This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3093316, IEEE Transactions on Mobile Computing

Matrix Gaussian Mechanisms for Differentially-Private Learning

Jungang Yang, Liyao Xiang, *Member, IEEE,* Jiahao Yu, Xinbing Wang, *Senior Member, IEEE,* Bin Guo, *Senior Member, IEEE,* Zhetao Li, and Baochun Li, *Fellow, IEEE*

Abstract—The wide deployment of machine learning algorithms has become a severe threat to user data privacy. As the learning data is of high dimensionality and high orders, preserving its privacy is intrinsically hard. Conventional differential privacy mechanisms often incur significant utility decline as they are designed for scalar values from the start. We recognize that it is because conventional approaches do not take the data structural information into account, and fail to provide sufficient privacy or utility. As the main novelty of this work, we propose *Matrix Gaussian Mechanism* (MGM), a new (ϵ, δ)-differential privacy mechanism for preserving learning data privacy. By imposing the unimodal distributions on the noise, we introduce two mechanisms based on MGM with an improved utility. We further show that with the utility space available, the proposed mechanisms can be instantiated with optimized utility, and has a closed-form solution scalable to large-scale problems. We experimentally show that our mechanisms, applied to privacy-preserving federated learning, are superior than the state-of-the-art differential privacy mechanisms in utility.

1 Introduction

Powered by abundant data, machine learning models and APIs have cultivated a variety of applications from object detection, machine translation to auto driving and behavior modeling. For example, the input suggestion system on a smartphone learns the user's language habits locally to give better recommendations. It can also learn from other users' expressions to give more precise suggestion. However, the learning models and APIs pose great threats to user data privacy, as an individual can be easily singled out from a large amount of data records. The situation is even worse in federated learning as participants jointly build a shared model on their local proprietary datasets. For example, through the gradients exchanged in each training iteration, an attacker is able to infer if a data record is in the training dataset. A variety of privacy-preserving techniques such as cryptography-based approaches, differential privacy and others, have been proposed for protecting data privacy in learning.

Differential privacy has become the gold standard of privacy guarantee in general data analytics and processing [1], [2], [3], [4]. Both as a rigorous mathematical concept and privacy-preserving mechanisms, differential privacy constrains an adversary's capability to infer anything about the sensitive data independent of its occurrence in the dataset [5]. Hence, the mechanism prevents any adversary from gaining additional information about any single record. The technique has been integerated with machine learning to prevent the released model/gradients from leaking out private training data [6], [7], [8].

However, the current differential privacy mechanisms for learning algorithms have not taken the characteristics of learning data into account. The learning data not only includes the training or testing inputs, but also the intermediate-layer features, gradients and model parameters. Since machine learning, in particular deep learning, involves learning data of high dimensionality or high orders, conventional differential privacy mechanisms typically introduce great amount of randomized noise, resulting in notable accuracy degradation of the model. Moreover, merely treating the learning data as a collection of scalar-valued elements, or flattening them into vectors do not preserve the structural information, leading to loss of utility as well. Hence there is a growing need to formalize privacy guarantee on the learning data, especially relieving the tension between the privacy guarantee and the model utility.

Preserving utility and privacy at the same time for high-dimensional/order data is difficult. This originates from the fundamental trade-off between data privacy and utility. For matrix-valued data, the problem is more severe as an overwhelming amount of noise may be inserted, leading to less useful data. Practical schemes have been proposed to alleviate such loss, as in [9], [10], [11], [12]. As most of the solutions are heuristic, there are no utility guaranteed, nor scalable approaches to large-scale applications.

In this work, we formalize the study of differential privacy for matrix-valued data and propose a mechanism called *Matrix Gaussian Mechanism* (MGM). Preserving data's original structure, MGM adds differentially-private, matrix-valued noise to the data. The idea is to utilize the matrix Gaussian distribution to guarantee (ϵ, δ)-differential privacy, and such a guarantee only depends on the covariance matrices of the noise. We rigorously prove that MGM meets differential privacy, and more importantly show it has a tighter noise bound, in light of which higher utility than previous works can be achieved. Actually, MGM implies a

*Jungang Yang and Liyao Xiang are the cofirst authors. Jungang Yang, Liyao Xiang (xiangliyao08@sjtu.edu.cn, the corresponding author), Jiahao Yu, and Xinbing Wang, are with Shanghai Jiao Tong University, China. Bin Guo is with Northwestern Polytechnical University (NPU), China. Zhetao Li is with Xiangtan University, China. Baochun Li is with the University of Toronto, Canada.

*This work was partially supported by NSF China (61902245, 62032020, 61960206002, 61822206, 62020106005, 61829201, 62041205, 61532012), National Key R&D Program of China 2018YFB1004700, and the Science and Technology Innovation Program of Shanghai (19YF1424500).

set of mechanisms, which leaves much design space for us to manipulate for achieving better data utility. By additionally imposing some restrictions on the noise, we further propose two mechanisms based on MGM with improved utility.

From the utility aspect, we observe different directions of the learning data may have different impacts to the output, depending on how the data is involved in the task. For example, for parameterized models, an attribute may be more important to the final result given it is associated with more significant model weights. We can improve the performance of such tasks by adding carefully-crafted directional noise, *i.e.*, a structural noise which incur minimum impact on the final result. Hence we propose an optimization framework for MGM to seek an optimal noise direction in terms of utility. Closed-form solutions are derived, rendering MGM readily be deployed in large-scale problems.

Based on MGM, we devise two practical schemes for privacy-preserving federated learning over sensitive datasets. In federated learning, the gradients and features exchanged in each training iteration pose as great threats to data privacy. We apply MGM respectively to gradients and features in different federated learning scenarios. We show the superiority of MGM over state-of-the-art differential privacy mechanisms both from theoretical and experimental perspectives.

Highlights of our contribution include: first, we propose a (ϵ, δ)-differential privacy mechanism MGM for matrix-valued data, which enjoys higher utility over previous works. Second, a utility optimization framework is introduced based on MGM, of which closed-form solutions are derived. Finally, we derive two practical privacy-preserving schemes for federated learning, and experimental results support that MGM is scalable, practical, and enjoys higher utility than previous works.

2 Related Work

Our work is mostly related to works in the following categories.

2.1 Primitive Mechanisms

Primitive mechanisms refer to those whose privacy guarantee is self-contained, *i.e.,* it does not depend on any other mechanism. They include Gaussian mechanism [13], Laplace mechanism [14], Exponential mechanism [13], Matrix Variate Gaussian (MVG) [11], Matrix Mechanism (MM) [10], etc.

Our work is related to the additive noise mechanisms such as Gaussian, Laplace, MVG, and MM. The Gaussian mechanism applies i.i.d. Gaussian noise scaled to the l_2-sensitivity and guarantees (ϵ, δ)-differential privacy. [15] improves the conventional Gaussian mechanism by using a necessary and sufficient condition rather than a sufficient condition for vectorized queries. Likewise, the Laplace mechanism adds noise drawn from the Laplace distribution scaled to the l_1-sensitivity of the query function, and guarantees strong ϵ-differential privacy. MM is designed for linear queries, which adds a vector-valued noise (Gaussian or Laplace) to the data and seeks an optimal transform to minimize the perturbation impact. MVG is proposed for matrix-valued queries, and adds matrix-valued noise to guarantee (ϵ, δ)-differential privacy. Belonging to the additive noise mechanism, our work proposes a (ϵ, δ)-differential privacy mechanism for matrix-valued queries with utility guarantees.

Our work is also aligned with works addressing the utility of additive noise such as [9], [11], [12]. The optimal noise distribution is found by Geng *et al.* [9] in terms of the magnitude of the noise, but has restriction on data dimensions. Similar to [12], we formulate the problem of seeking the optimal noise distribution as a constrained optimization problem, and such a distribution in fact indicates directional noise as introduced in [11].

2.2 Learning with Differential Privacy

There is a wide range of works applying differential privacy mechanisms to machine learning algorithms [2], [3], [16], [17]. Depending on different privacy-preserving goals, we have differentially-private inputs [11], [18], [19], [20], outputs [4], [8], [21], gradients [6], [7], [22], [23], [24], [25], [26], [27], [28], and objective functions [29], [30], [31], etc.

On the basis of privacy-preserving learning methods, many architectures have been proposed to build models on the sensitive training data. Shokri *et al.* [6] introduce a practical system for federated learning which allows multiple participants to learn neural network models by sharing selective parameters in a differentially-private way. Triastcyn *et al.* [32] apply Bayesian differential privacy (BDP) in federated learning to get a tighter privacy guarantee. Different from the centralized differential privacy used in previous works, Truex *et al.* [33] adopt local differential privacy in the federated learning and further reduce the impact of noise by proposing α-CLDP. Wei *et al.* [34] propose noising before model aggregation FL (NbAFL) mechanism and prove that NbAFL could still guarantee (ϵ, δ)-differential privacy as the variances of noise vary. Different from these mechanisms, we minimize the impact of the noise by designing the noise distribution. Therefore, our mechanism has a broader application range as it fits the noise distribution to different scenarios.

