1} \rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0} \rangle = \mathsf{A}_{j_0,i}^\top (f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top A_{3,*,i_1}$$

- **Part 2.**

$$\langle f(x)_{j_0}, A_{3,*,i_1} \rangle \cdot \langle f(x)_{j_0}, h(y)_{i_0} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle = \langle f(x)_{j_0}, h(y)_{i_0} \rangle \cdot \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top A_{3,*,i_1}$$

- **Part 3.**

$$\langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}^\top, A_{3,*,i_1} \rangle = \mathsf{A}_{j_0,i}^\top \operatorname{diag}(f(x)_{j_0}) A_{3,*,i_1}$$

- **Part 4.**

$$\langle f(x)_{j_0}, A_{3,*,i_1} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle = \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top A_{3,*,i_1}$$

Proof. **Proof of Part 1.**

$$\begin{aligned} \langle f(x)_{j_0}, A_{3,*,i_1} \rangle \cdot \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, h(y)_{i_0} \rangle &= \langle f(x)_{j_0} \circ h(y)_{i_0}, \mathsf{A}_{j_0,i} \rangle f(x)_{j_0}^\top A_{3,*,i_1} \\ &= \mathsf{A}_{j_0,i}^\top (f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top A_{3,*,i_1} \end{aligned}$$

where the first step follows from Fact A.1, and the second step follows from Fact A.1.

Proof of Part 2.

$$\langle f(x)_{j_0}, A_{3,*,i_1} \rangle \cdot \langle f(x)_{j_0}, h(y)_{i_0} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle = \langle f(x)_{j_0}, h(y)_{i_0} \rangle \, \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top A_{3,*,i_1}$$

where the first step follows from Fact A.1.

Proof of Part 3.

$$\begin{aligned} \langle f(x)_{j_0} \circ \mathsf{A}_{j_0,i}, A_{3,*,i_1} \rangle &= (f(x)_{j_0} \circ \mathsf{A}_{j_0,i})^\top A_{3,*,i_1} \\ &= (\operatorname{diag}(f(x)_{j_0}) \, \mathsf{A}_{j_0,i})^\top A_{3,*,i_1} \\ &= \mathsf{A}_{j_0,i}^\top \operatorname{diag}(f(x)_{j_0}) A_{3,*,i_1} \end{aligned}$$

where the first, second, and last step follows from Fact A.1.

Proof of Part 4.

$$\langle f(x)_{j_0}, A_{3,*,i_1} \rangle \cdot \langle f(x)_{j_0}, \mathsf{A}_{j_0,i} \rangle = \mathsf{A}_{j_0,i}^\top f(x)_{j_0} f(x)_{j_0}^\top A_{3,*,i_1}$$

where the first step follows from Fact A.1.

□

H.3 CREATING $B(x,y)$

In this section, we give a formal definition of $B(x,y)$.

Definition H.3. *We define* $B(x,y)$

$$B(x,y) = B_{\mathrm{diag}}^1 + B_{\mathrm{rank}}^1 + B_{\mathrm{rank}}^2 + B_{\mathrm{rank}}^1$$

where

- $B_{\mathrm{rank}}^1(x,y) = (f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top$
- $B_{\mathrm{rank}}^2(x,y) = -\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$
- $B_{\mathrm{diag}}^1(x,y) = -c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$
- $B_{\mathrm{rank}}^3(x,y) = c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$

Lemma H.4. *If the following conditions*

- *Let* $B(x,y)$ *be defined as Definition H.3.*

Then, we have

- **Part 1.**

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_0} \mathrm{d} x} = \mathsf{A}_{j_0}^\top B(x,y) A_3 \in \mathbb{R}^{d^2 \times d}$$

- **Part 2.** $i_1 \neq i_0$

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_1} \mathrm{d} x} = \mathsf{A}_{j_0}^\top \mathbf{0}_{n \times n} A_3 \in \mathbb{R}^{d^2 \times d} = \mathbf{0}_{d^2 \times d}$$

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

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_0,i_2} \mathrm{d} x_i} = \mathsf{A}_{j_0,i}^\top B(x,y) A_{3,*,i_2}$$

where the first step follows from combining Lemma H.1 and Lemma H.2.

Then, we can have

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_0} \mathrm{d} x} = \mathsf{A}_{j_0}^\top B(x,y) A_3$$

Proof of Part 2. We have

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_1,i_2} \mathrm{d} x_i} = \mathsf{A}_{j_0,i}^\top \mathbf{0}_{n \times n} A_{3,*,i_2} = \mathbf{0}_{n \times n}$$

where the first step follows from combining Lemma H.1 and Lemma H.2.

Then, we can have

$$\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d} y_{i_1} \mathrm{d} x} = \mathsf{A}_{j_0}^\top \mathbf{0}_{n \times n} A_3 = \mathbf{0}_{n \times n}$$

□

I LIPSCHITZ FOR HESSIAN OF x, y

In Section I.1, we present the main results of the Lipschitz property of $H_{x,y}$. In Section I.2, we summarize the results from the following steps 1-4. In Section I.3, we compute the upper bound of basic functions for the following proof. In Section I.4, we compute the Lipschitz Property of basic functions for the following proof. In Section I.5, we analyze the first step of Lipschitz function $(f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top$. In Section I.6, we analyze the second step of Lipschitz function $-\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$. In Section I.7, we analyze the third step of Lipschitz function $-c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$. In Section I.8, we analyze the fourth step of Lipschitz function $c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$. In Section I.9, we compute the PSD upper bound for the Hessian matrix. In Section I.10, we summarize PSD upper bound of $G(x,y)$.

I.1 MAIN RESULTS

In this section, we present the main result of Section I.

