Differentially Private Attention Computation

Yeqi Gao* Zhao Song[†] Xin Yang[‡]

Abstract

Large language models (LLMs) have had a profound impact on numerous aspects of daily life including natural language processing, content generation, research methodologies and so on. However, one crucial issue concerning the inference results of large language models is security and privacy. In many scenarios, the results generated by LLMs could possibly leak many confidential or copyright information. A recent beautiful and breakthrough work [Vyas, Kakade and Barak 2023] focus on such privacy issue of the LLMs from theoretical perspective. It is well-known that computing the attention matrix is one of the major task during the LLMs computation. Thus, how to give a provable privately guarantees of computing the attention matrix is an important research direction.

Previous work [Alman and Song 2023, Brand, Song and Zhou 2023] have proposed provable tight result for fast computation of attention without considering privacy concerns. One natural mathematical formulation to quantity the privacy in theoretical computer science graduate school textbook is differential privacy. Inspired by [Vyas, Kakade and Barak 2023], in this work, we provide a provable result for showing how to differentially private approximate the attention matrix.

From technique perspective, our result replies on a pioneering work in the area of differential privacy by [Alabi, Kothari, Tankala, Venkat and Zhang 2022].

^{*}a916755226@gmail.com. The University of Washington.

[†]zsong@adobe.com. Adobe Research.

[‡]yangxin199207@gmail.com. The University of Washington.

1 Introduction

The development of large language models (LLMs) has been rapid and significant in recent years, with numerous breakthroughs and advancements in the field. BERT [DCLT18] achieved state-of-the-art performance on a wide range of language tasks by training on a massive amount of text data in 2018. Since then, the GPT (Generative Pre-trained Transformer) family of models has further advanced the field. GPT-2 [RWC+19] and GPT-3 [BMR+20], with billions of parameters, are able to generate highly coherent and human-like text. Other notable LLMs include XLNet [YDY+19], which addresses some of the limitations of BERT [DCLT18], and RoBERTa [LOG+19], which improves upon BERT [DCLT18]'s training methods to achieve better performance. The rapid development of LLMs has been fueled by advancements in hardware, software, and data availability, allowing researchers and companies to train and deploy these models at an unprecedented scale.

As a result of their development, LLMs have found a wide range of applications in various fields. In the field of natural language processing (NLP) [VSP+17, RNS+18, DCLT18, BMR+20], LLMs are used for tasks such as language translation [HWL21], sentiment analysis [UAS+20], and creative writing [Ope23]. In addition, LLMs are being used to develop chatbots and virtual assistants that can understand and respond to natural language queries [BMR+20, Ope23]. Outside of NLP, LLMs are being used in scientific research to generate new hypotheses and discover novel patterns in large datasets. The applications of LLMs are expanding rapidly, and it is likely that they will play an increasingly important role in many fields, such as computer vision [RF18], robotics [KNK21], and autonomous vehicles [ZTL+17, BKO18].

Despite their many benefits, large language models (LLMs) have the potential to pose several privacy and security risks [Sag18, VKB23, KGW⁺23, EMM⁺23]. One concern is the risk of data breaches, as LLMs require large amounts of data to be trained and the data used for training is often collected from public sources without the explicit consent of the individuals involved. This data could include sensitive personal information, such as medical records, financial data, or personally identifiable information [TPG⁺17, ERLD17]. Furthermore, LLMs can potentially be used to generate convincing fake text [RWC⁺19, RSR⁺20], which could be used for malicious purposes such as phishing attacks, spreading misinformation or impersonating individuals online. Additionally, LLMs can be used for so-called "model inversion" attacks [FJR15], where an attacker can extract private information about individuals by querying the model. For example, an attacker could use the model to infer sensitive information, such as an individual's political views or sexual orientation, based on their text input. These privacy and security concerns highlight the need for ethical considerations and responsible use of LLMs, as well as for the development of robust security mechanisms to protect against potential attacks.

As one of the most common large language models, the Transformer [VSP⁺17] has been the focus of several studies on privacy issues related to its training and computation recently. [VKB23] learned conditional generative models can output samples similar to copyrighted data in their training set, which can lead to copyright infringement issues. The proposed solution is near access-freeness (NAF), which involves defining generative models that do not access potentially copyrighted data. [VKB23] provide formal definitions of NAF and generative model learning algorithms that produce models with strong bounds on the probability of sampling protected content. The proposed approach can address privacy concerns related to the use of learned conditional generative models.

Besides the problem mentioned above, the potential harms of large language models also include intellectual property violations and the dissemination of misinformation. To address these issues, a watermarking framework for proprietary language models can be developed [KGW⁺23]. The framework involves embedding invisible signals into generated text that can be algorithmically detected, promoting the use of "green" tokens, and using statistical tests for detection. The

framework has a negligible impact on text quality and includes an efficient open-source algorithm for detection.

Compared to previous works [ZHDK23, AS23, BSZ23, LSZ23, DLS23, GMS23, DMS23, GSY23], our work will concentrate on static computation for attention computation. To be specific, static computation is a technique used in implementing attention mechanisms in deep learning models, especially in the field of natural language processing. It involves computing the attention weights between the encoder and decoder only once and reusing them during decoding, rather than dynamically computing the attention weights for each time step during decoding. This technique can improve computational efficiency and reduce the overall computation time during decoding, particularly for longer sequences.

Here, let us recall the formal mathematical definition of attention computation in static setting,

Definition 1.1 (Attention computation, see [ZHDK23, AS23, BSZ23] as examples). Given matrices $Q \in \mathbb{R}^{n \times d}$, $K \in \mathbb{R}^{n \times d}$ and $V \in \mathbb{R}^{n \times d}$, the goal is to compute

$$\mathsf{Att}(Q, K, V) := D^{-1}AV$$

where $A = \exp(QK^{\top}) \in \mathbb{R}^{n \times n}$ (we apply $\exp()$ entry-wisely to the matrix), and $D = \operatorname{diag}(A\mathbf{1}_n)$.

Following from the setting of work [DMS23], we consider the symmetric attention approximation problem where we treat Q = K and ignore the effect of V. The formal formulation is

Definition 1.2. Given $X \in \mathbb{R}^{n \times d}$, the goal is to find some $Y \in \mathbb{R}^{n \times m}$ such that

$$||D(XX^{\top})^{-1} \exp(XX^{\top}) - D(Y)^{-1} \exp(YY^{\top})|| \le \text{small}$$

where $\|\cdot\|$ is some certain norm and $D(XX^{\top}) = \operatorname{diag}(\exp(XX^{\top}) \cdot \mathbf{1}_n)$.

It is crucial to consider privacy in attention computation, as attention mechanisms require encoding and decoding of input data, which may contain sensitive personal information or trade secrets. This sensitive information could potentially be exposed through the attention weights in the model. Specifically, if sensitive information such as personal identifying information or trade secrets are included in the computation of attention weights, this information could potentially be exposed if the model weights are compromised. As such, our research will specifically concentrate on addressing privacy and security issues in the application of static computation for attention, which is a critical component in large language models.

In the recent work by Vyas, Kakade and Barak [VKB23], they choose the angle of near access-freeness to study the privacy concerns in LLMs. In this work, we use the differential privacy notation which is a common concept in graduate school textbook, the formal definition of differential privacy can be written as follows.

Definition 1.3 (Differential Privacy [DMNS06, DKM⁺06]). A randomized mechanism \mathcal{M} is (ϵ, δ) -differentially private if for any event $\mathcal{O} \in \text{Range}(\mathcal{M})$ and for any pair of neighboring databases S, S' that differ in a single data element, one has

$$\Pr[\mathcal{M}(S) \in \mathcal{O}] \le \exp(\epsilon) \cdot \Pr[\mathcal{M}(S') \in \mathcal{O}] + \delta.$$

Finally, we're ready to define our differentially private attention computation problem

Definition 1.4. For a given matrix $X \in \mathbb{R}^{n \times d}$ with $d \gg n$, let \mathcal{M} denote some mapping that maps $\mathbb{R}^{n \times d}$ to $\mathbb{R}^{n \times n}$, let $A = \mathcal{M}(X)$, for parameter $\epsilon, \delta \in (0, 0.1)$, the goal is to design an (ϵ, δ) -differentially private algorithm that generates a PSD matrix $B \in \mathbb{R}^{n \times n}$ such that

$$||D(A)^{-1}f(A) - D(B)^{-1}f(B)|| \le g(\epsilon, \delta)$$

where $f(z) \in \{\exp(z), \cosh(z)\}, D(A) = \operatorname{diag}(f(A)\mathbf{1}_n)$ and where g is some function.