We pay particular attention to the federated learning applications as the models are typically trained on the sensitive datasets of different participants. Our mechanism can be considered as a primitive differential privacy mechanism which can be applied to the matrix-valued inputs, outputs or gradients in machine learning algorithms.

3 Preliminaries

In this section, we prepare the readers with prior knowledge for ease of understanding our work.

3.1 Differenital Privacy

Differential privacy is proposed to constrain an attacker's capability to gain additional knowledge about a particular data record despite that it is in the dataset or not. The privacy guarantee is expressed by the logarithmic distance between the posterior probability distributions of two adjacent inputs given the outputs. Adjacent inputs are defined on two datasets differ by one unit of distance. Different

TABLE 1
Notations

U	a matrix
N	Standard Normal Distribution variable
$s_2(f)$	l_2-sensitivity of $f(X)$ (Def. 3)
$\mathcal{MN}_{m,n}\,(M,\Sigma_1,\Sigma_2)$	Matrix Gaussian distribution (Def. 4)
M	mean of $\mathcal{MN}_{m,n}$
Σ_k	covariance matrix of $\mathcal{MN}_{m,n},\ k=1,2$
U_k	$U_kU_k^\top=\Sigma_k,\ k=1,2$
$\epsilon,\ \delta$	parameters of differential privacy
$\zeta(\delta)^2$	$-2\ln\delta+2\sqrt{-mn\ln\delta}+mn$
α	$s_2^2(f)$
β	$2\zeta(\delta)s_2(f)$
B	$\left(-\beta+\sqrt{\beta^2+8\alpha\epsilon}\right)^2/4\alpha^2,$
W_{U_k},S_k	the SVD of $\Sigma_k=W_{U_k}S_kW_{U_k}^\top,\ k=1,2$
S_{U_k}	$S_{U_k}S_{U_k}^\top=S_k,\ k=1,2$
$\mathbf{E}_1,\ \mathbf{E}_2$	the identity matrix of dimension m and n
W_1,W_2	the utility subspaces

metrics of the distance can be used, which leads to different variants of differential privacy. We use ϵ to define the upper bound of the distribution distance and δ to denote the residual probability. Formally, letting X and X' be the pair of adjacent inputs, $\mathcal{O}$ be the output set and $\mathbf{M}$ be the private mechanism, we have

Definition 1 ((ϵ,δ)-Differential Privacy)**.** *A randomized mechanism $\mathbf{M}$ gives (ϵ,δ)-differential privacy if for any datasets X and X' differing by at most one unit, and for any possible output $\mathcal{O}$,*

$$\Pr(\mathbf{M}(X)\in\mathcal{O})\le e^\epsilon\Pr(\mathbf{M}(X')\in\mathcal{O})+\delta. \tag{1}$$

In the special case of $\delta=0$ we call $\mathbf{M}$ ϵ-differentially private. The definition above does not specify the concept of adjacent datasets, which is typically defined application-wise. By default in this paper, we refer to a pair of datasets differing by a single record as adjacent datasets.

3.2 Relevant Definitions and Lemmas

As we mainly focus on matrix-valued random variable, we clarify some of the related definitions and lemmas adopted in this paper.

Definition 2 (Standard Normal Distribution (SND))**.** *If a matrix-valued random variable $N\in\mathbb{R}^{m\times n}$ follows a standard normal distribution (SND), its probability density function is*

$$\Pr(N)=\frac{1}{(2\pi)^{mn/2}}\exp\left\{-\frac{1}{2}\|N\|_F^2\right\}, \tag{2}$$

where $\|\cdot\|_F$ denotes frobenius norm.

Note that if a matrix-valued random variable follows SND, each element of the matrix N_{ij} follows the normal distribution $\mathcal{N}(0,1)$. Besides the above definitions, we introduce some related lemmas concerning our work.

Lemma 1 (Matrix norm inequality)**.** *Let $U_1\in\mathbb{R}^{m\times m}$, $U_2\in\mathbb{R}^{n\times n}$, and $X,Y\in\mathbb{R}^{m\times n}$ be a pair of matrices which satisfy: $X=U_1YU_2$. Then we have the matrix norm inequality that*

$$\|X\|_F\le\|Y\|_F\|U_1\|_F\|U_2\|_F. \tag{3}$$

Lemma 2 (The bound of the SND matrix-valued random variable)**.** *For a matrix-valued random variable $N\in\mathbb{R}^{m\times n}$ following the SND, $\delta\in[0,1)$ and $\zeta(\delta)^2=-2\ln\delta+2\sqrt{mn\ln\delta}+mn$, we have*

$$\Pr[\|N\|_F^2\le\zeta(\delta)^2]\ge1-\delta. \tag{4}$$

We list the notations used in this paper in Table 1 for ease of reading.

4 Matrix Gaussian Mechanism

In this section, we first introduce the matrix-valued differential privacy mechanism called Matrix Gaussian Mechanism (MGM), and give the main theorem along with its proof. The matrix-valued differential privacy mechanism is mostly different from the scalar-valued one in that the data or query are in matrix form, and we need to guarantee differential privacy regardless of the specific shape of the matrix.

For a more fluent narrative of our mechanism, we first define l_2-sensitivity on a pair of adjacent matrices as follows:

Definition 3 (l_2-sensitivity)**.** *The l_2-sensitivity of the query function $f(X)\in\mathbb{R}^{m\times n}$ is defined as,*

$$s_2(f)=\sup_{d(X,X')=1}\|f(X)-f(X')\|_F,$$

where $\|\cdot\|_F$ is the Frobenius norm, $d(X,X')=1$ means that X and X' are neighboring datasets differing by only a single record.

Based on the standard normal distribution on matrices (Def. 2), we define matrix Gaussian distribution and its variable $\mathcal{Z}$ as below:

Definition 4 (Matrix Gaussian distribution)**.** *The probability density function for the $m\times n$ matrix-valued random variables $\mathcal{Z}$ which follows the matrix normal distribution $\mathcal{MN}_{m,n}(M,\Sigma_1,\Sigma_2)$ is*

$$\Pr(\mathcal{Z}|M,\Sigma_1,\Sigma_2)=\frac{\exp\{-\frac{1}{2}\|U_1^{-1}(\mathcal{Z}-M)U_2^{-\top}\|_F^2\}}{(2\pi)^{(mn)/2}|\Sigma_2|^{n/2}|\Sigma_1|^{m/2}}, \tag{5}$$

where $U_kU_k^\top=\Sigma_k,\ k=1,2,\ |\cdot|$ is the matrix determinant, $M\in\mathbb{R}^{m\times n}$ is mean, $\Sigma_1\in\mathbb{R}^{m\times m}$ is the row-wise covariance and $\Sigma_2\in\mathbb{R}^{n\times n}$ is the column-wise covariance.

Note that if $N\in\mathbb{R}^{m\times n}$ is SND random variable defined by Eq. 2, then we could get that $N=U_1^{-1}(\mathcal{Z}-M)U_2^{-\top}$, and N is a special case of matrix Gaussian random variable that $N\sim\mathcal{MN}_{m,n}(\mathbf{0},\mathbf{E}_1,\mathbf{E}_2)$. With the above definition, we apply additive matrix Gaussian noise following $\mathcal{MN}_{m,n}(\mathbf{0},\Sigma_1,\Sigma_2)$ distribution in the MGM mechanism stated in the following.

Definition 5 (Matrix Gaussian Mechanism)**.** *For a given query function $f(X)\in\mathbb{R}^{m\times n}$ and a matrix variate Gaussian $Z\sim\mathcal{MN}_{m,n}(\mathbf{0},\Sigma_1,\Sigma_2)$ the Matrix Gaussian Mechanism is defined as:*

$$\mathbf{MGM}(f(X))=f(X)+\mathcal{Z}. \tag{6}$$

Similar to the Gaussian mechanism [13], MGM adds zero-mean randomized noise to the query result. Note that $\Sigma_1\in\mathbb{R}^{m\times m}$ and $\Sigma_2\in\mathbb{R}^{n\times n}$ are the covariance matrices of different modes for Z, which are subject to design. In our main theorem to be discussed, we mainly show what forms of the covariance matrices would ensure the mechanism to be differentially-private.