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

- $\max_{j_0 \in [n]} \| \mathsf{A}_{j_0} \| \leq R$
- *Let* $H(x,y)_{j_0,i_0} \in \mathbb{R}^{d^2 \times d}$ *denote* $\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}y_{i_0}}$
- $\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}y_{i_1}} = \mathbf{0}_{d^2 \times d}$
- *Let* $H(x,y) \in \mathbb{R}^{d^2 \times d^2}$ *be*

$$H(x,y) := \begin{bmatrix} \sum_{j_0=1}^n H_{j_0,1}(x,y) & \sum_{j_0=1}^n H_{j_0,2}(x,y) & \cdots & \sum_{j_0=1}^n H_{j_0,d}(x,y) \end{bmatrix}$$

Then we have

- *Part 1. For* $j_0 \in [d], i_0 \in [n]$

$$\|H(x,y)_{j_0,i_0} - H(\widetilde{x},\widetilde{y})_{j_0,i_0}\| \leq n^{1.5} \exp(20R^2) \cdot (\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

- *Part 2.*

$$\|H(x,y) - H(\widetilde{x},\widetilde{y})\| \leq n^{2.5} d \exp(20R^2)(\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

Proof. **Proof of Part 1.** It follows from Lemma I.2.

Proof of Part 2. We can show that

$$\|H(x,y) - H(\widetilde{x},\widetilde{y})\| \leq nd \cdot n^{1.5} \exp(20R^2)(\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

where the first step follows from that we can write H as summation of nd terms H_{j_0,i_0} for all $j_0 \in [d]$, $i_0 \in [d]$. □

I.2 SUMMARY OF FOUR STEPS ON LIPSCHITZ FOR MATRIX FUNCTIONS

In this section, we summarize the four steps for analyzing the Lipschitz for different matrix functions.

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

- $G_1(x,y) = (f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top$
- $G_2(x,y) = -\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$
- $G_3(x,y) = -c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$
- $G_4(x,y) = c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$

Then, we have

$$\sum_{k=1}^4 \|G_k(x,y) - G_k(\widetilde{x},\widetilde{y})\| \leq n^{1.5} \exp(20R^2)(\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

Proof. The proof follows from Lemma I.5, Lemma I.6, Lemma I.7, and Lemma I.8. □

I.3 A CORE TOOL: UPPER BOUND FOR SEVERAL BASIC FUNCTIONS

In this section, we give an upper bound for each of the basic functions.

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

- *Let* $f(y)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10.*
- *Let* $h(y)_{i_0} \in \mathbb{R}^n$ *be defined as Definition A.11.*
- *Let* $c(x,y)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12.*
- *Let* $R \geq 4$
- $\|A_3\| \leq R$
- $\|y_{i_0}\| \leq R$
- $\|b_{j_0,i_0}\|_2 \leq R$

Then, we have

- *Part 1.* $\|h(y)_{i_0}\|_2 \leq R^2$
- *Part 2.* $|c(x,y)_{j_0,i_0}| \leq 2R^2$

Proof. **Proof of Part 1.**

$$\begin{aligned} \|h(y)_{i_0}\|_2 &= \|A_3 y_{i_0}\|_2 \\ &\leq \|A_3\| \|y_{i_0}\|_2 \\ &\leq R^2 \end{aligned}$$

where the first step is due to Definition A.11, the second step is based on Fact A.3 and the third step is because of Lemma E.4.

Proof of Part 2.

$$\begin{aligned} |c(x,y)_{j_0,i_0}| &= |\langle f(x)_{j_0}, h(y)_{i_0} \rangle - b_{j_0,i_0}| \\ &\leq \|f(x)_{j_0}\|_2 \|h(y)_{i_0}\|_2 + |b_{j_0,i_0}| \\ &\leq R^2 + R \\ &\leq 2R^2 \end{aligned}$$

where the first step is because of Definition A.12, the second step is based on triangle inequality and Cauchy–Schwarz inequality, the third step is due to Lemma E.4, and the last step follows from $R \geq 4$. □

I.4 A CORE TOOL: LIPSCHITZ PROPERTY FOR SEVERAL BASIC FUNCTIONS

In this section, we introduce the Lipschitz property for several basic functions.

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

- *Let* $f(y)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10.*
- *Let* $h(y)_{i_0} \in \mathbb{R}^n$ *be defined as Definition A.11.*
- *Let* $c(x,y)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12.*
- *Let* $R \geq 4$
- $\|A_3\| \leq R$
- $\|y_{i_0}\| \leq R$
- $\|b_{j_0,i_0}\|_2 \leq R$
- *Let* R_0 *be defined as Definition E.6.*

Then, we have

- *Part 1.* $\|h(y)_{i_0} - h(\widetilde{y})_{i_0}\|_2 \leq R\|y - \widetilde{y}\|_2$
- *Part 2.* $|c(x,y)_{j_0,i_0} - c(\widetilde{x},y)_{j_0,i_0}| \leq R^2 \cdot R_0\|x - \widetilde{x}\|$
- *Part 3.* $|c(x,y)_{j_0,i_0} - c(x,\widetilde{y})_{j_0,i_0})| \leq R\|y - \widetilde{y}\|_2$

Proof. **Proof of Part 1.**

$$\begin{aligned}\|h(y)_{i_0} - h(\widetilde{y})_{i_0}\|_2 &= \|A_3 y_{i_0} - A_3\widetilde{y}_{i_0}\|_2 \\ &\leq \|A_3\|\|y_{i_0} - \widetilde{y}_{i_0}\|_2 \\ &\leq R\|y - \widetilde{y}\|_2\end{aligned}$$

where the first step follows from Definition A.11, the second step is based on Fact A.3, and the third step is due to Lemma E.4.