1.1 Our Result

Our results rely on good properties of the input data, which are defined as follows.

Definition 1.5 (Dataset). Fix $\eta > 0$, $\alpha > 0$. We say our dataset X is (η, α) -good if

- $XX^{\top} \succeq \eta \cdot I_n$.
- For all $i \in [d], ||X_{*,i}||_2 \le \alpha$.

Definition 1.6 (Neighboring data). Let X, \widetilde{X} denote two datasets from distribution \mathcal{D} , we say that X and \widetilde{X} are β -close if

- there exists exact one $i \in [d]$ so that $||X_{*,i} \widetilde{X}_{*,i}||_2 \le \beta$
- for all $j \in [d] \setminus \{i\}, X_{*,j} = \widetilde{X}_{*,j}$

In this work we consider two datasets to be neighboring if they are β -close.

We state our result as follows:

Theorem 1.7 (Main result, informal of Theorem 8.1). If the following conditions hold

- Let $d \ge n$.
- Let $X \in \mathbb{R}^{n \times d}$.
- We define $r \in (0, 0.1)$ as bounded ratio.
- Let $f(z) \in {\exp(z), \cosh(z)}$.
- Let $\epsilon \in (0, 0.1)$ denote the parameter of DP.
- We define $\delta \in (0,0.1)$ as the parameter of DP.
- Let $\Delta = 0.1 \min\{\frac{\epsilon}{\sqrt{k \log(1/\delta)}}, \frac{\epsilon}{\log(1/\delta)}\}$
- Let $A = \mathcal{M}(X) = XX^{\top}$
- Let $||A||_{\infty} \leq r$
- For all X sampled from \mathcal{D} , X is (α, η) -good (see Definition 1.5).
- Let $\eta < r$.
- Let β be the parameter for neighboring dataset.
- Let $2\alpha\beta\sqrt{n}/\eta < \Delta$
- Let Δ denote the sensitivity parameter that $\mathcal M$ satisfies a sensitivity bound that

$$\|\mathcal{M}(X)^{1/2}\mathcal{M}(\widetilde{X})^{-1}\mathcal{M}(X)^{1/2} - I\|_F \le \Delta$$

for any neighboring datasets $X \in \mathbb{R}^{n \times d}$, $\widetilde{X} \in \mathbb{R}^{n \times d}$ (see Definition 1.6).

• Let
$$\rho = \sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k$$

• Let $\rho < 0.1\epsilon$

An algorithm exists that can take the input $A = \mathcal{M}(X)$ and produce the matrix B as output such that

- $\| D(A)^{-1} f(A) D(B)^{-1} f(B) \|_{\infty} \le 4 \cdot (1 + \epsilon + 2r) \cdot r$
- It holds with probability 1γ .
- With respect to X, the algorithm is (ϵ, δ) -differential private.

Roadmap. Our paper is organized as follows. We provide an overview of our techniques in Section 2. Section 3 contains the preliminary information required for our work. In Section 4, we analyze the perturbations in attention computation. We introduce some useful tools related to differential privacy in Section 5. Section 6 presents the proof of the existence of differential privacy using our Gaussian sampling mechanism. In Section 7, we provide sensitivity bound. Finally, our main result is presented in Section 8, by combining the conclusions from Section 6 and Section 4.

2 Technique Overview

The objective of our research is to develop a differential privacy algorithm that addresses the challenges of computing attention on large datasets. Specifically, we focus on scenarios where the size of the data matrix X is extremely large, with the number of features d significantly exceeding the number of samples n (i.e., $d \gg n$). In these cases, the attention matrix A is obtained as the output of the function $\mathcal{M}(X) = XX^{\mathsf{T}}$, and our goal is to ensure that the computation of A is performed in a differentially private [DMNS06, DKM⁺06] manner.

Perturb PSD Matrix We define the attension computation D(X) as Definiton 4.2. By employing a more general version of Perturbation analysis presented in [DMS23], we select f as specified in Definition 4.1. To complete the error analysis of attention computation, we will utilize the perturbation analysis of the diagonal normalization matrix and the PSD matrix presented in Section 4.3. Under the assumption the relative error between input matrix $\mathcal{M}(X)$ and privacy required matrix

output B is less than or equal to $\epsilon \in (0, 0.1)$ where

$$(1 - \epsilon)B \leq A \leq (1 + \epsilon)B$$
.

And with the error of attention computation under control, we can obtain:

$$\| \operatorname{D}(A)^{-1} f(A) - \operatorname{D}(B)^{-1} f(B) \|_{\infty} \le 4 \cdot (1 + \epsilon + 2r) \cdot r$$

A (α, η) -good Dateset Our work relies on the basic assumptions that $X \in \mathbb{R}^{n \times d}$ is a (η, α) -good dataset (See Definition 1.5) and that X and \widetilde{X} are β -close to each other (See Definition 1.6). We choose $\mathcal{M}(X) := XX^{\top}$. Now we will demonstrate the property of our function $\mathcal{M}(X) = XX^{\top}$ based on the given assumptions. Since X and \widetilde{X} are neighbor datasets, we have the following:

$$\|\mathcal{M}(X)^{1/2}\mathcal{M}(\widetilde{X})^{-1}\mathcal{M}(X)^{1/2} - I\|_F \le 2\alpha\beta\sqrt{n}$$

The proof details can be found in Section 7, which can be easily derived from Fact 3.2. Let us denote Δ as defined in Definition 6.6. By choosing $2\alpha\beta\sqrt{n}/\eta < \Delta$, we will have

$$\|(\underbrace{XX^{\top}}_{:=\mathcal{M}(X)})^{1/2}(\underbrace{\widetilde{X}\widetilde{X}^{\top}}_{:=\mathcal{M}(X)})^{-1}(\underbrace{XX^{\top}}_{:=\mathcal{M}(X)})^{1/2} - I\|_{F} \le \Delta \tag{1}$$

The assumption specified in the **Requirement 5** of Theorem 6.12 will be satisfied. Next, we will introduce our main algorithm using Eq. (1).

Differential Privacy Algorithm Next, we will demonstrate that our algorithm is able to output a matrix that satisfies the **Part 1** of our main result (See Theorem 8.1).

To begin with, we demonstrate that there exists an algorithm capable of taking input A and producing a matrix B as output such that the difference between A and B is small enough, which can be seen as a small error resulting from the perturbation of A by

$$\underbrace{O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)}_{:=\rho}.$$

In other words, we have

$$(1-\rho)A \leq B \leq (1+\rho)A$$
.

The above equation holds with probability $1 - \gamma$. Note that k and γ can be chosen according to our requirements. We can ensure that a satisfactory ρ is obtained. By choosing a small enough $\rho \leq 0.1\epsilon$ and using the conclusions on perturbed PSD matrices, the algorithm can certainly output a satisfactory B which promises our attention computation is privacy [DMNS06, DKM⁺06].

3 Preliminary

Section 3.1 presents the notations that are used throughout our paper. These notations are essential for a clear and concise presentation of our work. In Section 3.2, we provide an introduction to some basic algebraic concepts that are relevant to our research. This includes fundamental mathematical operations and properties that are used in the analysis and development of our differential privacy algorithm.

3.1 Notations

For a event C, $\Pr[C]$ represents the probability of event C occurring. $\mathbb{E}[X]$ represents the expected value (or mean) of a random variable X.