By Lemma 1 and 2, we can prove the main theorem on MGM defined in Def. 5:

Theorem 1. *We have a query function $f(X)\in\mathbb{R}^{m\times n}$, and a matrix Gaussian noise $\mathcal{Z}\sim\mathcal{MN}_{m,n}(\mathbf{0},\Sigma_1,\Sigma_2)$. Σ_1,Σ_2 are*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3093316, IEEE Transactions on Mobile Computing

the covariance matrices and $U_1 \in \mathbb{R}^{m\times m}, U_2 \in \mathbb{R}^{n\times n}$ satisfies $U_k U_k^\top = \Sigma_k$ for each $k = 1, 2$. The MGM guarantees (ϵ, δ)-differential privacy if U_1, U_2 satisfy

$$\|U_1^{-1}\|_F^2 \|U_2^{-1}\|_F^2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4\alpha^2}. \tag{7}$$

$\alpha = s_2^2(f)$, and $\beta = 2\zeta(\delta)s_2(f)$, where $s_2(f)$ is the l_2-sensitivity of $f(X)$ and $\zeta(\delta)$ is defined in Lemma 2.

Note that the right side of Eq. (7) is a constant once the privacy parameters ϵ and δ are given. Hence the theorem shows that to guarantee differential privacy for matrix-valued data, one only needs to satisfy the constraint on Frobenius norms concerning covariance matrices of the additive noise Z. We include the full proof here

Proof. By Def. 1, to guarantee (ϵ, δ)-differential privacy, we should have the following satisfied for each pair of datasets X, X' and any possible output set $\mathcal{O}$:

$$\Pr(f(X) + \mathcal{Z} \in \mathcal{O}) \leq e^\epsilon \cdot \Pr(f(X') + \mathcal{Z} \in \mathcal{O}) + \delta,$$

which can be rewritten as

$$\Pr(\mathcal{Z} \in \mathcal{O} - f(X)) \leq e^\epsilon \cdot \Pr(\mathcal{Z} \in \mathcal{O} - f(X')) + \delta.$$

We express $\mathcal{Z}$ in terms of a SND matrix by Eq. (2) and define the following events:

$$\mathbf{R}_1 = \{N : \|N\|_F^2 \leq \zeta^2(\delta)\}, \mathbf{R}_2 = \{N : \|N\|_F^2 > \zeta^2(\delta)\},$$

where $\zeta^2(\delta)$ is defined in Lemma 2. By the definition of $\zeta^2(\delta)$ and Lemma 2, we have $\Pr(\{\mathcal{Z} \in \mathcal{O} - f(X)\} \cap \mathbf{R}_2) \leq \Pr(\mathbf{R}_2) \leq \delta$. And thus we only needs to find the sufficient conditions for the following inequality to hold:

$$\Pr(\{\mathcal{Z} \in \mathcal{O} - f(X)\} \cap \mathbf{R}_1) \leq e^\epsilon \cdot \Pr(\{\mathcal{Z} \in \mathcal{O} - f(X')\} \cap \mathbf{R}_1).$$

Letting $\mathcal{O}' = \mathcal{O} - f(X)$ and $\Delta = f(X) - f(X')$, with the Def. 3 of l_2-sensitivity $\|\Delta\|_F = s_2(f)$. We find that

$$\begin{aligned}
&\Pr(\mathcal{Z} \in \mathcal{O}' \cap \mathbf{R}_1) \leq e^\epsilon \cdot \Pr(\mathcal{Z} \in (\mathcal{O}' + \Delta) \cap \mathbf{R}_1) \\
\Leftrightarrow &\frac{\int_{\mathcal{O}' \cap \mathbf{R}_1} \exp(-\frac{1}{2}\|U_1^{-1}\mathcal{Z}U_2^{-\top}\|_F^2) d\mathcal{Z}}{\int_{(\mathcal{O}'+\Delta)\cap \mathbf{R}_1} \exp(-\frac{1}{2}\|U_1^{-1}\mathcal{Z}U_2^{-\top}\|_F^2) d\mathcal{Z}} \leq e^\epsilon.
\end{aligned}$$

The equation will have to hold for any output set $\mathcal{O}'$ for the differential privacy condition to hold true. Therefore, we could choose $\mathcal{O}'$ as an arbitrary point. And if the condition holds for any point, then it will hold for any output set $\mathcal{O}'$.

$$\begin{aligned}
&\Pr(\mathcal{Z} \in \mathcal{O}' \cap \mathbf{R}_1) \leq e^\epsilon \cdot \Pr(\mathcal{Z} \in (\mathcal{O}' + \Delta) \cap \mathbf{R}_1) \\
\Leftrightarrow &\frac{\exp(-\frac{1}{2}\|U_1^{-1}\mathcal{Z}U_2^{-\top}\|_F^2)}{\exp(-\frac{1}{2}\|U_1^{-1}(\mathcal{Z}+\Delta)U_2^{-\top}\|_F^2)} \leq e^\epsilon \\
\Leftrightarrow &\frac{1}{2}\|U_1^{-1}(\mathcal{Z}+\Delta)U_2^{-\top}\|_F^2 - \frac{1}{2}\|U_1^{-1}\mathcal{Z}U_2^{-\top}\|_F^2 \leq \epsilon, \\
\Leftrightarrow &\frac{1}{2}\|\Delta'\|^2 + \mathrm{vec}(\Delta')^\top \mathrm{vec}(Q') \leq \epsilon,
\end{aligned}$$

where $\Delta' = U_1^{-1}\Delta U_2^{-\top}$, $Q' = U_1^{-1}\mathcal{Z}U_2^{-\top}, \forall\ \mathcal{Z} \in \mathcal{O}' \cap \mathbf{R}_1$, and $\mathrm{vec}(\Delta')$ denotes the vectorization of Δ', which transforms the matrix into a column vector. It is obvious that the last inequality consists of two parts and we will prove the bound for each.

For conciseness, we define $\phi = \|U_1^{-1}\|_F \|U_2^{-1}\|_F$. By Lemma 1, it can be proved that the first part satisfies

$$\begin{aligned}
\|\Delta'\|_F^2 &= \|U_1^{-1}\Delta U_2^{-\top}\|_F^2 &\text{(8a)}\\
&\leq \|\Delta\|_F^2 \|U_1^{-1}\|_F^2 \|U_2^{-\top}\|_F^2 \leq s_2^2(f)\phi^2. &\text{(8b)}
\end{aligned}$$

With the definition of Q' and $U_1^{-1}\mathcal{Z}U_2^{-\top} = N$ from Def. 2, we get $\|Q'\|_F^2 \leq \zeta^2(\delta)$. Similarly, we derive the bound for the second part:

$$\mathrm{vec}(\Delta')^\top \mathrm{vec}(Q') \leq \|\Delta'\|_F \|Q'\|_F \leq s_2(f)\zeta(\delta)\phi. \tag{9}$$

The first inequality is due to the Cauchy inequality, and the second inequality is similar to Eq. (8a)(8b). By combining the results of two parts, the sufficient condition is

$$s_2(f)^2\phi^2 + 2s_2(f)\zeta(\delta)\phi \leq 2\epsilon. \tag{10}$$

Note that ϕ can only be non-negative. By solving inequality Eq. (10), we have

$$\phi \leq \frac{-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}}{2\alpha},$$

where $\alpha = s_2^2(f),\ \beta = 2s_2(f)\zeta(\delta)$. And this is exactly the noise bound in Thm. 1. □

Theorem 1 gives the condition that MGM should hold for satisfying (ϵ, δ)-differential privacy. It is obvious that this condition is only related to the covariance matrices of the additive noise, which leaves much space for designing the specific covariance matrices and the noise. In the next section, we will introduce mechanisms with careful consideration of the design space.

5 Unimodal Gaussian Noise

In this section, we present two special forms of MGM where noise bounds can be further improved, and thus a better utility can be achieved at the same privacy guarantee.

5.1 Unimodal Directional Noise

We first show an improvement over the general MGM by adding unimodal directional noise (UDN). W.l.o.g., we assume row of $\mathcal{Z}$ is the directional noise and the column are i.i.d., *i.e.,* $U_2 = \mathbf{E}_2$ ($\mathbf{E}_2$ represents the identity matrix with the same shape of U_2). Note that the result is not a special case of Thm. 1. We derive a new bound which is n times tighter than that of Thm. 1.

We further use a toy example in Fig. 1 to clarify the point. Fig. 1 shows a matrix-valued noise decomposed into two vectors, which are respectively row and column vectors. The lower figure describes the matrix noise projected on row and column. The yellow dots represent the SND matrix-valued random variable N whereas the blue dots denote $\mathcal{Z} = U_1 N$. In practice, such unimodal noise can be generated by applying the unimodal direction to the SND matrix, which only changes the row-wise noise.