Proof of Part 2.

$$\begin{aligned}|c(x,y)_{j_0,i_0} - c(\widetilde{x},y_{j_0,i_0})| &= |\langle f(x)_{j_0}, h(y)_{i_0}\rangle - b_{j_0,i_0} - (\langle f(\widetilde{x})_{j_0}, h(y)_{i_0}\rangle - b_{j_0,i_0})| \\ &\leq \|f(x)_{j_0} - f(\widetilde{x})_{j_0}\|_2\|h(y)_{i_0}\|_2 \\ &\leq R^2 \cdot R_0\|x - \widetilde{x}\|_2\end{aligned}$$

where the first step is due to Definition A.12, the second step follows from Cauchy–Schwarz inequality, and the third step is because of **Part 1** of Lemma I.3 and **Part 3** of Lemma E.5.

Proof of Part 3.

$$\begin{aligned}|c(x,y)_{j_0,i_0} - c(x,\widetilde{y})_{j_0,i_0})| &= |\langle f(x)_{j_0}, h(y)_{i_0}\rangle - b_{j_0,i_0} - (\langle f(x)_{j_0}, h(\widetilde{y})_{i_0}\rangle - b_{j_0,i_0})| \\ &\leq \|f(x)_{j_0}\|_2 \cdot \|h(y)_{i_0} - h(\widetilde{y})_{i_0}\|_2 \\ &\leq R\|y - \widetilde{y}\|_2\end{aligned}$$

where the first step follows from Definition A.12, the second step is due to Cauchy–Schwarz inequality and the third step is because of **Part 4** of Lemma E.4 and **Part 1** of this Lemma. □

I.5 CALCULATION: STEP 1 LIPSCHITZ FOR MATRIX FUNCTION $(f(x)_{j_0} \circ h(y)_{i_0})f(x)_{j_0}^\top$

In this section, we calculate the Lipschitz for $(f(x)_{j_0} \circ h(y)_{i_0})f(x)_{j_0}^\top$.

Lemma I.5. *If the following conditions*

- *Let* $G_1(x,y) = (f(x)_{j_0} \circ h(y)_{i_0})f(x)_{j_0}^\top$
- *Let* R_0 *be defined in Definition E.6.*
- *Let* $\alpha(x)_{j_0} \in \mathbb{R}$ *be defined as Definition A.9*
- *Let* $f(x)_{j_0} \in \mathbb{R}^n$ *be defined as Definition A.10*
- *Let* $c(x,y)_{j_0,i_0} \in \mathbb{R}$ *be defined as Definition A.12*
- *Let* $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v\rangle \in \mathbb{R}$
- $\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\|\mathsf{A}_{j_0}\| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$
- *Let* $R \geq 4$

Then, we have

$$\|G_1(x,y) - G_1(\widetilde{x},\widetilde{y})\| \leq 2R^2 \cdot R_0(\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

Proof. We define

$$\begin{aligned}
G_{1,1} &= (f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top - (f(\widetilde{x})_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top \\
G_{1,2} &= (f(\widetilde{x})_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top - (f(\widetilde{x})_{j_0} \circ h(\widetilde{y})_{i_0}) f(x)_{j_0}^\top \\
G_{1,3} &= (f(\widetilde{x})_{j_0} \circ h(\widetilde{y})_{i_0}) f(x)_{j_0}^\top - (f(\widetilde{x})_{j_0} \circ h(\widetilde{y})_{i_0}) f(\widetilde{x})_{j_0}^\top
\end{aligned}$$

where the first step follows from definition of $G_{1,1}$, the second step is based on Fact A.2 and the third step is due to Lemma E.4.

We have

$$\begin{aligned}
\|G_{1,1}\| &= \|(f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top - (f(\widetilde{x})_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top\| \\
&\leq \|f(x)_{j_0} - f(\widetilde{x})_{j_0}\|_\infty \cdot \|h(y)_{i_0}\|_2 \cdot \|f(x)_{j_0}\|_2 \\
&\leq R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step follows from definition of $G_{1,1}$, the second step is due to Fact A.3, and the third step is based on combining Lemma E.4, Lemma E.5, and Lemma I.3.

Also, we have

$$\begin{aligned}
\|G_{1,2}\| &= \|(f(\widetilde{x})_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top - (f(\widetilde{x})_{j_0} \circ h(\widetilde{y})_{i_0}) f(x)_{j_0}^\top\| \\
&\leq \|f(\widetilde{x})_{j_0}\|_2 \cdot \|h(y)_{i_0} - h(\widetilde{y})_{i_0}\|_2 \cdot \|f(x)_{j_0}\|_2 \\
&\leq R \|y - \widetilde{y}\|_2
\end{aligned}$$

where the first step is based on definition of $G_{1,2}$, the second step is because of Fact A.3, and the third step follows from Lemma I.4.

Additionally,

$$\begin{aligned}
\|G_{1,3}\| &= \|(f(\widetilde{x})_{j_0} \circ h(\widetilde{y})_{i_0}) f(x)_{j_0}^\top - (f(\widetilde{x})_{j_0} \circ h(\widetilde{y})_{i_0}) f(\widetilde{x})_{j_0}^\top\| \\
&\leq \|f(\widetilde{x})_{j_0}\|_2 \cdot \|h(\widetilde{y})_{i_0}\|_2 \cdot \|f(x)_{j_0} - f(\widetilde{x})_{j_0}\|_2 \\
&\leq R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step follows from the definition of $G_{1,3}$, the second step follows from Fact A.3, and the third step is because of Lemma E.5.

Combining all the above equations we complete the proof. □

I.6 CALCULATION: STEP 2 LIPSCHITZ FOR MATRIX FUNCTION $-\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$

In this section, we calculate the Lipschitz for $-\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$.