We use χ_d^2 to denote a Chi-squared random variable with d degrees of freedom. N represents the set of natural numbers, which consists of all positive integers including 1, 2, 3, and so on.

If M and N are symmetric matrices, we define $M \succeq N$ to mean that for all vectors x, the inequality $x^{\top}Mx \geq x^{\top}Nx$ holds. If M is a symmetric matrix of dimension $n \times n$, we define M to be positive semidefinite $(M \succeq 0)$ if the inequality $x^{\top}Mx \geq 0$ holds for all vectors $x \in \mathbb{R}^n$.

We use the notation $\mathbf{0}_n$ to denote an *n*-dimensional vector whose entries are all zero, and $\mathbf{1}_n$ to denote an *n*-dimensional vector whose entries are all one. The symbol I_n represents the $n \times n$ identity matrix, which is a square matrix with ones on the main diagonal and zeros elsewhere.

Let x be an arbitrary vector in \mathbb{R}^n . We define $\exp(x) \in \mathbb{R}^n$ as a vector whose i-th entry $\exp(x)_i$ is equal to $\exp(x_i)$, where $\exp(\cdot)$ denotes the exponential function. We use $\langle x, y \rangle$ to denote $\sum_{i=1}^n x_i y_i$.

For any matrix A, we use ||A|| to denote the spectral norm of A, i.e., $||A|| = \max_{||x||_2=1} ||Ax||_2$, $||A||_F$ to denote its Frobenius norm and $||A||_{\infty}$ to denote the infinity norm. $A_{i,j}$ represents the element in the i-th row and j-th column of matrix A. det(A) represents the determinant of matrix A. For a square and symmetric matrix $A \in \mathbb{R}^{n \times n}$, we say A positive semi-definite $(A \succeq 0)$ if for all vectors $x \in \mathbb{R}^n$, we have $x^\top A x \geq 0$.

We denote the inverse of a matrix M as M^{-1} and its transpose as M^{\top} . We refer to λ_i as the i-th eigenvalue of N.

 \mathbb{S}^n_+ denotes the set of $n \times n$ positive semidefinite (PSD) matrices.

3.2 Basic Algebra

Fact 3.1. We have

- Part 1. $cosh(x) = \sum_{i=0}^{\infty} (1/(2i)!) \cdot x^{2i}$.
- Part 2. $\exp(x) = \sum_{i=0}^{\infty} (1/(i!)) \cdot x^i$.
- Part 3. We have $|\exp(x) 1| \le |x| + x^2$, $\forall x \in (-0.1, 0.1)$.
- Part 4. $|\exp(x) \exp(y)| \le \exp(x) \cdot (|x-y| + |x-y|^2)$ for $|x-y| \le 0.1$.
- Part 5. We have $|\cosh(x) 1| \le x^2$, $\forall x \in (-0.1, 0.1)$.
- Part 6. $|\cosh(x) \cosh(y)| \le \cosh(x) \cdot |x y|^2$ for $|x y| \le 0.1$.

Fact 3.2. We have

- Part 1. Let $A \in \mathbb{R}^{n \times n}$, then we have $||A||_F \leq \sqrt{n} ||A||$.
- Let $A \in \mathbb{R}^{n \times n}$, then we have $||A|| \leq ||A||_F$
- For two vectors $a, b \in \mathbb{R}^n$, then we have $||ab^{\top}|| \leq ||a||_2 \cdot ||b||_2$

4 Error Control from Logit Matrix to Attention Matrix

Section 4.1 provides definitions of key terms and concepts in Section 4. In Section 4.2, we discuss the perturbation of positive semi-definite (psd) matrices, which is a crucial step in ensuring the differential privacy of our algorithm. Section 4.3 focuses on the perturbation of diagonal normalization matrices, which is another important aspect of our error control approach. In Section 4.4, we analyze the error in the attention matrix computation that arises from these perturbations. Finally, in Section 4.5, we present the main result of Section 4, which summarizes the effectiveness of our error control mechanisms in achieving differential privacy for the computation of the attention matrix.

4.1 Definitions

This section introduces the definitions of the key terms and concepts used in Section 4.

Definition 4.1. Let f(z) denote one of the following functions

- $\bullet \exp(z)$
- $\cosh(z)$

The motivation of considering $\exp(z)$ is due to recent LLMs. The motivation of considering $\cosh(z)$ is from recent progress in potential function design of convex optimization [CLS19, LSZ19, Son19, Bra20, JSWZ21, DLY21, GS22, QSZZ23].

Definition 4.2. Given that $A \in \mathbb{R}^{n \times n}$, we define f as Definition 4.1. Let us define

$$\mathsf{D}(A) := \mathrm{diag}(f(A)\mathbf{1}_n)$$

where we apply f to matrix entrywisely.

4.2 Perturb PSD Matrix

In Section 4.2, we discuss the perturbation of positive semi-definite (psd) matrices. This is a crucial step in ensuring the differential privacy of our algorithm.

Lemma 4.3 (Lemma 3.1 in [DMS23]). We denote $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times n}$ as psd matrices. If all of the following requirements are met

- Requirement 1. We have $-r \leq A_{i,j} \leq r, \forall (i,j) \in [n] \times [n]$.
- Requirement 2. $(1 \epsilon)B \leq A \leq (1 + \epsilon)B$;

Then, it follows that

$$B_{i,j} \in [-(1+\epsilon)r, (1+\epsilon)r].$$

Lemma 4.4 (A general version of Lemma 3.2 in [DMS23]). If all of the following requirements are met

- Requirement 1. $A_{i,j} \in [-r,r]$.
- Requirement 2. $B_{i,j} \in [-(1+\epsilon)r, (1+\epsilon)r].$
- Requirement 3. $r \in (0, 0.1), \epsilon \in (0, 0.1)$.
- Requirement 4. Let $f(z) \in \{\exp(z), \cosh(z)\}.$

It follows that

• Part 1.

$$|f(A_{i,j}) - f(B_{i,j})| \le f(A_{i,j}) \cdot (2 + 2\epsilon + 4r) \cdot r \quad \forall i, j \in [n] \times [n].$$

• Part 2.

$$|f(A_{i,j}) - f(B_{i,j})| \le f(B_{i,j}) \cdot (2 + 2\epsilon + 4r) \cdot r \ \forall i, j \in [n] \times [n].$$

Proof. According to Requirement 1., Requirement 2. and Requirement 3., we have

$$|A_{i,j} - B_{i,j}| \le (2 + \epsilon)r. \tag{2}$$

Proof of Part 1. It follows that

$$|f(A_{i,j}) - f(B_{i,j})| \le f(A_{i,j}) \cdot (|A_{i,j} - B_{i,j}| + |A_{i,j} - B_{i,j}|^2)$$

$$\le f(A_{i,j}) \cdot |A_{i,j} - B_{i,j}| \cdot (1 + |A_{i,j} - B_{i,j}|)$$

$$\le f(A_{i,j}) \cdot |A_{i,j} - B_{i,j}| \cdot (1 + (2 + \epsilon)r)$$

$$\le f(A_{i,j}) \cdot (2 + \epsilon)r \cdot (1 + (2 + \epsilon)r)$$

$$= f(A_{i,j}) \cdot (2 + \epsilon + (2 + \epsilon)^2 r)r$$

$$\le f(A_{i,j}) \cdot (2 + 2\epsilon + 4r)r$$

where the 1st step is the result of Fact 3.1, the 2nd step follows from straightforward algebraic manipulations, the 3rd step is a consequence of Eq.(2), the 4th step is a consequence of Eq.(2), the 5th step follows from algebraic manipulations, and the 6th step is a result of satisfying **Requirement** 3 in the Lemma statement.

Proof of Part 2. Similarly, we can prove it.

4.3 Error Control for Normalization

This section focuses on the perturbation of diagonal normalization matrices, which is another important aspect of our error control approach.