Now we see how the unimodal directional noise can improve the noise bound. For clear comparison, we first state a direct extension of Thm. 1 to UDN. By directly substituting $U_2 = \mathbf{E}_2$ to the left-hand side of the inequality (7), we get

$$\|U_1{}^{-1}\|_F^2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4n\alpha^2}. \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3093316, IEEE Transactions on Mobile Computing

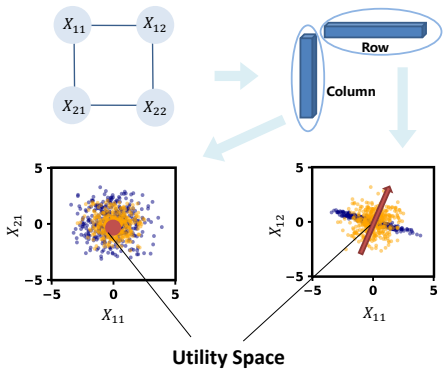

Fig. 1. The diagram of unimodal directional noise. Upper: a matrix $\mathcal{Z} \in \mathbb{R}^{2\times 2}$. Lower: the two subfigures are projections of the same matrix in row and column. The yellow dots represent a SND variable N with all its dimensions being i.i.d. The blue dots denote a variable $\mathcal{Z} = U_1 N$ with row being directional noise and the column being i.i.d. The distribution of its row is decided by the covariance matrix $\Sigma_1 = U_1 U_1^\top$. The red line (or dot) indicates the utility subsapce of the data.

By our new theorem, the right-hand side bound can be improved by n:

Theorem 2 (Unimodal Directional Noise)**.** *Given $U_2 = \mathbf{E}_2$, MGM guarantees (ϵ, δ)-differential privacy if*

$$\|{U_1}^{-1}\|_F^2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4\alpha^2}. \tag{12}$$

where $\alpha = s_2^2(f)$, and $\beta = 2\zeta(\delta)s_2(f)$.

The proof is similar to that of Thm. 1 except that we replace U_2 with $\mathbf{E}_2$ in (8a).

It is clear that the theorem presents that a tighter noise bound for the covariance matrices of the noise. Since the differential privacy condition only depends on U_1 here, we are only required to meet the constraint of Eq. (12). This also suggests room to design U_1 specific to the application.

Note that the covariance matrix $\Sigma_k = U_k U_k^\top$ and the singular value decomposition (SVD) of Σ_k is $\Sigma_k = W_{U_k} S_k W_{U_k}^\top$. We set $U_k = W_{U_k} S_{U_k}$ where $S_{U_k} S_{U_k}^\top = S_k$. Observing that in designing U_1, we have the freedom to substitute any unitary matrix W_{U_1} into the SVD of U_1.

5.2 Independent Directional Noise

The independent directional noise (IDN) follows the setting from UDN and set the row-wise noise $W_{U_1} = \mathbf{E}_1$. W.l.o.g., we assume U_1 is a diagonal matrix and $U_2 = \mathbf{E}_2$. Moreover, we assume that the data to be protected can be scaled to the same range, *i.e.*, each element of $f(X)$ is in range $[a, b]$, then we can improve the bound for $\|U_1^{-1}\|_F$ with the following theorem.

Theorem 3 (Independent Directional Noise)**.** *Let $\mathcal{Z} \sim \mathcal{MN}_{m,n}(\mathbf{0}, \Sigma_1, \mathbf{E}_2) \in \mathbb{R}^{m\times n}$ where $\Sigma_1 = U_1 U_1^\top$ and $U_1 = diag[\sigma_1, ..., \sigma_m] \in \mathbb{R}^{m\times m}$. If we normaize each element of $f(X)$ to the same range $[a, b]$, $\mathbf{MGM}(f(X)) = f(X) + \mathcal{Z}$ guarantees (ϵ, δ)-differential privacy if*

$$\|U_1^{-1}\|_F^2 \leq \frac{m}{\hat{s}_2^2(f)} \left(-\zeta(\delta) + \sqrt{\zeta^2(\delta) + 2\epsilon}\right)^2, \tag{13}$$

with $\zeta(\delta)$ defined in the Lemma 2 and $\hat{s}_2(f) = (b-a)\sqrt{mn}$.

Proof. The proof is similar to the proof of Thm. 1. We define the set of events $\mathbf{R}_1$ and $\mathbf{R}_2$ as in Thm. 1. And we will focus on the sufficient condition of

$$\Pr(\mathcal{Z} \in \mathcal{O}' \cap \mathbf{R}_1) \leq e^\epsilon \cdot \Pr(\mathcal{Z} \in (\mathcal{O}' + \Delta) \cap \mathbf{R}_1). \tag{14}$$

Since Σ_1 is a diagonal matrix, the matrix-valued random variable $\mathcal{Z} \sim \mathcal{MN}_{m,n}(0, \Sigma_1, \mathbf{E}_2)$ and can be expressed as

$$\mathcal{Z} = U_1 N.$$

By substituting the pdf of $\mathcal{Z}$ into the inequality (14), we obtain

$$\Pr(\mathcal{Z} \in \mathcal{O}' \cap \mathbf{R}_1) \leq e^\epsilon \cdot \Pr(\mathcal{Z} \in (\mathcal{O}' + \Delta) \cap \mathbf{R}_1)$$
$$\Leftrightarrow \sum_{i=1}^m \frac{1}{2\sigma_i^2} \left(\sum_{j=1}^n \Delta_{ij}^2 + 2\Delta_{ij} z_{ij}\right) \leq \epsilon.$$

By the definition of neighboring datasets, Δ is a matrix with only one row of nonzero values. W.l.o.g., we suppose that the k^{th} column is nonzero. The left-hand side of the last inequality can be written as

$$\sum_{i=1}^m \frac{\sum_{j=1}^n \Delta_{ij}^2 + 2\Delta_{ij} z_{ij}}{2\sigma_i^2} = \sum_{i=1}^m \frac{\Delta_{ik}^2}{2\sigma_i^2} + \sum_{i=1}^m \frac{\Delta_{ik} z_{ik}}{\sigma_i^2}.$$

We bound the two parts respectively in the last equation. Considering we normalize each feature to the same range, we have $0 \leq \Delta_{ij}^2 \leq \frac{\hat{s}_2^2(f)}{m}$ for every i and j. Hence we have

$$\sum_{i=1}^m \frac{\Delta_{ik}^2}{2\sigma_i^2} \leq \sum_{i=1}^m \frac{\hat{s}_2^2(f)}{2m\sigma_i^2} = \frac{\hat{s}_2^2(f)}{2m} \|U_1^{-1}\|_F^2. \tag{15}$$

For the second part, we rewrite $\mathcal{Z}$ as $U_1 N$ and use $U_1 = diag[\sigma_1, ..., \sigma_m] \in \mathbb{R}^{m\times m}$, we represent each entry of $\mathcal{Z}$ as

$$z_{ij} = \sqrt{\sigma_i} x_{ik},\ x_{ik} \sim \mathcal{N}(0, 1),\ \forall i \in [m],\ \text{and } j \in [n].$$

Then, the second part could be written as

$$\sum_{i=1}^m \frac{\Delta_{ik} z_{ik}}{\sigma_i} = \sum_{i=1}^m \frac{\Delta_{ik} x_{ik}}{\sqrt{\sigma_i}} \leq \sqrt{\sum_{i=1}^m \frac{\Delta_{ik}^2}{\sigma_i}} \sqrt{\sum_{i=1}^m x_{ik}^2}.$$

The inequality is by Cauchy inequality to single out x_{ik}.

According to the definition of $\mathbf{R}_1$, we know that if $N \in \mathbf{R}_1$, $\|N\|_F^2 \leq \zeta^2(\delta)$. Hence,

$$\sum_{i=1}^m x_{ik}^2 \leq \sum_{i=1}^m \sum_{j=1}^n x_{ij}^2 = \|N\|_F^2 \leq \zeta^2(\delta).$$

Thus we have the following inequality holds:

$$\frac{\hat{s}_2^2(f)}{2m} \|U_1^{-1}\|_F^2 + \frac{\hat{s}_2(f)}{\sqrt{m}} \zeta(\delta) \|U_1^{-1}\|_F^2 \leq \epsilon.$$

by the inequality (15).

This is a quadratic inequality of $\|U_1^{-1}\|_F^2$, and with the condition $\|U_1^{-1}\|_F^2 > 0$, which can be solved by

$$\|U_1^{-1}\|_F^2 \leq \frac{m}{\hat{s}_2^2(f)} (-\zeta(\delta) + \sqrt{\zeta^2(\delta) + 2\epsilon})^2,$$

which completes the proof. □

To see how the result compares with that of unimodal directional noise, we only need to replace U_1 as a diagonal matrix in Thm. 2. Thus we have

$$\|U_1^{-1}\|_F^2 \leq \frac{1}{\hat{s}_2^2(f)}\left(-\zeta(\delta)+\sqrt{\zeta^2(\delta)+2\epsilon}\right)^2.$$

Obviously, we have improved the noise bound by m times.

6 Privacy and Utility

We show in this section that MGM has a natural form to be optimized w.r.t. the utility subspace. If we know some dimensions/directions of the matrix are more important than others, we can select noise directions/distributions such that less noise is inserted to the more important part of the matrix, at the cost of a higher level of noise adding to the less important part. By treating the MGM theorem as a differentially-private constraint, we formulate the problem from an optimization perspective. A closed-form solution is obtained which minimizes the total impact of the noise on the output, and the form of the solution is scalable to large-size problems. Based on the solutions, two practical noise generation algorithms are proposed.