Lemma I.6. *If the following conditions*

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,y)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$, $|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let $G_2(x,y) = -\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$*

Then, we have

$$\|G_2(x,y) - G_2(\widetilde{x},\widetilde{y})\| \leq 3R^2 R_0 (\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

Proof. We define

$$\begin{aligned}
G_{2,1} = & -\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top - (-\langle f(\widetilde{x})_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top) \\
G_{2,2} = & -\langle f(\widetilde{x})_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top - (-\langle f(\widetilde{x})_{j_0}, h(\widetilde{y})_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top) \\
G_{2,3} = & -\langle f(\widetilde{x})_{j_0}, h(\widetilde{y})_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top - (-\langle f(\widetilde{x})_{j_0}, h(\widetilde{y})_{i_0} \rangle f(\widetilde{x})_{j_0} f(x)_{j_0}^\top) \\
G_{2,4} = & -\langle f(\widetilde{x})_{j_0}, h(\widetilde{y})_{i_0} \rangle f(\widetilde{x})_{j_0} f(x)_{j_0}^\top - (-\langle f(\widetilde{x})_{j_0}, h(\widetilde{y})_{i_0} \rangle f(\widetilde{x})_{j_0} f(\widetilde{x})_{j_0}^\top)
\end{aligned}$$

We have

$$\begin{aligned}
\|G_{2,1}\| = & \| - \langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top - (-\langle f(\widetilde{x})_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top) \| \\
\leq & \|f(x)_{j_0} - f(\widetilde{x})_{j_0}\|_2 \cdot \|h(y)_{i_0}\|_2 \cdot \|f(x)_{j_0}\|_2 \cdot \|f(x)_{j_0}\|_2 \\
\leq & R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step is based on the definition of $G_{2,1}$, the second step follows from Fact A.1, and the third step is because of Lemma E.4.

and

$$\begin{aligned}
\|G_{2,2}\| = & \| - \langle f(\widetilde{x})_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top - (-\langle f(\widetilde{x})_{j_0}, h(\widetilde{y})_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top) \| \\
\leq & \|f(\widetilde{x})_{j_0}\|_2 \cdot \|h(y)_{i_0} - h(\widetilde{y})_{i_0}\| \cdot \|f(x)_{j_0}\|_2 \cdot \|f(x)_{j_0}\|_2 \\
\leq & R \|y - \widetilde{y}\|_2
\end{aligned}$$

where the first step is due to the definition of $G_{2,1}$, the second step is based on Fact A.1, and the third step follows from Lemma I.4.

Similarly, we have

$$\begin{aligned}
\|G_{2,3}\| \leq & R^2 \cdot R_0 \|x - \widetilde{x}\|_2 \\
\|G_{2,4}\| \leq & R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

Combining all the above equations we complete the proof. □

I.7 CALCULATION: STEP 3 LIPSCHITZ FOR MATRIX FUNCTION $-c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$

In this section, we calculate the Lipschitz for $-c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$.

Lemma I.7. *If the following conditions*

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,y)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let R_0 be defined as Definition E.6.*
- *Let $G_3(x,y) = -c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$*

Then, we have

$$\|G_3(x,y) - G_3(\widetilde{x},\widetilde{y})\| \leq 3R^2 \cdot R_0 (\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

Proof. We define

$$\begin{aligned}
G_{3,1} = & - c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0}) - (-c(\widetilde{x},y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})) \\
G_{3,2} = & - c(\widetilde{x},y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0}) - (-c(\widetilde{x},\widetilde{y})_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})) \\
G_{3,3} = & - c(\widetilde{x},\widetilde{y})_{j_0,i_0} \operatorname{diag}(f(x)_{j_0}) - (-c(\widetilde{x},\widetilde{y})_{j_0,i_0} \operatorname{diag}(f(\widetilde{x})_{j_0}))
\end{aligned}$$

For $G_{3,1}$, we have

$$\begin{aligned}
\|G_{3,1}\| = & \| - c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0}) - (-c(\widetilde{x},y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0}))\| \\
\leq & |c(x,y)_{j_0,i_0} - c(\widetilde{x},y)_{j_0,i_0}| \cdot \|f(x)_{j_0}\|_2 \\
\leq & R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

where the first step follows from definition of $G_{3,1}$, the second step is based on Fact A.2 and the third step is because of Lemma I.4.

Similarly, we have

$$\begin{aligned}
\|G_{3,2}\| \leq & R\|y - \widetilde{y}\|_2 \\
\|G_{3,3}\| \leq & 2R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

Combining all the above equations we complete the proof. □

I.8 CALCULATION: STEP 4 LIPSCHITZ FOR MATRIX FUNCTION $c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$

In this section, we calculate the Lipschitz for $c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$.

Lemma I.8. *If the following conditions*

- *Let $\alpha(x)_{j_0} \in \mathbb{R}$ be defined as Definition A.9*
- *Let $f(x)_{j_0} \in \mathbb{R}^n$ be defined as Definition A.10*
- *Let $c(x,y)_{j_0,i_0} \in \mathbb{R}$ be defined as Definition A.12*
- *Let $\gamma(x)_{j_0} = \langle f(x)_{j_0}, v \rangle \in \mathbb{R}$*
- *$\|A_1\|, \|A_2\|, \|A_3\| \leq R$, $\| \mathsf{A}_{j_0} \| \leq R$, $\|x\|_2 \leq R$,$|b_{j_0,i_0}| \leq R$, $\|v\|_2 \leq R^2$*
- *Let $R \geq 4$*
- *Let R_0 be defined in Definition E.6.*
- *Let $G_4(x,y) = c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$*

Then, we have

$$\|G_4(x,y) - G_4(\widetilde{x},\widetilde{y})\| \leq 5R^2 \cdot R_0 (\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

Proof. We define

$$\begin{aligned}
G_{4,1} = & c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top - c(\widetilde{x},y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top \\
G_{4,2} = & c(\widetilde{x},y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top - c(\widetilde{x},\widetilde{y})_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top \\
G_{4,3} = & c(\widetilde{x},\widetilde{y})_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top - c(\widetilde{x},\widetilde{y})_{j_0,i_0} f(\widetilde{x})_{j_0} f(x)_{j_0}^\top \\
G_{4,4} = & c(\widetilde{x},\widetilde{y})_{j_0,i_0} f(\widetilde{x})_{j_0} f(x)_{j_0}^\top - c(\widetilde{x},\widetilde{y})_{j_0,i_0} f(\widetilde{x})_{j_0} f(\widetilde{x})_{j_0}^\top
\end{aligned}$$

For $G_{4,1}$, we have

$$\begin{aligned}
\|G_{4,1}\| = & \|c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top - c(\widetilde{x},y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top\| \\
\leq & |c(x,y)_{j_0,i_0} - c(\widetilde{x},y)_{j_0,i_0}| \cdot \|f(x)_{j_0}\|_2 \cdot \|f(x)_{j_0}\|_2
\end{aligned}$$

$$\leq R^2 \cdot R_0 \|x - \widetilde{x}\|_2$$

where the first step is due to definition of $G_{4,1}$, the second step is because of Fact A.2 and the third step follows from Lemma E.4 and Lemma E.5.