Lemma 4.5 (Error Control for Normalization, A general version Lemma 3.3 in [DMS23]). If the following condition holds

- Requirement 1. We define f as Definition 4.1.
- Requirement 2. We define D as Definition 4.2.
- Requirement 3. $\forall (i,j) \in [n] \times [n]$, we have $|f(A_{i,j}) f(B_{i,j})| \leq f(A_{i,j}) \cdot c_0 r$.
- Requirement 4. $\forall (i,j) \in [n] \times [n]$, we have $f(A_{i,j}) f(B_{i,j}) | \leq f(B_{i,j}) \cdot c_0 r$.

Then, it follows that,

• Part 1.

$$|\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}| \le \mathsf{D}(A)_{i,i} \cdot c_0 r \ \forall i \in [n]$$

• Part 2.

$$|\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}| \le \mathsf{D}(B)_{i,i} \cdot c_0 r \quad \forall i \in [n]$$

Proof. **Proof of Part 1.** From the above conditions in the lemma statement, it follows that

$$| D(A)_{i,i} - D(B)_{i,i}| = |(f(A_{i,*}) - f(B_{i,*})) \cdot \mathbf{1}_n|$$

$$= | \sum_{j=1}^n (f(A_{i,j}) - f(B_{i,j}))|$$

$$\leq \sum_{j=1}^n |f(A_{i,j}) - f(B_{i,j})|$$

$$\leq \sum_{j=1}^{n} f(A_{i,j}) \cdot c_0 r$$
$$= f(A_{i,*}) \mathbf{1}_n \cdot c_0 r$$
$$= D(A)_{i,i} \cdot c_0 r$$

where the 1st step follows from algebraic manipulations, the 2nd step is due to algebraic manipulations, the 3rd step is the result of triangle inequality, the 4th step is based on **Requirement 2** in Lemma statement, the 5th step comes from algebraic manipulations and the last step is the result of algebraic manipulations.

Proof of Part 2.

The proof is similar to Part 1. So we omit the details here.

4.4 Error of Attention Matrix

In this section, we analyze the error in the attention matrix computation that arises from the perturbations of psd and diagonal normalization matrices.

Lemma 4.6 (A general version of Lemma 3.4 in [DMS23]). Let $c_1 > 0$ and $c_2 > 0$. If all of the following requirements are met

- Requirement 1. We define f as Definition 4.1.
- Requirement 2. We define D as Definition 4.2.
- Requirement 3.

$$|\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}| \le c_1 \cdot r \cdot \min\{\mathsf{D}(A)_{i,i}, \mathsf{D}(B)_{i,i}\} \quad \forall i \in [n],$$

• Requirement 4.

$$|f(A_{i,j}) - f(B_{i,j})| \le c_2 \cdot r \cdot \min\{f(A_{i,j}), f(B_{i,j})\} \ \forall i, j \in [n] \times [n]$$

It follows that

$$\| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty} \le (c_1 + c_2) \cdot r.$$

Proof. We first decompose the difference into

$$\| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty}$$

$$\leq \| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty} + \| D(B)^{-1} f(B) - D(B)^{-1} f(B) \|_{\infty}$$

$$= Z_1 + Z_2$$

where last step is obtained by

$$Z_1 := \| \mathsf{D}(B)^{-1} f(B) - \mathsf{D}(B)^{-1} f(B) \|_{\infty},$$

and

$$Z_2 := \| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty}.$$

We will present the proof in two parts.

The first term. $\forall (i,j) \in [n] \times [n]$, it follows that

$$Z_{1} = |(\mathsf{D}(A)^{-1}f(A) - \mathsf{D}(B)^{-1}f(B))_{i,j}|$$

$$= |\mathsf{D}(A)_{i,i}^{-1} \cdot (f(A)_{i,j} - f(B)_{i,j})|$$

$$\leq \mathsf{D}(A)_{i,i}^{-1} \cdot |f(A)_{i,j} - f(B)_{i,j})|$$

$$\leq \mathsf{D}(A)_{i,i}^{-1} \cdot c_{2} \cdot r \cdot f(A)_{i,j}$$

$$\leq c_{2}r \cdot (\mathsf{D}(A)^{-1}f(A))_{i,j}$$

$$\leq c_{2}r,$$

where the 1st step comes from definition, the 2nd step is the result of algebraic manipulations, the 3rd step comes from triangle inequality, the 4th step is based on **Requirement 4** in the lemma statement, the 5th step is the result of algebraic manipulations, and the last step is according to the definition of D.

The second term. $\forall (i,j) \in [n] \times [n]$, it follows that

$$\begin{split} Z_2 &= |(\mathsf{D}(B)^{-1} f(B) - \mathsf{D}(B)^{-1} f(B))_{i,j}| \\ &= |(\mathsf{D}(A)_{i,i}^{-1} - \mathsf{D}(A)_{i,i}^{-1}) f(B)_{i,j}| \\ &= |\frac{\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}}{\mathsf{D}(A)_{i,i} \mathsf{D}(B)_{i,i}} f(B)_{i,j}| \\ &\leq |\frac{\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}}{\mathsf{D}(A)_{i,i} \mathsf{D}(B)_{i,i}}| \cdot |f(B)_{i,j}| \\ &\leq |\frac{c_1 r \, \mathsf{D}(A)_{i,i}}{\mathsf{D}(A)_{i,i} \, \mathsf{D}(B)_{i,i}}| \cdot |f(B)_{i,j}| \\ &= c_1 r \cdot |\mathsf{D}(B)_{i,i}^{-1}| \cdot |f(B)_{i,j}| \end{split}$$

where the 1st step based on definition, the 2nd steps follow from algebraic manipulations, the 3rd step is the result of algebraic manipulations, the 4th step is due to triangle inequality, the 5th step is due to **Requirement 3** in the lemma statement, the last step is due to algebraic manipulations.

Then we have

$$Z_2 = c_1 r \cdot |\mathsf{D}(B)_{i,i}^{-1}| \cdot |f(B)_{i,j}|$$

$$= c_1 r \cdot |\mathsf{D}(B)_{i,i}^{-1} f(B)_{i,j}|$$

$$= c_1 r \cdot (\mathsf{D}(B)^{-1} f(B))_{i,j}$$

$$\leq c_1 r$$

where the 1st step is the result of the above equation, the 2nd step is due to all the entries are positive, the 3rd step is due to algebraic manipulations and the last step is due to definition of D.

Based on the above deduction, it follows that

$$\| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty} \le Z_1 + Z_2$$

$$< (c_1 + c_2)r.$$

Thus we complete the proof.

4.5 Main Result

The main result of Section 4 is presented in this section.

Theorem 4.7. If all of the following requirements are met

- Let $\epsilon \in (0, 0.1)$
- Let $r \in (0, 0.1)$
- $||A||_{\infty} \leq r$
- $(1 \epsilon)B \leq A \leq (1 + \epsilon)B$
- We define D Definition 4.2.
- We define f as Definition 4.1.

It follows that

$$\| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty} \le 4 \cdot (1 + \epsilon + 2r) \cdot r$$

Proof. By Lemma 4.3 and $(1 - \epsilon)B \leq A \leq (1 + \epsilon)B$, we have

$$B_{i,j} \in [-(1+\epsilon)r, (1+\epsilon)r]. \tag{3}$$

By Lemma 4.4 and Eq. (3), it follows that

• Part 1.

$$|f(A_{i,j}) - f(B_{i,j})| \le f(A_{i,j}) \cdot (2 + 2\epsilon + 4r) \cdot r \ \forall (i,j) \in [n] \times [n].$$

• Part 2.

$$|f(A_{i,j}) - f(B_{i,j})| \le f(B_{i,j}) \cdot (2 + 2\epsilon + 4r) \cdot r \ \forall (i,j) \in [n] \times [n].$$

According to the discussion above and using Lemma 4.5, we have

• Part 1.

$$|\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}| \le \mathsf{D}(A)_{i,i} \cdot c_0 r \ \forall i \in [n]$$

• Part 2.

$$|\mathsf{D}(A)_{i,i} - \mathsf{D}(B)_{i,i}| \le \mathsf{D}(B)_{i,i} \cdot c_0 r \ \forall i \in [n]$$

And then by using Lemma 4.6, $c_1 = (2 + 2\epsilon + 4r)$ and $c_2 = (2 + 2\epsilon + 4r)$, we have

$$\| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty} \le 4 \cdot (1 + \epsilon + 2r) \cdot r$$

5 Differential Privacy

This section introduces several differential privacy tools that will be used in the proof of Section 6. These tools are essential for demonstrating the differential privacy properties of our algorithm.