Assume the task has linear utility subspace such that:

$$Y = W_1 f(X) W_2^\top,$$

where $f(X) \in \mathbb{R}^{m\times n}$, $W_1 \in \mathbb{R}^{m'\times m}$, $W_2 \in \mathbb{R}^{n'\times n}$ represents the utility subspace. $Y \in \mathbb{R}^{m'\times n'}$ is the output. To preserve privacy, we apply MGM by sampling a noise $\mathcal{Z} \sim \mathcal{MN}_{m,n}(\mathbf{0}, \Sigma_1, \Sigma_2)$ and make predictions on the perturbed query result:

$$\hat{Y} = W_1[f(X) + \mathcal{Z}]W_2^\top.$$

We generate the noise $\mathcal{Z}$ from N such that

$$\mathcal{Z} = U_1 N U_2^\top.$$

According to [9], [10] and other mechanisms, we define our objective as the error on the original query result, measured by the expected norm of weighted noise. If the error is minimized, it means less perturbation is done to the output. The goal is to minimize:

$$\min_{U_1,U_2} \mathbb{E}\|Y-\hat{Y}\|_F^2 \Leftrightarrow \min_{U_1,U_2} \mathbb{E}\|W_1U_1NU_2^\top W_2^\top\|_F^2. \quad (16)$$

In order to estimate the MGM error, we first need to prove the following lemma:

Lemma 3. *Suppose that A is a matrix valued variable with the size $m \times m$, then*

$$\mathbb{E}\left(\boldsymbol{tr}A\right) = \boldsymbol{tr}\left(\mathbb{E}A\right),$$

where $\boldsymbol{tr}A$ represents the trace of A.

Proof. We find that

$$\mathbb{E}\left(\boldsymbol{tr}A\right) = \mathbb{E}(\sum_{i=1}^m A_{ii}) = \sum_{i=1}^m \mathbb{E}A_{ii},$$

$$\boldsymbol{tr}\left(\mathbb{E}A\right) = \boldsymbol{tr}\left(\mathbb{E}A_{ij}\right)_{m\times m} = \sum_{i=1}^m \mathbb{E}A_{ii}.$$

Therefore, $\mathbb{E}\left(\boldsymbol{tr}A\right) = \boldsymbol{tr}\left(\mathbb{E}A\right)$. □

With the above lemma, we could present the theorem of calculating the expection of $\|\mathcal{Z}\|_F^2$.

Theorem 4. *For the given noise*

$$\mathcal{Z} = U_1 N U_2^\top \in \mathbb{R}^{m\times n}, \quad (17)$$

where $N \in \mathbb{R}^{m\times n}$ is a SND noise from Def 2, we have

$$\mathbb{E}\,\|\mathcal{Z}\|^2 = \|U_1\|_F^2\|U_2\|_F^2. \quad (18)$$

Proof. First, we obtain

$$\mathbb{E}\,\|\mathcal{Z}\|_F^2 = \mathbb{E}\,\|U_1NU_2^\top\|_F^2 = \mathbb{E}\,\boldsymbol{tr}(U_1NU_2^\top(U_1NU_2^\top)^\top) \quad (19)$$

Then, with the Lemma. 3, we could switch the trace and the expectation. Thus

$$\begin{aligned}\mathbb{E}\,\|\mathcal{Z}\|_F^2 &= \boldsymbol{tr}(\mathbb{E}\,U_1NU_2^\top U_2N^\top U_1^\top)\\ &=\boldsymbol{tr}(U_1\mathbb{E}\,[NU_2^\top U_2N^\top]U_1^\top).\end{aligned} \quad (20)$$

Hence we focus on the $\mathbb{E}\,[NU_2^\top U_2N^\top]$. Assume that $N^\top = (\boldsymbol{n}_1, \boldsymbol{n}_2, \cdots, \boldsymbol{n}_m)$, we have

$$\left(NU_2^\top U_2N^\top\right)_{ij} = \boldsymbol{n}_i^\top U_2^\top U_2 \boldsymbol{n}_j.$$

Therefore, if $i \neq j$, all the random variables in $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$ are independent. Hence we could get that

$$\mathbb{E}\,[NU_2^\top U_2N^\top]_{ij} = 0.$$

If $i = j$, we assume that $\boldsymbol{z}_i = U_2\boldsymbol{n}_i = (z_{1i}, z_{2i}, \ldots, z_{mi})^\top$, and thus

$$\mathbb{E}\,[NU_2^\top U_2N^\top]_{ii} = \mathbb{E}\,\sum_{k=1}^n z_{ki}^2$$

where $z_{ki} \sim \mathcal{N}(0, \sum_{l=1}^n U_{2lk}^2)$. Therefore,

$$\mathbb{E}\,\sum_{k=1}^n z_{ki}^2 = \sum_{k=1}^n \mathbb{E}\,z_{ki}^2 = \sum_{k=1}^n\sum_{l=1}^n U_{2lk}^2 = \|U_2\|_F^2.$$

Hence,

$$\mathbb{E}\,[NU_2^\top U_2N^\top] = \|U_2\|_F^2\mathbf{E}_1$$

Finally, we substitute the expectation into (20) to obtain

$$\boldsymbol{tr}(U_1\mathbb{E}\,[NU_2^\top U_2N^\top]U_1^\top) = \|U_1\|_F^2\|U_2\|_F^2. \quad (21)$$

The proof completes. □

By Thm. 4, we could formulate the optimization problem as follows:

$$\begin{aligned}&\min_{U_1,U_2}\ \mathbb{E}\|Y-\hat{Y}\|_F^2\\ \Leftrightarrow\ &\min_{U_1,U_2}\ \mathbb{E}\|W_1U_1NU_2^\top W_2^\top\|_F^2\\ \Leftrightarrow\ &\min_{U_1,U_2}\ \|W_1U_1\|_F^2\|W_2U_2\|_F^2\\ \Leftrightarrow\ &\min_{U_1,U_2}\ \|W_1W_{U_1}S_{U_1}\|_F^2\|W_2W_{U_2}S_{U_2}\|_F^2\end{aligned} \quad (22)$$

where $S_{U_1} = diag(\sigma_{11}, ..., \sigma_{1m})$ and $S_{U_2} = diag(\sigma_{21}, ..., \sigma_{2n})$. If we let $P_{1i} = \sum_{j=1}^{m'}(W_1W_{U_1})_{ji}^2$ and $P_{2i} = \sum_{j=1}^{n'}(W_2W_{U_2})_{ji}^2$, we can write our objective as

$$\min_{U_1,U_2}\ \left(\sum_{i=1}^m P_{1i}\sigma_{1i}^2\right)\left(\sum_{i=1}^n P_{2i}\sigma_{2i}^2\right). \quad (23)$$

We assume P_{1i} and P_{2i} are constant. Together with the differential privacy constraint, we have a geometric programming problem:

$$\begin{aligned} &\min_{\sigma_{1i},\sigma_{2i}} \left(\sum_{i=1}^{m} P_{1i}\sigma_{1i}^2\right)\left(\sum_{i=1}^{n} P_{2i}\sigma_{2i}^2\right), \\ &\text{s.t.} \quad \left(\sum_{i=1}^{m} \frac{1}{\sigma_{1i}^2}\right)\left(\sum_{i=1}^{n} \frac{1}{\sigma_{2i}^2}\right) \le B, \end{aligned} \tag{24}$$

where $B = \frac{\left(-\beta+\sqrt{\beta^2+8\alpha\epsilon}\right)^2}{4\alpha^2}$. The we convert the problem into a convex one by letting $e^{x_{ij}} = \sigma_{ij}^2$:

$$\begin{aligned} &\underset{x}{\text{minimize}} \log(g(x)), \\ &s.t. \quad \log(g_1(x)) \le \log(B), \end{aligned}$$

where

$$\begin{aligned} g(x) &= \left(\sum_{i=1}^{m} P_{1i}e^{x_{1i}}\right)\left(\sum_{i=1}^{n} P_{2i}e^{x_{1i}}\right), \\ g_1(x) &= \left(\sum_{i=1}^{m} e^{x_{1i}}\right)\left(\sum_{i=1}^{n} e^{x_{2i}}\right). \end{aligned}$$

KKT conditions [35] can be applied and we obtain the optimal solution:

$$\sigma_{1k}^2\sigma_{2l}^2 = \frac{\left(\sum_{i=1}^{m}\sqrt{P_{1i}}\right)\left(\sum_{j=1}^{n}\sqrt{P_{2j}}\right)}{\sqrt{P_{1k}P_{2l}}B}, \quad \forall k \in [m], l \in [n]. \tag{25}$$

We consider all $\sigma_{1k}^2\sigma_{2l}^2$ satisfying the above equation are optimal solutions to the problem. Given the optimal solutions, we can calculate the minimum value of the error as:

$$\text{Error}_{\text{MGM}}(Y,\epsilon,\delta) = \frac{\left(\sum_{i=1}^{m}\sqrt{P_{1i}}\right)^2\left(\sum_{i=1}^{n}\sqrt{P_{2i}}\right)^2}{B}, \tag{26}$$

and the optimization scheme is summarized in Alg. 1 for MGM.