Similarly, we have

$$\begin{aligned}
\|G_{4,2}\| &\leq R \|y - \widetilde{y}\|_2 \\
\|G_{4,3}\| &\leq 2R^2 \cdot R_0 \|x - \widetilde{x}\|_2 \\
\|G_{4,4}\| &\leq 2R^2 \cdot R_0 \|x - \widetilde{x}\|_2
\end{aligned}$$

Combining all the above equations we complete the proof. □

I.9 PSD UPPER BOUND FOR HESSIAN x, y

In this section, we analyze the PSD upper bound for Hessian.

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

- $\max_{j_0 \in [n]} \| \mathsf{A}_{j_0} \| \leq R$
- *Let* $H(x,y)_{j_0,i_0} \in \mathbb{R}^{d^2 \times d}$ *denote* $\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}y_{i_0}}$
- $\frac{\mathrm{d}^2 L_{j_0,i_0}}{\mathrm{d}x \mathrm{d}y_{i_1}} = \mathbf{0}_{d^2 \times d}$
- *Let* $H(x,y) \in \mathbb{R}^{d^2 \times d^2}$ *be*

$$H(x,y) := \begin{bmatrix} \sum_{j_0=1}^n H_{j_0,1}(x,y) & \sum_{j_0=1}^n H_{j_0,2}(x,y) & \cdots & \sum_{j_0=1}^n H_{j_0,d}(x,y) \end{bmatrix}$$

Then we have

- *Part 1. For* $j_0 \in [d], i_0 \in [n]$

$$\|H(x,y)_{j_0,i_0}\| \leq 10R^2$$

- *Part 2.*

$$\|H(x,y)\| \leq nd \cdot 10R^2$$

Proof. **Proof of Part 1.** It follows from Lemma I.10.

Proof of Part 2. We can show that

$$\begin{aligned}
\|H(x,y)\| &= \sum_{j_0=1}^d \sum_{i_0=1}^n \|H(x,y)_{j_0,i_0}\| \\
&\leq nd \cdot 10R^2
\end{aligned}$$

where the first step is due to the assumption of $H(x,y)$, and the second step comes from **Part 1.** □

I.10 UPPER BOUND ON HESSIAN SPECTRAL NORMS

In this section, we find the upper bound for the Hessian spectral norms.

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

- $G_1(x,y) = (f(x)_{j_0} \circ h(y)_{i_0}) f(x)_{j_0}^\top$
- $G_2(x,y) = -\langle f(x)_{j_0}, h(y)_{i_0} \rangle f(x)_{j_0} f(x)_{j_0}^\top$

- $G_3(x,y) = -c(x,y)_{j_0,i_0} \operatorname{diag}(f(x)_{j_0})$
- $G_4(x,y) = c(x,y)_{j_0,i_0} f(x)_{j_0} f(x)_{j_0}^\top$

Then, we have

- *Part 1.* $\|G_1(x,y)\| \leq R^2$
- *Part 2.* $\|G_2(x,y)\| \leq R^2$
- *Part 3.* $\|G_3(x,y)\| \leq 2R^2$
- *Part 4.* $\|G_4(x,y)\| \leq 2R^2$
- *Part 5.*

$$\sum_{k=1}^4 \|G_k(x,y)\| \leq 10R^2$$

Proof. The proof is straightforward by using upper bound on each term □

J GENERATING A SPECTRAL SPARSIFIER VIA TENSORSKETCH

Tensor type sketching has been widely used in problems Song et al. [2019], Diao et al. [2018, 2019], Ahle et al. [2020], Song et al. [2021a, 2024b, 2022], Zhang [2022], Song et al. [2023]. Section J.1 presents the definition of oblivious subspace embedding. In Section J.2, we give an overview of TensorSRHT and introduce its basic property. In Section J.3, we present the definition of the property of TensorSparse. In Section J.4, we introduce the fast approximation for hessian via sketching.

J.1 OBLIVIOUS SUBSPACE EMBEDDING

We define oblivious subspace embedding,

Definition J.1 (Oblivious subspace embedding, Sarlos [2006])**.** *We define (ϵ, δ, d, n)-Oblivious subspace embedding (OSE) as follows: Suppose Π is a distribution on $m \times n$ matrices S, where m is a function of n, d, ϵ, and δ. Suppose that with probability at least $1-\delta$, for any fixed $n \times d$ orthonormal basis U, a matrix S drawn from the distribution Π has the property that the singular values of SU lie in the range $[1-\epsilon, 1+\epsilon]$.*

J.2 TENSORSRHT

We define a well-known sketching matrix family called TensorSRHT Lu et al. [2013], Ahle et al. [2020]. It has been used in many optimization literature Song et al. [2021a, 2024b, 2022].

Definition J.2 (Tensor subsampled randomized Hadamard transform (TensorSRHT) Ahle et al. [2020], Song et al. [2021a])**.** *The TensorSRHT $S : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ is defined as*

$$S := \frac{1}{\sqrt{m}} P \cdot (HD_1 \otimes HD_2),$$

where each row of $P \in \{0,1\}^{m \times n^2}$ contains only one 1 at a random coordinate and one can view P as a sampling matrix. H is a $n \times n$ Hadamard matrix, and D_1, D_2 are two $n \times n$ independent diagonal matrices with diagonals that are each independently set to be a Rademacher random variable (uniform in $\{-1, 1\}$).