Theorem 5.1 (Empirical covariance estimator for Gaussian [Ver18]). Let $\Sigma \in \mathbb{R}^{d \times d}$ be PSD, $X_1, \dots, X_n \sim \mathcal{N}(0, \Sigma)$ be i.i.d and $\widetilde{\Sigma} = \frac{1}{n} \sum_{i=1}^n X_i X_i^{\top}$. Then with probability $1 - \gamma$, it holds that

$$\|\Sigma^{-1/2}\widetilde{\Sigma}\Sigma^{-1/2} - I\|_F \le \rho$$

for some
$$\rho = O(\sqrt{\frac{d^2 + \log(1/\gamma)}{n}} + \frac{d^2 + \log(1/\gamma)}{n})$$
.

Theorem 5.2 (Lemma 1.5 in [Vad17], Section 1.1 of [BS16]). For a (randomized) mechanism \mathcal{M} and datasets x, y, define the function

$$f_{xy}(z) := \log(\frac{\Pr[\mathcal{M}(x) = z]}{\Pr[\mathcal{M}(y) = z]})$$

If $\Pr[f_{xy}(\mathcal{M}(x)) > \epsilon] \leq \delta$ for all adjacent datasets x, y, then \mathcal{M} is (ϵ, δ) -DP.

Lemma 5.3 (Sub-exponential tail bound, Proposition 2.9 in [Wai19]). Suppose that X is sub-exponential with parameters (ν, α) . Then

$$\Pr[X - \mu \ge t] \le \max\{\exp(-\frac{t^2}{2v^2}), \exp(\frac{t}{2\alpha})\}$$

Lemma 5.4 (χ_1^2 sub-exponential parameters, Example 2.11 in [Wai19]). A chi-squared random variable with 1 degree of freedom (χ_1^2) is sub-exponential with parameters (ν , α) = (2, 4)

Lemma 5.5 (Sub-exponential parameters of independent sum, Chapter 2 of [Wai19]). Consider an independent sequence X_1, \dots, X_k of random variables, such that X_i is sub-exponential with parameters (ν_i, α_i) . Then the variable $\sum_{i=1}^k X_i$ is sub-exponential with parameters (ν_*, α_*) , where

$$a_* = \max_{i \in [k]} \alpha_i \quad and \quad \nu_* = (\sum_{i=1}^k \nu_i^2)^{1/2}.$$

6 Analysis of Gaussian Sampling Mechanism

We denote the output of our privacy algorithm as Z. In Section 6.1, we introduce the definition of Z and some other key concepts. In Section 6.2, we present the computation tools that we use to implement our approach. In Section 6.3, we perform spectral decomposition of $\mathcal{M}(\mathcal{Y})^{1/2}\mathcal{M}(\mathcal{Y}')^{-1}\mathcal{M}(\mathcal{Y})^{1/2}$ and derive some important conclusions from it. Then, in Section 6.4,

we transform Z into a format that is based on the spectral decomposition of A. In Section 6.5, We present the upper bound of $\mathbb{E}[Z]$, which is useful in the following section. In Section 6.6, we demonstrate that Z is sub-exponential, which allows us to control the upper bound of $\Pr[Z \geq \epsilon]$ where $\epsilon \in (0,1)$. Finally, we present our main result in Section 6.7, which is that our Algorithm 1 is differential privacy.

6.1 Definitions

This section is dedicated to introducing several key concepts that are crucial for understanding our approach to achieving differential privacy.

Definition 6.1. We denote the $\mathcal{N}(0,\Sigma)$ density function as follows

$$f_{\Sigma}(x) = (2\pi)^{-\frac{n}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp(-0.5x^{\top}\Sigma x)$$

Definition 6.2. Let $\mathcal{M}: (\mathbb{R}^n)^d \to \mathbb{R}^{n \times n}$ be a (randomized) algorithm that given a dataset of d points in \mathbb{R}^n outputs a PSD matrix. Then, we define

$$M := \|\mathcal{M}(\mathcal{Y})^{1/2}\mathcal{M}(\mathcal{Y}')^{-1}\mathcal{M}(\mathcal{Y})^{1/2} - I\|_F$$

Definition 6.3. Let g_1, g_2, \dots, g_k be i.i.d samples from $\mathcal{N}(0, \Sigma_1)$ output by Algorithm 1. Then, we define

- $h_{i,j} := \langle \Sigma_1^{-1/2} g_i, v_j \rangle$
- $Z := \sum_{i=1}^k \log(\frac{f_{\Sigma_1}(g_i)}{f_{\Sigma_2}(g_i)})$

Note that the random variables $h_{i,j}$ are i.i.d copies of $\mathcal{N}(0,1)$.

Definition 6.4. Let \mathcal{M} be denoted in Definition 6.2 and $\Sigma(\mathcal{Y}) := \mathcal{M}(\mathcal{Y})$. We define

- $\Sigma_1 := \Sigma(\mathcal{Y})$
- $\Sigma_2 := \Sigma(\mathcal{Y}')$

Definition 6.5. We define Σ_1, Σ_2 as Definition 6.4. Let us define

- $A := \Sigma_1^{1/2} \Sigma_2^{-1} \Sigma_1^{1/2}$
- $B := \Sigma_2^{1/2} \Sigma_1^{-1} \Sigma_2^{-1/2}$
- $C := \Sigma_1^{-1/2} \Sigma_2^{1/2}$

Definition 6.6. We define

$$\Delta := \min \left\{ \frac{\epsilon}{\sqrt{8k \log(1/\delta)}}, \frac{\epsilon}{8 \log(1/\delta)} \right\}$$

6.2 Computation Tools

This section is dedicated to presenting the computational tools that we use to implement our approach.

Lemma 6.7. Let A, B and C be defined as Definition 6.5. Then we have

- Part 1. $A^{-1} = CC^{\top}$.
- Part 2. $B = C^{\top}C$.
- Part 3. A^{-1} , B have the same eigenvalue.

Proof. Note that Σ_1 and Σ_2 are symmetric, we can easily have the proof as follows.

Proof of Part 1.

$$A^{-1} = (\Sigma_{1}^{1/2} \Sigma_{2}^{-1} \Sigma_{1}^{1/2})^{-1}$$

$$= (\Sigma_{1}^{1/2} \Sigma_{2}^{-1/2} \Sigma_{2}^{-1/2} \Sigma_{1}^{1/2})^{-1}$$

$$= (\Sigma_{2}^{-1/2} \Sigma_{1}^{1/2})^{-1} (\Sigma_{1}^{1/2} \Sigma_{2}^{-1/2})^{-1}$$

$$= (\Sigma_{1}^{1/2} \Sigma_{2}^{-1/2}) (\Sigma_{2}^{-1/2} \Sigma_{1}^{1/2})$$

$$= CC^{\top}$$
(4)

Proof of Part 2.

$$B = \Sigma_2^{-1/2} \Sigma_1 \Sigma_2^{-1/2}$$

$$= (\Sigma_2^{-1/2} \Sigma_1^{1/2}) (\Sigma_1^{1/2} \Sigma_2^{-1/2})$$

$$= C^T C$$
(5)

Proof of Part 3. It simply follows from Eq.(4) and Eq.(5).

6.3 Spectral Decomposition

This section is focused on the spectral decomposition of A, which we perform to gain insights into its properties. By analyzing the spectral decomposition, we are able to draw important conclusions about A that are relevant to our approach.