Algorithm 1 Generating Optimized Matrix Noise

Input: (a) privacy parameters ϵ, δ, (b) l_2 sensitivity $s_2(f)$, (c) the utility subspace $W_1 \in \mathbb{R}^{m'\times m}$, $W_2 \in \mathbb{R}^{n'\times n}$ (d) the row-wise noise directions $W_{U_1} \in \mathbb{R}^{m\times m}$ and the the column-wise noise directions $W_{U_2} \in \mathbb{R}^{n\times n}$.
Output: $f(X) + \mathcal{Z}$

compute $\alpha,\ \beta$ as $\alpha = s_2^2(f)$, and $\beta = 2\zeta(\delta)s_2(f)$
compute $B = \frac{(-\beta+\sqrt{\beta^2+8\alpha\epsilon})^2}{4\alpha^2}$
for $i \in \{1, ..., m\}$ **do**
  for $j \in \{1, \ldots, n\}$ **do**
    $P_{1i} = \sum_{l=1}^{m'}(W_1W_{U_1})_{li}^2$, $P_{2j} = \sum_{k=1}^{n'}(W_2W_{U_2})_{ki}^2$
  end for
end for
$\sigma_{1k}^2\sigma_{2l}^2 = \frac{\left(\sum_{i=1}^{m}\sqrt{P_{1i}}\right)\left(\sum_{j=1}^{n}\sqrt{P_{2j}}\right)}{\sqrt{P_{1k}P_{2l}}B}, \quad \forall k \in [m], l \in [n].$
sampling N_{ij} from $\mathcal{N}(0,1)$ for all $1 \le i \le m, 1 \le j \le n$
compute $\mathcal{Z} = U_1NU_2$ with the value of $\sigma_{1k}^2\sigma_{2l}^2$.
return $f(X) + \mathcal{Z}$

7 Comparison with Other Mechanisms

For a better understanding of the position of this work in the current literature, we compare MGM with existing differential privacy mechanisms on matrix-valued data, mainly Matrix Mechanism [10] and Matrix Variate Gaussian [11].

7.1 Comparison with Matrix Mechanism

Matrix Mechanism (MM) has been adopted for answering linear queries with differentially-private vector data:

Definition 6 (Matrix mechanism [10])**.** *Given an $m \times n$ workload matrix W, a $p \times n$ strategy matrix A that supports W and a differentially private algorithm $\mathcal{K}(A, x)$ that answers A with a given database instance x. The matrix mechanism $\mathcal{M}_{\mathcal{K},A}$ outputs the following vector:*

$$\mathcal{M}_{\mathcal{K},A}(W, x) = WA^+\mathcal{K}(A, x),\ \mathcal{K}(A, x) = Ax + \|A\|\tilde{b}$$

where $\tilde{b} = (b_1, \ldots, b_n)$ is an i.i.d random vector that does not depend on W or x.

Note that W above is analogous to the linear utility subspace in MGM. The goal of MM is to minimize the following error defined on $A, A^+, \tilde{b}$:

$$\text{Error}_{\text{MM}} = \mathbb{E}\|A\|\|WA^+\tilde{b}\|_F^2 = \|A\|\|WA^+\|_F^2 Var(b_1).$$

Hence, MM seeks pseudoinverse matrix A^+ to minimize the above error. However, the optimization problem is a semidefinite program and has no analytic solution, which costs about $O(m^4(m + n)^4)$ to search the solution. The semidefinite programming procedure largely constrains the scalability of the mechanism. It would be extremely complicated to solve A^+ in a very high-dimensional scenario. In comparison, the error of MGM is defined by Eq. (16) and the error minimization problem can be transformed into a convex one with a closed-form solution, which has a lower time complexity of $O(mn^2)$.

7.2 Comparison with Matrix Variate Gaussian

Matrix Variate Gaussian (MVG) mechanism guarantees (ϵ, δ)-differential privacy for matrix-valued queries through the matrix variate Gaussian distribution:

$$\mathcal{MVG}_{m,n}(M, \Sigma, \Psi) = \mathcal{MN}_{m,n}(X|M, \Sigma, \Psi),$$

where $\Sigma \in \mathbb{R}^{m\times m}$ is the row-wise covariance and $\Psi \in \mathbb{R}^{n\times n}$ is the column-wise covariance. Similar to MGM, MVG is also an additive noise scheme:

Definition 7 (MVG [11])**.** *Given a matrix-valued query function $f(X) \in \mathbb{R}^{m\times n}$, and a matrix-valued random variable $\mathcal{Z} \sim \mathcal{MVG}_{m,n}(\mathbf{0}, \Sigma, \Psi)$, the MVG mechanism is defined as*

$$\mathcal{MVG}_{m,n}(f(X)) = f(X) + \mathcal{Z}.$$

The differential privacy guarantee is imposed by the contraints on Σ and Ψ:

Theorem 5 (MVG [11])**.** *Let*

$$\begin{aligned} \sigma(\Sigma^{-1}) &= [\sigma_1(\Sigma^{-1}), ..., \sigma_m(\Sigma^{-1})]^T, \\ \sigma(\Psi^{-1}) &= [\sigma_1(\Psi^{-1}), ..., \sigma_n(\Psi^{-1})]^T, \end{aligned}$$

be the vectors of the non-increasingly ordered singular value of Σ^{-1} and Ψ^{-1} respectively. The MVG mechanism guarantees (ϵ, δ)-differential privacy if Σ and Ψ satisfy the following condition:

$$\|\sigma(\Sigma^{-1})\|_2\|\sigma(\Psi^{-1})\|_2 \le \frac{(-\beta_0 + \sqrt{\beta_0^2 + 8\alpha_0\epsilon})^2}{4\alpha_0^2}, \tag{27}$$

where $\alpha_0 = [H_r + H_{r,1/2}]\gamma^2 + 2H_r\gamma s_2(f)$, $\beta_0 = 2(mn)^{1/4}H_r\zeta(\delta)s_2(f)$, $\gamma = \sup_X \|f(X)\|_F$, $r = \min\{m, n\}$ *and* H_r *is generalized harmonic numbers of order* r.

For fair comparison with MVG, we list the conclusion of MGM in the below. By Thm. 1, noise $\mathcal{Z}$ needs to satisfy

$$\|\sigma(\Sigma^{-1})\|_2\|\sigma(\Psi^{-1})\|_2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4\alpha^2\sqrt{mn}} \quad (28)$$

to guarantee (ϵ, δ)-differential privacy. In the equation, $\alpha = s_2^2(f)$, and $\beta = 2\zeta(\delta)s_2(f)$. Compared to α_0 in Eq. (27), α is reduced by $\frac{1}{s_2^2(f)}[(H_r+H_{r,1/2})\gamma^2+2H_r\gamma s_2(f)]$. And β is reduced by $(mn)^{\frac{1}{4}}H_r$ comparing with β_0 in Eq. (27). Overall, the right-hand side of inequality (28) is about H_r^2 times larger than that of inequality (27). A larger right-hand side value indicates a smaller amount of noise minimally required to ensure differential privacy, and thus better utility. Such utility improvement is mainly because we use Lemma 1 rather than the harmonic numbers in the proof. More importantly, our UDN and IDN schemes further improve the utility of MGM with directional noise.

8 Evaluations

To show the wide application range of our proposed mechanisms, we run a series of experiments in different settings, including a variety of datasets, models, as well as federated learning frameworks. Experimental results are compared against a number of existing mechanisms to show the superiority of MGM.

TABLE 2
Setup for Type I Experiments

Dataset	MNIST	CIFAR-10	IMDB	Adult
Model	LeNet	ResNet-18	LSTM	MLP
Training	55,000	50,000	25,000	32,561
Testing	5,000	10,000	25,000	16,281
Clip Value	0.01	0.05	0.3	0.015
Batch size	64	16	256	64
Rounds	25	50	10	10
Local epochs	3	3	2	2
Gradient Shape	120 ×400	4,608×512	20,002 ×128	105 × 12

8.1 Setup

Baselines and metrics. We compare MGM with other differential privacy mechanisms dealing with high-dimensional data. The baselines include: i.i.d. Gaussian mechanism, Matrix Variate Gaussian (MVG) [11], Matrix Mechanism (MM) [10]. For MGM, we implement UDN and IDN as specialized instances fitting to different application scenarios. As the Gaussian mechanism is conventionally designed for scalar-valued queries, we simply view the matrix as a collection of its elements and add i.i.d. noise to each of the elements. MM is only applicable to small-scale datasets due to its high complexity. For most of the experiments, we use testing accuracy as the utility metric.