It is known Ahle et al. [2020] that TensorSRHT matrices imply the OSE.

Lemma J.3 (Ahle et al. [2020], Song et al. [2021a] , see for example, Lemma 2.12 in Song et al. [2021a])**.** *Let S be a TensorSRHT matrix defined in Definition J.2. If*

$$m = O(\epsilon^{-2} d^2 \log^3(nd/\epsilon\delta)),$$

then S is an $(\epsilon, \delta, d^2, n^2)$-OSE for degree-2 tensors.

Further for matrices $A_1, A_2 \in \mathbb{R}^{n \times d}$, $S(A_1 \otimes A_2)$ can be computed in $\widetilde{O}(nd + md^2)$ time.

J.3 TENSORSPARSE

Song et al. [2022] define TensorSparse by compose Sparse embedding Nelson and Nguyên [2013], Cohen [2016] with tensor operation Pagh [2013].

Definition J.4 (TensorSparse, see Definition 7.6 in Song et al. [2022])**.** *Let $h_1, h_2 : [n] \times [s] \to [m/s]$ be $O(\log 1/\delta)$-wise independent hash functions and let $\sigma_1, \sigma_2 : [n] \times [s] \to \{\pm 1\}$ be $O(\log 1/\delta)$-wise independent random sign functions. Then, the degree two tensor sparse transform, $S : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ is given as:*

$$R_{r,(i,j)} = \exists k \in [s] : \sigma_1(i,k)\sigma_2(j,k)/\sqrt{s} \cdot \mathbf{1}[((h_1(i,k) + h_2(j,k)) \textit{ mod } m/s) + (k-1)m/s = r]$$

Lemma J.5 (Theorem 7.10 in Song et al. [2022])**.** *Let $\epsilon \in (0,1)$ be precision parameter and $\delta \in (0,1)$ be success probability. Let $S \in \mathbb{R}^{m \times n^2}$ be a* TensorSparse *matrix (Def. J.4). Suppose $m = \Omega(\epsilon^{-2} d^2 \log(n/\delta))$ and $s = \epsilon^{-1} \log(n/\delta)$, then* TensorSparse *provides $(\epsilon, \delta, d^2, n^2)$-*OSE*.*

Further for matrices $A_1, A_2 \in \mathbb{R}^{n \times d}$, $S(A_1 \otimes A_2)$ can be computed in $O((\mathrm{nnz}(A_1) + \mathrm{nnz}(A_2))s + md^2)$ time

J.4 FAST APPROXIMATION FOR HESSIAN VIA SKETCHING

In this section, we present the fast approximation for hessian via sketching.

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

- *Let $A_1 \in \mathbb{R}^{n \times d}$, let $A_2 \in \mathbb{R}^{n \times d}$*
- *Let $\mathsf{A} = (A_1 \otimes A_2) \in \mathbb{R}^{n^2 \times d^2}$*
- *Let $W \in \mathbb{R}^{n \times n}$ denote a positive diagonal matrix*
- *Let $\overline{A}_1 = W A_1$*
- *Let $\overline{\mathsf{A}} = (\overline{A}_1 \otimes A_2) \in \mathbb{R}^{n^2 \times d^2}$*

Then, we have

- **Part 1.**
$$\mathsf{A}^\top (W^2 \otimes I_n) \mathsf{A} = \overline{\mathsf{A}}^\top \overline{\mathsf{A}}$$
- **Part 2.** *For any constant $\epsilon \in (0, 0.1)$, there is an algorithm runs in $\widetilde{O}(nd + d^4)$ time to compute $S\overline{\mathsf{A}}$ such that*
$$(1-\epsilon) \cdot \overline{\mathsf{A}}^\top \overline{\mathsf{A}} \preceq \overline{\mathsf{A}}^\top S^\top S \overline{\mathsf{A}} \preceq (1+\epsilon) \cdot \overline{\mathsf{A}}^\top \overline{\mathsf{A}}$$
holds with probability $1 - \delta$.
- **Part 3.** *For any $\epsilon \in (0, 0.1)$, there is an algorithm runs in $\widetilde{O}(\mathrm{nnz}(A_1) + \mathrm{nnz}(A_2) + d^4)$ time to compute $S\overline{\mathsf{A}}$ such that*
$$(1-\epsilon) \cdot \overline{\mathsf{A}}^\top \overline{\mathsf{A}} \preceq \overline{\mathsf{A}}^\top S^\top S \overline{\mathsf{A}} \preceq (1+\epsilon) \cdot \overline{\mathsf{A}}^\top \overline{\mathsf{A}}$$
holds with probability $1 - \delta$.

Proof. **Proof of Part 1.**

We can show

$$\begin{aligned} \mathsf{A}^\top (W^2 \otimes I_n) \mathsf{A} = & \ \mathsf{A}^\top (W \otimes I_n) \cdot (W \otimes I_n) \mathsf{A} \\ = & \ ((W \otimes I_n)(A_1 \otimes A_2))^\top \cdot ((W \otimes I_n)(A_1 \otimes A_2)) \\ = & \ (\overline{A}_1 \otimes A_2)^\top (\overline{A}_1 \otimes A_2) \\ = & \ \overline{\mathsf{A}}^\top \overline{\mathsf{A}} \end{aligned}$$

where the first step follows from $(W^2 \otimes I) = (W \otimes I_n) \cdot (W \otimes I_n)$ (where $\otimes$ operation and W is a diagonal matrix), the second step follows from the definition of A,

the third step follows from the definition of $\overline{A}_1$, and the last step follows from the definition of $\overline{\mathsf{A}}$.