Lemma 6.8. If all of the following requirements are met

- Requirement 1. We define A as Definition 6.5.
- Requirement 2. Let $\lambda_1 \cdots \lambda_n$ be eigenvalues of A.
- Requirement 3. Let $A = \sum_{j=1}^{n} \lambda_j v_j v_j^{\top}$ be spectral decomposition for A.
- Requirement 4. Let Δ be denoted as Definition 6.6.
- Requirement 5. Let M, \mathcal{M} be denoted as Definition 6.2 and $M \leq \Delta$.

We have

- $\bullet \sum_{i=1}^{n} (\lambda_i 1)^2 \le \Delta^2.$
- $\sum_{j=1}^{n} (1 \frac{1}{\lambda_i})^2 \le \Delta^2$.

Proof. we have

$$\sum_{j=1}^{n} (\lambda_j - 1)^2 = ||A - I||_F^2$$

$$\leq \Delta^2$$

where the 1st step is based on **Requirement 3** in the lemma statement and the last step is due to **Requirement 5** in lemma statement.

Similarly, we have

$$\sum_{j=1}^{n} (1 - \frac{1}{\lambda_j})^2 = \|I - A^{-1}\|_F^2$$
$$= \|I - B\|_F^2$$
$$\leq \Delta^2$$

where the 1st step is due to **Requirement 3** in the lemma statement, the 2nd step follows from swapping the roles of $\mathcal{Y}, \mathcal{Y}'$ and the last step is due to Lemma 6.7.

6.4 The transformation for Output

In Section 6.4, we describe the process of transforming the output Z of our privacy algorithm into a format that is based on the spectral decomposition of A.

Lemma 6.9. If all of the following requirements are met

- Requirement 1. We define Z and $h_{i,j}$ as Definition 6.3.
- Requirement 2. Let A be denoted as Definition 6.5.
- Requirement 3. Let $\lambda_1, \dots, \lambda_n$ demote the eigenvalue of A.

Then we have

$$Z = \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{n} \left((\lambda_j - 1) h_{i,j}^2 - \log(\lambda_j) \right)$$

Proof. The privacy loss random variable Z can be expressed as follows:

$$Z = \sum_{i=1}^{k} \log \left(\frac{\det(\Sigma_{1})^{-\frac{1}{2}} \exp(-\frac{1}{2}g_{i}^{\top} \Sigma_{1}^{-1}g_{i})}{\det(\Sigma_{2})^{-\frac{1}{2}} \exp(-\frac{1}{2}g_{i}^{\top} \Sigma_{2}^{-1}g_{i})} \right)$$

$$= \sum_{i=1}^{k} \left(\frac{1}{2} g_{i}^{\top} (\Sigma_{2}^{-1} - \Sigma_{1}^{-1}) g_{i} - \frac{1}{2} \log \left(\frac{\det(\Sigma_{1})}{\det(\Sigma_{2})} \right) \right)$$

$$= \frac{1}{2} \sum_{i=1}^{k} \left(\left(\Sigma_{1}^{-1/2} g_{i} \right)^{\top} (A - I) \left(\Sigma_{1}^{-1/2} g_{i} \right) - \log \det(A) \right)$$

$$= \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{n} \left((\lambda_{j} - 1) h_{i,j}^{2} - \log(\lambda_{j}) \right)$$

where the 1st step is based on **Requirement 1** in the lemma statement, the 2nd step follows from rearranging the terms, the 3rd step is based on **Requirement 2** in the lemma statement, and the last step is by taking the spectral decomposition of A.

6.5 The Upper Bound for Expectation

In Section 6.5, we provide an upper bound on the expected value of Z, which is a useful result for the subsequent section.

Lemma 6.10. If all of the following requirements are met

- Requirement 1 We define Z as Definition 6.3.
- Requirement 2 Let $\epsilon \in (0,1)$ and $k \in \mathbb{N}$.
- Requirement 3. Let A be denoted as Definition 6.5.
- Requirement 4. Let $\lambda_1, \dots, \lambda_n$ denote the eigenvalue of A.
- Requirement 5. Let Δ be denoted as Definition 6.6.
- Requirement 6. Let M, \mathcal{M} be denoted as Definition 6.2 and $M \leq \Delta$.

we have

$$\mathbb{E}[Z] \le \frac{\epsilon}{2}$$

Proof.

$$\mathbb{E}[Z] = \frac{k}{2} \sum_{j=1}^{n} (\lambda_j - 1 - \log(\lambda_j))$$

$$\leq \frac{k}{2} \sum_{j=1}^{n} (\lambda_j - 2 + \frac{1}{\lambda_j})$$

$$= \frac{k}{2} \sum_{j=1}^{n} (\lambda_j - 1)(1 - \frac{1}{\lambda_j})$$

$$\leq \|A - I\|_F \cdot \|I - A^{-1}\|_F$$

$$\leq \frac{k}{2} \Delta^2$$

$$\leq \frac{\epsilon}{2}$$

where the 1st step follows from linearity of expectation and Lemma 6.9, the 2nd step is the result of $\lambda_j > 0$ and $\log(x) > 1 - \frac{1}{x}$ for x > 0, the 3rd step follows from simple factorization, the fourth step follows from Cauchy-Schwarz, the fifth step follows from Lemma 6.8 and **Requirement 6** in the lemma statement, and the last step follows from $\Delta < \frac{\epsilon}{\sqrt{k}}$ and $\epsilon < 1$.

6.6 Sub-Exponential

In Section 6.6, evidence is provided that supports the claim that Z is sub-exponential. This is significant because it enables us to limit the maximum probability of the event $Z \ge \epsilon$, which is crucial in ensuring differential privacy.

Lemma 6.11. If all of the following requirements are met

- Requirement 1. We define Z as Definition 6.3.
- Requirement 2. Let $\epsilon \in (0,1)$ and $\delta \in (0,1)$.
- Requirement 3. Let Δ be denoted as Definition 6.6 and $\Delta < 1$.

- Requirement 4. Let M, M be denoted as Definition 6.2 and $M \leq \Delta$.
- Requirement 5. $k \in \mathbb{N}$.

we have

$$\Pr[Z > \epsilon] \le \delta$$

Proof. First, we will prove Z is sub-exponential.

Proof of Sub Exponential Let A be dented as Definition 6.5 and $h_{i,j}$ be denoted as Definition 6.3.

Since $h_{i,j} \sim \chi_1^2$, Lemma 5.5 and Lemma 5.4, we can say Z is sub-exponential with

- $\nu = \sqrt{k} ||I A||_F$
- $\alpha = 2||I A||_F$

By Lemma 6.8, we have

- $\nu = \sqrt{k} ||A I||_F < \sqrt{k} \Delta$
- $\alpha = 2||A I||_F < 2\Delta$

Proof of Upper Bound for \mathbb{E}[Z]. Under Requirement 3 and Requirement 4, by using Lemma 6.10, we have

$$\mathbb{E}[Z] \le \epsilon/2 \tag{6}$$

Proof of Upper Bound By using Lemma 5.3 (sub-exponential tail bound), we have

$$\begin{aligned} \Pr[Z > \epsilon] < & \Pr[Z - \mathbb{E}[Z] > \epsilon/2] \\ \leq & \max \left\{ \exp(-\frac{(\epsilon/2)^2}{2\nu^2}), \exp(-\frac{\epsilon/2}{2\alpha}) \right\} \\ < & \delta \end{aligned}$$

where the 1st step is the reuslt of Eq. (6), the 2nd step is the reuslt of Lemma 5.3, and the last step follows from Requirement 3 in the lemma statement.

6.7Main Result

This section contains our main result in Section 6, which we present as follows.

Algorithm 1 The Gaussian Sampling Mechanism

- 1: **procedure** ALGORITHM(Σ, k)
- PSD matrix $\Sigma \in \mathbb{R}^{n \times n}$ and parameter $k \in \mathbb{N}$
- Obtain vectors $g_1, g_2, \dots, g_k^{\top}$ by sampling $g_i \sim \mathcal{N}(0, \Sigma)$, independently for each $i \in [k]$ Compute $\widehat{\Sigma} = \frac{1}{k} \sum_{i=1}^{k} g_i g_i^{\top}$ \triangleright This is Covariance estimates 3:
- 4:

▶ This is Covariance estimate.