Datasets and tasks. We select 7 typical learning tasks from multiple areas where the data are likely to be sensitive. For computer vision, we have two image classification tasks, respectively on MNIST [36] and CIFAR-10 [37]. For data mining, we run classification tasks respectively on Adult [38], Purchase[1], Texas hospital stays[2] and Locations[3]. Adult is a small-scale dataset on which one predicts whether the income exceeds a threshold. Due to its huge computional cost, MM can only be applied in the data mining datasets. For text mining, we choose a binary classification task on IMDB [39] dataset.

Federated learning. According to the distribution of data among all parties in the feature and sample space, federated learning is divided into horizontal and vertical federated learning.

In horizontal FL, the training data gets partitioned horizontally among parties, *i.e.*, data matrices or tables are partitioned by rows and data from different rows share the same attributes. In the initial round of training, participants pull a randomly initialized model from the centralized server, and train the model on their respective local datasets. In each following round, each participant uploads locally computed gradients to the server for aggregating their updates. The central server maintains a global model, and uses the averaged gradients to update the global model. As a final step, the server sends the updated model to each participant for the next round of local training. Since the gradients reveal private information of each participant, it is critical to preserve privacy on the uploaded gradients.

Different from horizontal FL, vertical FL aligns samples rather than attributes. Each participant has its own local neural network which may be different depending on the features they process. Different parties jointly build the interactive layer which puts the intermediate features of all participants together as one output. The interactive layer is owned by the central server, who also builds the top neural network model and feeds the top model with the output of the interactive layer. In the forward propagation, each participant feeds its input to the respective local model to produce interactive-layer features. The features are fed to the top model to calculate the loss. The backward loop propagates the error from the output layer to the interactive layer and then to each local model. The local model gets updated accordingly. As the intermediate-layer features leak private information of the participant, privacy-preserving mechanisms need to be applied to the features.

Privacy-preserving federated learning. As aforementioned, the privacy-preserving objectives differ for horizontal and vertical FL. In the horizontal FL, we implement differential privacy on the gradients in each round, since they are uploaded to a potentially malicious centralized server. In the vertical FL, we consider differentially-private features would leak private information on training datasets. In each training iteration, participants release differentially-private features which would be further trained on the top model. We state the implementation detail by types as follows.

8.2 Implementation Details

Composition and Sampling. We introduce the composition and sampling theorems used in the implementation. Since

1. Purchase dataset is based on Kaggle's "acquire valued shoppers" challenge dataset. https://kaggle.com/c/acquire-valued-shoppers-challenge/data
2. https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm
3. https://sites.google.com/site/yangdingqi/home/foursquare-dataset

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3093316, IEEE Transactions on Mobile Computing

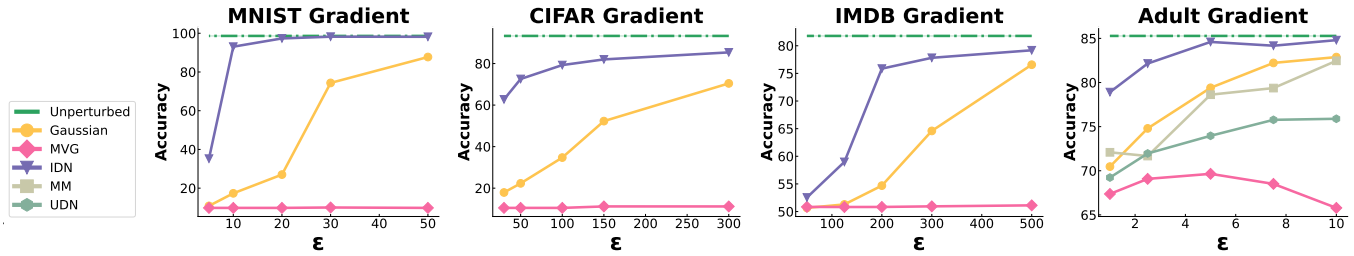

Fig. 2. Differentially-private horizontal federated learning over a variety of datasets. IDN has the best performance overall. Gaussian performs better than MVG. MM has a similar performance with Gaussian on Adult, which is better than UDN.

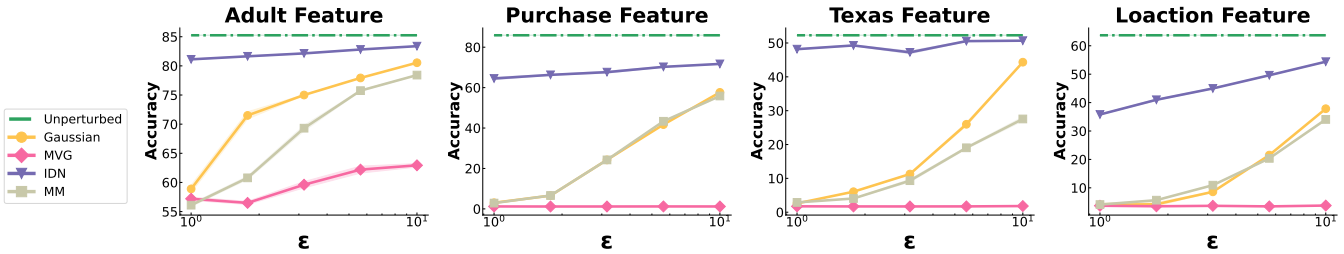

Fig. 3. Differentially-private vertical federated learning on a variety of datasets. IDN performs best overall with an increasing advantage over others as ϵ is smaller. IDN's performance is followed by that of the Gaussian mechanism and MM. The performance of Gaussian and MM is close to each other. MVG has the worst performance of all.

TABLE 3
Setup for Type II Experiments

Dataset	Adult	Purchase	Location	Texas
Model	MLP	FCN	NN	NN
Training Data	32,561	10,000	1,600	10,000
Testing Data	16,281	97,324	3410	57,330
Attributes	14	600	446	6170
Feature Shape	2×10	100×64	100×128	30×64

Gaussian mechanism $\mathcal{M}$ is applied to each element of X, and each mechanism satisfies (ϵ, δ)-differential privacy, the composition result satisfies $(mn\epsilon, mn\delta)$-differential privacy. Therefore, the ϵ we present in the experiment result is the ϵ for Gaussian mechanism on each element. For MVG, MGM, and MM in Type I experiment, we adopt the composition theorem (Theorem 3.4 from [40]) to ensure the overall (ϵ, δ)-differential privacy.

Also, we adopt the 'privacy amplification via sampling' from [41]:

Theorem 6 (lemma 2 in [41]). *Let $\mathcal{A}$ be an ϵ^*-differentially private algorithm. Construct an algorithm $\mathcal{B}$ that on input a database $D = (d_1, \ldots, d_n)$, constructs a new database D_s whose i-th entry is d_i with probability $f(\epsilon, \epsilon^*) = (\exp(\epsilon) - 1)/(\exp(\epsilon^*) + \exp(\epsilon) - \exp(\epsilon - \epsilon^*) - 1)$, and $\perp$ otherwise, and then runs $\mathcal{A}$ on D_s. Then, $\mathcal{B}$ is ϵ-differentially private.*

For example, during the training process, we take a random sample from the training set with sampling probability q. Then we have $f(\epsilon, \epsilon^*) = q$, and ϵ^* can be calculated as the private budget of the mechanism after sampling.

Type I: Private SGD in horizontal FL. We deploy experiments on datasets including MNIST, CIFAR-10, IMDB and Adult across 5 FL participants. For horizontal FL, we divide the dataset into five independent subsets. The five participants train respectively over a subset, and updates the gradients with the global model. Here we set the privacy parameter δ to 10^{-5} (according to $\delta = o(1/n)$ where n is the sample numbers per participant) and choice of ϵ is presented in Fig. 2. Each participant applies differential privacy on the gradients trained on the local dataset, of which the procedures are as follows:

1) Download the model parameters from the server.

2) Local epochs: compute gradients on the local dataset to update the local model, and then repeats.

3) Clip the accumulated gradients by its l_∞ norm.

4) **Apply perturbation** to the clipped accumulated gradients and upload them to the server.

5) Server averages the gradients for all the clients, and updates the corresponding model parameters with the perturbed gradients and go back to 1).

In 2), after training on the local dataset for a number of epochs, each participant uploads the accumulated gradients to the server who performs aggregation and publishes the new model to each participant.

Query function. Note that the query function here is a sum function on gradients:

$$f(X) = \sum_i g_i(X), \tag{29}$$

where $g_i(X)$ is the gradient computed on the i-th local epoch. The shape of $f(X)$ is the same as the model. For neighboring datasets $\{X, X'\}$, the l_2-sensitivity is

$$s_2(f) = \sup_{X, X'} \|f(X) - f(X')\|_F = 2C\sqrt{mn}, \tag{30}$$

where C is the clip value in Table 2, and (m, n) is the shape of the gradient.