Proof of Part 2.

It follows from using Lemma J.3.

Proof of Part 3.

It follows from using Lemma J.5. □

K ANALYSIS OF ALGORITHM 1

We introduce the concept of a (l, M)-good function in Section K.1 and discuss the notion of a well-initialized point. Subsequently, we will present our approximation and update rule methods in Section K.2. In light of the optimization problem introduced in Definition 1.2, we put forward Algorithm 1, and in this section, we establish the correctness and convergence of the algorithm.

K.1 (l, M)-GOOD LOSS FUNCTION

We will now introduce the definition of a (l, M)-Good Loss Function. Next, let's revisit the optimization problem defined in Definition A.7 as follows:

$$L(X, Y) := 0.5 \cdot \| \underbrace{D(X)^{-1}}_{n \times n} \underbrace{\exp(A_1 X A_2^\top)}_{n \times n} \underbrace{A_3}_{n \times d} \underbrace{Y}_{d \times d} - \underbrace{B}_{n \times d} \|_F^2$$

We will now demonstrate that our optimization function possesses the following properties.

Definition K.1 ((l, M)-good Loss function)**.** *For a function $L : \mathbb{R}^d \to \mathbb{R}$, if the following conditions hold,*

- **Hessian is M-Lipschitz.** *If there exists a positive scalar $M > 0$ such that*

$$\|\nabla^2 L(x, y) - \nabla^2 L(\widetilde{x}, \widetilde{y})\| \leq M \cdot (\|x - \widetilde{x}\|_2 + \|y - \widetilde{y}\|_2)$$

- **l-local Minimum.** *Given $l > 0$ as a positive scalar. If there exists a vector $x^* \in \mathbb{R}^{d^2}$ and $y^* \in \mathbb{R}^{d^2}$ such that the following holds*
  - $\nabla L(x^*, y^*) = \mathbf{0}_d$.
  - $\nabla^2 L(x^*, y^*) \succeq l \cdot I_{2d^2}$.
- **Good Initialization Point.** *Let x_0 and y_0 denote the initialization point. If $r_0 := (\|x_0 - x_*\|_2 + \|y_0 - y_*\|_2)$ satisfies*

$$r_0 M \leq 0.1 l.$$

we say L is (l, M)-good

Drawing upon Lemma C.1 and Lemma I.1, we can establish that our loss function (See Definition A.7) satisfies the aforementioned assumption.

K.2 CONVERGENCE

After introducing the approximation method 'Sparsifier via TensorSketch' in Section J, we will now proceed to introduce the update method employed in Algorithm 1. In this section, we demonstrate the concept of approximate update and present an induction hypothesis.

Definition K.2 (Approximate Update)**.** *The following process is considered by us*

$$\begin{bmatrix} x(t+1) \\ y(t+1) \end{bmatrix} \leftarrow \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} - \begin{bmatrix} g(x(t)) \\ g(y(t)) \end{bmatrix} \widetilde{H}^{-1}$$

A tool from previous work is presented by us now.

Lemma K.3 (Iterative shrinking, a variation of Lemma 6.9 on page 32 of Li et al. [2023c])**.** *If the following conditions hold*

- *Loss Function L is (l, M)-good (see Definition K.1).*
- *Let $\epsilon_0 \in (0, 0.1)$ (see Lemma J.6).*
- *Let x^*, y^* be defined in Definition K.1 and x_t, y_t be defined in Definition K.2.*
- *Let $r_t := \|x_t - x^*\|_2 + \|y_t - y^*\|_2$.*
- *Let $\overline{r}_t := M \cdot r_t$*

It follows that

$$r_{t+1} \leq 2 \cdot (\epsilon_0 + \overline{r}_t/(l - \overline{r}_t)) \cdot r_t.$$

In this context, where T denotes the total number of iterations in the algorithm, we require the following lemma based on the induction hypothesis to apply Lemma K.3. This lemma is a well-established concept in the literature, and for further details, you can refer to Li et al. [2023c].

Lemma K.4 (Induction hypothesis, Lemma 6.10 on page 34 of Li et al. [2023c])**.** *If the following condition hold*

- *$\epsilon = 0.01$ (see Lemma J.6)*
- *Let x^*, y^* be defined in Definition K.1 and x_t, y_t be defined in Definition K.2.*
- *Let $r_t := \|x_t - x^*\|_2 + \|y_t - y^*\|_2$.*
- *For each $i \in [T]$, $r_i \leq 0.4 \cdot r_{i-1}$, for all $i \in [t]$*
- *Let l and M be Defined in Definition K.1*
- *$M \cdot r_i \leq 0.1l$, for all $i \in [t]$.*

It follows that

- $r_{t+1} \leq 0.4 r_t$
- $M \cdot r_{t+1} \leq 0.1l$

L MAIN THEOREM

In this section, we incorporate our analysis together and present our main Theorem.

Theorem L.1 (Main Theorem, Formal version of Theorem 1.4)**.** *If the following conditions hold:*

- *Let $A_1, A_2, A_3, B \in \mathbb{R}^{n \times d}$.*
- *Let $X, Y \in \mathbb{R}^{d \times d}$.*
- *Let $D(X) \in \mathbb{R}^{n \times n}$ be defined as $D(X) := \mathrm{diag}(\exp(A_1 X A_2^\top)\mathbf{1}_n)$.*
- *Let $\epsilon \in (0, 0.1)$.*
- *Let $\omega \approx 2.37$.*
- *Let $r_0 = \|x_0 - x^*\|_2 + \|y_0 - y^*\|_2$*

Then, there exists an algorithm (see Algorithm 1) that runs in $\log(r_0/\epsilon)$ iterations and spends

$$\widetilde{O}(\mathcal{T}_{\mathrm{mat}}(n, d, n) + \mathcal{T}_{\mathrm{mat}}(n, d, d) + d^{2\omega})$$

per iteration and solves the attention optimization problem (defined in Definition 1.2):

$$\min_{X, Y \in \mathbb{R}^{d \times d}} \|D(X)^{-1} \exp(A_1 X A_2^\top) A_3 Y - B\|_F^2,$$

and finally outputs $\widetilde{x}$, $\widetilde{y}$ such that

$$(\|\widetilde{x} - x^*\|_2 + \|\widetilde{y} - y^*\|_2) \leq \epsilon$$

with probability $1 - 1/\operatorname{poly}(n)$.