- return $\widehat{\Sigma}$
- 6: end procedure

Theorem 6.12 (Analysis of the Gaussian Sampling Mechanism, Theorem 5.1 in [AKT⁺22]). If all of the following requirements are met

- Requirement 1. Let $\epsilon \in (0,1)$ and $\delta \in (0,1)$.
- Requirement 2. $k \in \mathbb{N}$.
- Requirement 3. Neighboring datasets $\mathcal{Y}, \mathcal{Y}'$ differ in a single data element.
- Requirement 4. Let Δ be denoted as Definition 6.6 and $\Delta < 1$.
- Requirement 5. Let M, \mathcal{M} be denoted as Definition 6.2 and $M \leq \Delta$.
- Requirement 6. An input $\Sigma = \mathcal{M}(\mathcal{Y})$.
- Requirement 7. $\rho = O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)$.

Then, there exists an algorithm 1 such that

- Part 1. Algorithm 1 is (ϵ, δ) -DP (with respect to the original dataset \mathcal{Y}).
- Part 2. outputs $\widehat{\Sigma} \in \mathbb{S}^n_+$ such that with probabilities at least 1γ ,

$$\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I_n\|_F \le \rho$$

• Part 3.

$$(1-\rho)\Sigma \preceq \widehat{\Sigma} \preceq (1+\rho)\Sigma$$

Proof. We denote Z as Definition 6.3 which is as the output of algorithm 1.

The utility guarantee is immediately implied by Theorem 5.1.

Now, we will focus on the proof of privacy. By Lemma 6.11, we have

$$\Pr[Z > \epsilon] \le \delta \tag{7}$$

And then by Theorem 5.2 and Eq. (7), Algorithm 1 is proved as (ϵ, δ) -differential private. **Proof of Part 3.**

$$\|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I_n\| \le \|\Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} - I_n\|_F$$

$$\le \rho$$

Thus,

$$(1-\rho)I_n \preceq \Sigma^{-1/2}\widehat{\Sigma}\Sigma^{-1/2} \preceq (1+\rho)I_n$$

which is equivalent to

$$(1-\rho)\Sigma \preceq \widehat{\Sigma} \preceq (1+\rho)\Sigma$$

7 Sensitivity for PSD Matrix

In this section, we demonstrate that $\mathcal{M}(X) = XX^{\top}$ satisfies the assumption specified in **Requirement 5** of Theorem 6.12 for $\mathcal{M}(X)$.

Lemma 7.1. If $X \in \mathbb{R}^{n \times d}$ and $\widetilde{X} \in \mathbb{R}^{n \times d}$ are neighboring dataset (see Definition 1.5 and Definition 1.6), then

$$(1 - 2\alpha\beta/\eta)XX^{\top} \preceq \widetilde{X}\widetilde{X}^{\top} \preceq (1 + 2\alpha\beta/\eta)XX^{\top}$$

Proof. Let $i \in [d]$ be index that $X_{*,i}$ and $\widetilde{X}_{*,i}$ are different (See Definition 1.6). We have

$$\begin{split} \widetilde{X}\widetilde{X}^{\top} &= \sum_{j=1}^{d} \widetilde{X}_{*,j} \widetilde{X}_{*,j}^{\top} \\ &= (\sum_{j \in [d] \setminus \{i\}} \widetilde{X}_{*,j} \widetilde{X}_{*,j}^{\top}) + \widetilde{X}_{*,i} \widetilde{X}_{*,i}^{\top} \\ &= (\sum_{j \in [d] \setminus \{i\}} X_{*,j} X_{*,j}^{\top}) + \widetilde{X}_{*,i} \widetilde{X}_{*,i}^{\top} \\ &= X X^{\top} - X_{*,i} X_{*,i}^{\top} + \widetilde{X}_{*,i} \widetilde{X}_{*,i} \end{split}$$

where the first step is the result of matrix multiplication, the second step is from simple algebra, the third step follows from Definition 1.6, and the last step comes from simple algebra.

We know that

$$||X_{*,i}X_{*,i}^{\top} - \widetilde{X}_{*,i}\widetilde{X}_{*,i}|| = ||X_{*,i}X_{*,i}^{\top} - X_{*,i}\widetilde{X}_{*,i}^{\top} + X_{*,i}\widetilde{X}_{*,i}^{\top} - \widetilde{X}_{*,i}\widetilde{X}_{*,i}||$$

$$\leq ||X_{*,i}X_{*,i}^{\top} - X_{*,i}\widetilde{X}_{*,i}^{\top}|| + ||X_{*,i}\widetilde{X}_{*,i}^{\top} - \widetilde{X}_{*,i}\widetilde{X}_{*,i}||$$

$$\leq ||X_{*,i}||_{2} \cdot ||X_{*,i} - \widetilde{X}_{*,i}||_{2} + ||X_{*,i} - \widetilde{X}_{*,i}||_{2} \cdot ||\widetilde{X}_{*,i}||_{2}$$

$$\leq 2\alpha\beta$$
(8)

where the first step is from adding a new term $X_{*,i}\widetilde{X}_{*,i}^{\top}$, the second step follows from the triangle inequality, the third step follows from Fact 3.2, and the last step is due to Definition 1.5 and Definition 1.6.

Thus, we have

$$\widetilde{X}\widetilde{X}^{\top} \succeq XX^{\top} - 2\alpha\beta I_n$$

 $\succeq (1 - 2\alpha\beta/\eta)XX^{\top}$

where the first step is due to Eq. 8, and the second step follows from $XX^{\top} \succeq \eta \cdot I_n$. Similarly, we have

$$\widetilde{X}\widetilde{X}^{\top} \leq XX^{\top} + 2\alpha\beta I_n$$

$$\leq (1 + 2\alpha\beta/\eta)XX^{\top}$$

Lemma 7.2. Let α and β be denoted in Definition 1.5 and Definition 1.6. If X and \widetilde{X} are neighboring datasets such that

$$(1 - 2\alpha\beta/\eta)XX^{\top} \preceq \widetilde{X}\widetilde{X}^{\top} \preceq (1 + 2\alpha\beta/\eta)XX^{\top}$$

then, we have

• Part 1.

$$\|(XX^{\top})^{-1/2}\widetilde{X}\widetilde{X}^{\top}(XX^{\top})^{-1/2} - I\| \le 2\alpha\beta/\eta$$

• Part 2.

$$\|(XX^{\top})^{-1/2}\widetilde{X}\widetilde{X}^{\top}(XX^{\top})^{-1/2} - I\|_F \le 2\sqrt{n}\alpha\beta/\eta$$

Proof. The proof is straightforward, and we omit the details here.

8 Main Result

This section presents the proof of our main result, which is based on the conclusions drawn in Section 4 and Section 6.

Theorem 8.1 (Main result, informal of Theorem 1.7). If all of the following requirements are met

- Let $d \geq n$.
- Let $X \in \mathbb{R}^{n \times d}$.
- We define $r \in (0, 0.1)$ as bounded ratio.
- Let $f(z) \in {\exp(z), \cosh(z)}$.
- Let $\epsilon \in (0, 0.1)$ denote the parameter of DP.
- We define $\delta \in (0,0.1)$ as the parameter of DP.
- Let $\Delta = 0.1 \min\{\frac{\epsilon}{\sqrt{k \log(1/\delta)}}, \frac{\epsilon}{\log(1/\delta)}\}$
- Let $A = \mathcal{M}(X) = XX^{\top}$
- Let $||A||_{\infty} \leq r$
- For all X sampled from \mathcal{D} , X is (α, η) -good (see Definition 1.5).
- Let $\eta < r$.
- Let β be the parameter for neighboring dataset.
- Let $2\alpha\beta\sqrt{n}/\eta < \Delta$
- Let Δ denote the sensitivity parameter that $\mathcal M$ satisfies a sensitivity bound that

$$\|\mathcal{M}(X)^{1/2}\mathcal{M}(\widetilde{X})^{-1}\mathcal{M}(X)^{1/2} - I\|_F \le \Delta$$

for any neighboring datasets $X \in \mathbb{R}^{n \times d}$, $\widetilde{X} \in \mathbb{R}^{n \times d}$ (see Definition 1.6).