Proof. This follows from combining Lemma B.4, Lemma B.5, Lemma B.3, Lemma C.1, Lemma E.1, Lemma F.1, Lemma G.1, Lemma H.1, and Lemma I.1.

Number of iterations.

By Lemma K.4, we have that

$$(\|x_T - x^*\|_2 + \|y_T - y^*\|_2) \leq 0.4^T (\|x_0 - x^*\|_2 + \|y_0 - y^*\|_2)$$

By choosing $T = \log(r_0/\epsilon)$, the accuracy is satisfied.

Analysis of time complexity.

The analysis of the time complexity can be divided into two parts (forward computation and backward computation).

Proof of forward computation.

This follows from Lemma B.3, where we can compute f, h, c in

$$O(\mathcal{T}_{\mathrm{mat}}(n, d, d) + \mathcal{T}_{\mathrm{mat}}(n, n, d))$$

time.

Proof of gradient computation.

This follows from Lemma B.4 and Lemma B.5, which takes

$$O(\mathcal{T}_{\mathrm{mat}}(n, n, d) + \mathcal{T}_{\mathrm{mat}}(n, d, d))$$

time.

Proof of Hessian computation.

This follows from Lemma J.6, which takes

$$\widetilde{O}(nd) + \mathcal{T}_{\mathrm{mat}}(d^2, d^2, d^2)$$

time.

Proof of g times inverse of approximate Hessian.

The running time of g times inverse of approximate hessian is as follows

$$\mathcal{T}_{\mathrm{mat}}(d^2, d^2, d^2) = d^{2\omega}$$

Therefore, for each iteration, the time spent is as follows

$$\widetilde{O}(\mathcal{T}_{\mathrm{mat}}(n, d, n) + \mathcal{T}_{\mathrm{mat}}(n, d, d) + d^{2\omega})$$

□

M MORE RELATED WORKS

Second-order Method Second-order method have been used for solving many convex optimization and non-convex optimization problems, such as linear programming Cohen et al. [2019], Brand [2020], Jiang et al. [2021], Song and Yu [2021], Gu and Song [2022], Huiberts et al. [2023], empirical risk minimization Lee et al. [2019], Qin et al. [2023b], support vector machines Gu et al. [2025], cutting plan method Lee et al. [2015], Jiang et al. [2020b], semi-definite programming Jiang et al. [2020a], Huang et al. [2022], Gu and Song [2022], Song et al. [2023], hyperbolic programming/polynomials Deng et al. [2023e], Zhang and Zhang [2023], streaming algorithm Liu et al. [2023b], Brand and Song [2023], Song et al. [2023], federated learning Bian et al. [2023].

Convergence and Deep Neural Network Optimization Many works focus on analyzing optimization, convergence guarantees, and training improvement. Li and Liang [2018] shows that stochastic gradient descent optimizes over-parameterized neural networks on structured data, while Du et al. [2019] demonstrates that gradient descent optimizes over-parameterized neural networks. In Allen-Zhu et al. [2019a], a convergence theory for over-parameterized deep neural networks via gradient descent is developed. Allen-Zhu et al. [2019b] analyzes the convergence rate of training recurrent neural networks. Arora et al. [2019a] provides a fine-grained analysis of optimization and generalization for over-parameterized two-layer neural networks. Arora et al. [2019b] studies exact computation with an infinitely wide neural network. Cai et al. [2019] proposes a Gram-Gauss-Newton method for optimizing over-parameterized neural networks. Zou and Gu [2019] improves the analysis of the global convergence of stochastic gradient descent when training deep neural networks, requiring a milder over-parameterization compared to prior research. Other research, such as Oymak and Soltanolkotabi [2020], Ji and Telgarsky [2020a], Zhang et al. [2020b], focuses on optimization and generalization, while Gao et al. [2023a], Li et al. [2023c] emphasize the convergence rate and stability. Works like Brand et al. [2021], Song et al. [2024b], Alman et al. [2024], Munteanu et al. [2022], Zhang [2022], Cao et al. [2024] concentrate on specialized optimization algorithms and techniques for training neural networks, and Lee et al. [2020], Huang et al. [2021] concentrate on leveraging neural network structure.

Algorithmic Regularization There is a significant body of research exploring the latent bias inherent in gradient descent when applied to separable classification tasks. This research typically employs logistic or exponentially-tailed loss functions to maximize margins, as demonstrated in previous studies Ji and Telgarsky [2020b], Gunasekar et al. [2018], Kini et al. [2021], Ji and Telgarsky [2021], Soudry et al. [2018], Moroshko et al. [2020], Nacson et al. [2019]. These novel findings have also been applied to non-separable data through the utilization of gradient-based techniques Ji et al. [2020], Ji and Telgarsky [2019, 2018]. Analysis of implicit bias in regression problems and associated loss functions is carried out using methods such as mirror descent Yun et al. [2021], Amid and Warmuth [2020a,b], Vaskevicius et al. [2019], Sun et al. [2022], Woodworth et al. [2020], Azizan et al. [2021], Gunasekar et al. [2018] and stochastic gradient descent HaoChen et al. [2021], Li et al. [2022], Liang and Rakhlin [2020], Zou et al. [2021], Damian et al. [2021], Li et al. [2019], Blanc et al. [2020]. These findings extend to the implicit bias of adaptive and momentum-based optimization methods Ji et al. [2021], Wang et al. [2021], Qian and Qian [2019].