- Let $\rho = \sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k$
- Let $\rho < 0.1\epsilon$

Then there exists an algorithm that takes $A = \mathcal{M}(X)$ as inputs, and outputs matrix B such that

- Part 1. $\| D(A)^{-1} f(A) D(B)^{-1} f(B) \|_{\infty} \le 4 \cdot (1 + \epsilon + 2r) \cdot r$
- Part 2. With respect to X, the algorithm is (ϵ, δ) -differential private.
- Part 3. It holds with probability 1γ .

Proof. The proof can be divided into two parts as follows.

Proof of Part 1 and Part 3. Our proof focus on the function $\mathcal{M}(X) := XX^{\top}$ first. Let α and η be denoted in Definition 1.5 and β be denoted as Definition 1.6. Based on the assumption on dataset above, we can obtain X is (η, α) -good (See Definition 1.5) while X and \widetilde{X} are β -close (See Definition 1.6).

According to Part 1 of Lemma 7.2, we can conclude the property on $\mathcal{M}(X) = XX^{\top}$ such that

$$\|(XX^{\top})^{-1/2}\widetilde{X}\widetilde{X}^{\top}(XX^{\top})^{-1/2} - I\|_F \le 2\sqrt{n}\alpha\beta/\eta$$

Let \mathcal{M} be the function denoted in the theorem statement and let ρ be denoted as follows:

$$\rho := O(\sqrt{(n^2 + \log(1/\gamma))/k} + (n^2 + \log(1/\gamma))/k)$$

Now, we will apply the conclusion drawn in Section 6. In order to satisfy the requirement specified in **Requirement 5** of Theorem 6.12, we need $\mathcal{M}(X)$ to meet the following assumption:

$$\|\mathcal{M}(X)^{1/2}\mathcal{M}(\widetilde{X})^{-1}\mathcal{M}(X)^{1/2} - I\|_F \le \Delta.$$

Now, if we choose

$$2\alpha\beta\sqrt{n}/\eta<\Delta,$$

we will guarantee that our $\mathcal{M}(X)$ satisfies the assumption specified in **Requirement 5** of Theorem 4.7. According to **Part 3** of Theorem 4.7, there exists Algorithm 1 which can produce a matrix $B \in \mathbb{R}^{n \times n}$ such that, with probability at least $1 - \gamma$

$$(1 - \rho)A \le B \le (1 + \rho)A \tag{9}$$

By choosing $\rho \in (0,0.1)\epsilon$, we will have

$$(1 - \epsilon)B \le A \le (1 + \epsilon)B \tag{10}$$

Now according to Theorem 4.7 and Eq. (10), we have

$$\| D(A)^{-1} f(A) - D(B)^{-1} f(B) \|_{\infty} \le 4 \cdot (1 + \epsilon + 2r) \cdot r$$

Now, the proofs of **Part 1** and **Part 3** are completed.

Proof of Part 2. It simply follows from **Part 1** of Theorem 6.12

References

- [AKT⁺22] Daniel Alabi, Pravesh K Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. Privately estimating a gaussian: Efficient, robust and optimal. arXiv preprint arXiv:2212.08018, 2022.
 - [AS23] Josh Alman and Zhao Song. Fast attention requires bounded entries. arXiv preprint arXiv:2302.13214, 2023.
 - [BKO18] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079, 2018.
- [BMR⁺20] 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.
 - [Bra20] Jan van den Brand. A deterministic linear program solver in current matrix multiplication time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 259–278. SIAM, 2020.
 - [BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part I*, pages 635–658. Springer, 2016.
 - [BSZ23] Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. arXiv preprint arXiv:2304.02207, 2023.
 - [CLS19] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *STOC*, 2019.
- [DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
- [DKM⁺06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.
 - [DLS23] Yichuan Deng, Zhihang Li, and Zhao Song. Attention scheme inspired softmax regression. arXiv preprint arXiv:2304.10411, 2023.
 - [DLY21] Sally Dong, Yin Tat Lee, and Guanghao Ye. A nearly-linear time algorithm for linear programs with small treewidth: a multiscale representation of robust central path. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1784–1797, 2021.

- [DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference*, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, pages 265–284. Springer, 2006.
 - [DMS23] Yichuan Deng, Sridhar Mahadevan, and Zhao Song. Randomized and deterministic attention sparsification algorithms for over-parameterized feature dimension. arxiv preprint: arxiv 2304.03426, 2023.
- [EMM⁺23] Alessandro Epasto, Jieming Mao, Andres Munoz Medina, Vahab Mirrokni, Sergei Vassilvitskii, and Peilin Zhong. Differentially private continual releases of streaming frequency moment estimations. arXiv preprint arXiv:2301.05605, 2023.
- [ERLD17] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751, 2017.
 - [FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.
 - [GMS23] Yeqi Gao, Sridhar Mahadevan, and Zhao Song. An over-parameterized exponential regression. arXiv preprint arXiv:2303.16504, 2023.
 - [GS22] Yuzhou Gu and Zhao Song. A faster small treewidth sdp solver. arXiv preprint arXiv:2211.06033, 2022.
 - [GSY23] Yeqi Gao, Zhao Song, and Junze Yin. An iterative algorithm for rescaled hyperbolic functions regression. arXiv preprint arXiv:2305.00660, 2023.
- [HWL21] Weihua He, Yongyun Wu, and Xiaohua Li. Attention mechanism for neural machine translation: A survey. In 2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), volume 5, pages 1485–1489. IEEE, 2021.
- [JSWZ21] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. Faster dynamic matrix inverse for faster lps. In *STOC*, 2021.
- [KGW⁺23] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. arXiv preprint arXiv:2301.10226, 2023.
 - [KNK21] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. The Journal of Machine Learning Research, 22(1):1395–1476, 2021.
- [LOG⁺19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.
 - [LSZ19] Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *Conference on Learning Theory (COLT)*, pages 2140–2157. PMLR, 2019.

- [LSZ23] Zhihang Li, Zhao Song, and Tianyi Zhou. Solving regularized exp, cosh and sinh regression problems. arXiv preprint, 2303.15725, 2023.
- [Ope23] OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.
- [QSZZ23] 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, 2023.
 - [RF18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
- [RNS⁺18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [RSR⁺20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.
- [RWC⁺19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
 - [Sag18] Matthew Sag. The new legal landscape for text mining and machine learning. J. Copyright Soc'y USA, 66:291, 2018.
 - [Son19] Zhao Song. Matrix theory: optimization, concentration, and algorithms. The University of Texas at Austin, 2019.
- [TPG⁺17] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. arXiv preprint arXiv:1704.03453, 2017.
- [UAS+20] Mohd Usama, Belal Ahmad, Enmin Song, M Shamim Hossain, Mubarak Alrashoud, and Ghulam Muhammad. Attention-based sentiment analysis using convolutional and recurrent neural network. Future Generation Computer Systems, 113:571–578, 2020.
 - [Vad17] Salil Vadhan. The complexity of differential privacy. Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich, pages 347–450, 2017.
 - [Ver18] Roman Vershynin. High-dimensional probability: An introduction with applications in data science, volume 47. Cambridge university press, 2018.
- [VKB23] Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models. arXiv preprint arXiv:2302.10870, 2023.
- [VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.
 - [Wai19] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

- [YDY⁺19] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.
- [ZHDK23] Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. arXiv preprint arXiv:2302.02451, 2023.
- [ZTL⁺17] Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural slam: Learning to explore with external memory. arXiv preprint arXiv:1706.09520, 2017.