The specific training hyperparameters can be found in Table 2 where the gradient shape reports only the largest shape of accumulated gradients we perturb. Step 4) is where we implement different privacy mechanisms. We apply l_∞ clipping to the gradients in order to satisfy the condition of IDN (Thm. 3). And we only implement UND on Adult due to the theoretical error of UND is relatively large for

TABLE 4
Comparison of Theoretical Errors

Method	$\mathbb{E}\ \|\mathcal{Z}\|^2$			
	Gaussian	MVG	UDN	IDN
Type I	$mn\sigma^2 \approx O(m^3n^3)$	$m^2n/B_{MVG} \approx O(m^4n^3\ln^2(m+1))$	$m^2n/B_{UDN} \approx O(m^4n^3)$	$m^2n/B_{IDN} \approx O(m^3n^3)$
Type II	$mn\sigma^2 \approx O(m^3n^3)$	$m^2n/B_{MVG} \approx O(m^4n^3\ln^2(m+1))$	$\frac{n(\sum_{i=1}^m\sqrt{P_i})^2}{B_{UDN}} \approx O(m^4n^3)$	$\frac{n(\sum_{i=1}^m\sqrt{P_i})^2}{B_{IDN}} \approx O(m^3n^3)$

the rest datasets. The directional matrix W_{U_1} of IDN and MVG could be any orthogonal matrix, and we choose the identity matrix. In MVG and MM, we set W (or W_1, W_2) to identity matrix since the utility subspace is unknown. In MVG, we implement the *binary precision allocation strategy* [11] to decide the importance of different directions and SVD to calculate the directional matrix.

Type II: Private training features in vertical FL. Four datasets are adopted: Adult, Purchase, Location, and Texas hospital stays, for vertical FL over 2 participants. We divide each dataset into two halves by data attributes. For example, we distribute the first 7 out of 14 attributes of Adult to a participant, and the rest to the other. Each participant trains its local model over its subset, and uploads noisy features to the top model. We set the privacy parameter ϵ to the range of $[1, 10]$ and δ to 10^{-5}. Privacy amplification scheme is also adopted as in Type I experiments. To meet the condition of IDN, we modify each model by replacing its activation function with $\tanh(\cdot)$ to normalize the released intermediate-layer feature to the range of $(-1, 1)$. Detailed configurations are given in Table 3.

Query function. We set the query function as the identity query $f(X) = X \in \mathbb{R}^{m\times n}$. X is the training features in the experiment. For neighboring datasets $\{X, X'\}$, the l_2-sensitivity is the feature size multiplied by the feature range. Here our range is set to $(-1, 1)$, and thus the l_2-sensitivity is

$$s_2(f) = \sup_{X,X'} \|X - X'\|_F = 2\sqrt{mn} \tag{31}$$

For IDN, MVG and MM, we set the utility subspace W (or W_1, W_2) to $\mathbf{E}$ as in Type I. For the i.i.d. Gaussian and MM, l_2-sensitivity is computed for each element and for IDN and MVG, l_2-sensitivity is computed on the feature matrix. The rest settings are the same as that of Type I.

8.3 Experimental Results

Before delving into the experimental results, we first present theoretical analysis of the expected error of each differential privacy mechanism under the same privacy condition. The theoretical results are presented in Table 4. In the table, m, n are respectively the row and column number. The directional noise is applied row-wise in UDN and IDN. For Gaussian mechanism, σ is the standard deviation of the i.i.d Gaussian distribution in the scalar-valued Gaussian mechanism [13], *i.e.*, $\sigma \geq c\Delta_2(f)/\epsilon$ and $c^2 > 2\ln(1.25/\delta)$.

For ease of understanding the theoretical results, we present the approximation of the expected magnitude of noise in each case besides the exact values. As we found, IDN and Gaussian have the same order of expected error. However, the results are given without considering directional noise. If the optimized scheme is applied, IDN is supposed to incur less error. MM is missing from Table. 4 since the error is data-dependent or not deterministic.

TABLE 5
Running time for different mechanisms.

Method	Running time(s)
Gaussian	0.00164
MVG	0.51146
MM	0.82323
UDN	0.00248
IDN	0.00224

In the following, we will present experimental results under a variety of privacy settings and see how much they agree with the theoretical error.

Type I Results. Fig. 2 reports accuracies over different datasets. 'Unperturbed' represents the case with no privacy guarantee. As we can tell, for all cases, accuracies steadily improve as ϵ increases, which agrees with the privacy-utility tradeoff. In general, the accuracy performance follows IDN $>$ Gaussian $>$ MVG in all the cases. In the experiment on Adult, the performance of MM is quite close to Gaussian, which is even better than UDN. The result of Type I presents that, most mechanisms have a similar performance with a large ϵ value. We consider it mainly due to a lack of optimization space with less amount of noise (larger ϵ).

The results of Type I are in accords with the theoretical results in the Table 4. Here, we can see that ϵ is larger on CIFAR-10 and IMDB than MNIST and Adult, to be able to yield a valid model. This may be that the shape of the gradients in the former two datasets are very large, where even a moderate ϵ value would incur an overwhelming amount of noise. Moreover, the CIFAR and IMDB tasks are more complicated than the other two, and hence are sensitive to noise.

Type II Results. The performance of IDN is also the best among all the mechanisms. The i.i.d. Gaussian has similar accuracy performance with MM, since in MM, the noise is also sampled from the Gaussian distribution. Actually, the performance of MM largely depends on whether the query matrix W is properly chosen. As we analyze above, Gaussian and IDN have similar performance over Adult dataset, mainly due to a smaller feature size of Adult dataset, and much smaller noise is inserted thereafter. The accuracy curve of MVG almost flattens over different ϵs, totally destroying model utility. Overall, the performance of all mechanisms is in accord with the theoretical results in Table 4.

To have an idea of how efficient our algorithms are, we examine the wall-clock running time for each mechanism, of which the result is shown in Table 5. The implementation is done in `Python 3.8.8` and all results are measured on a server with Intel Xeon Silver 4116 @ 2.10GHz. We observe that the running time for MVG and MM is much larger than other mechanisms. For MVG, the most time-consuming part is running SVD. And for MM, the most time-consuming part is to solve the optimization by `CVXPY`. In contrast, Gaussian mechanism takes the least time, as it does not need to

do matrix multiplications as performed in IDN and UDN. Still, our mechanisms yield reasonable running time among matrix-valued DP schemes.

9 Conclusion

In this paper, we propose a differential privacy mechanism for matrix-valued queries called MGM. We show MGM enjoys a tighter noise bound than previous works, and thus has better utility. Two special forms of MGM are discussed, respectively unimodal directional noise (UDN) and independent directional noise (IDN), both of which progressively achieve better utility under the same differential privacy guarantee. We implement MGM in the federated learning setting and the experimental results have shown the superiority of MGM in terms of data utility.

References

[1] A. Ghosh, J. Ding, R. Sarkar, and J. Gao, "Differentially private range counting in planar graphs for spatial sensing," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2233–2242.

[2] H. Asif, P. A. Papakonstantinou, and J. Vaidya, "How to accurately and privately identify anomalies," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 719–736.

[3] H. Sun, X. Xiao, I. Khalil, Y. Yang, Z. Qin, H. Wang, and T. Yu, "Analyzing subgraph statistics from extended local views with decentralized differential privacy," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 703–717.

[4] A. Xiong, T. Wang, N. Li, and S. Jha, "Towards effective differential privacy communication for users' data sharing decision and comprehension," *arXiv preprint arXiv:2003.13922*, 2020.

[5] C. Dwork, "A Firm Foundation for Private Data Analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[6] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 1310–1321.

[7] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2016, pp. 308–318.

[8] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Scalable Private Learning with PATE," in *Proc. of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[9] Q. Geng and P. Viswanath, "Optimal noise adding mechanisms for approximate differential privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 952–969, 2015.

[10] C. Li, G. Miklau, M. Hay, A. McGregor, and V. Rastogi, "The matrix mechanism: optimizing linear counting queries under differential privacy," *The VLDB journal*, vol. 24, no. 6, pp. 757–781, 2015.

[11] T. Chanyaswad, A. Dytso, H. V. Poor, and P. Mittal, "Mvg mechanism: Differential privacy under matrix-valued query," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 230–246.

[12] L. Xiang, J. Yang, and B. Li, "Differentially-private deep learning from an optimization perspective," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 559–567.

[13] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[15] B. Balle and Y.-X. Wang, "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[16] J. Champion, A. Shelat, and J. Ullman, "Securely sampling biased coins with applications to differential privacy," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 603–614.

[17] E. Lobo-Vesga, A. Russo, and M. Gaboardi, "A programming framework for differential privacy with accuracy concentration bounds," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 411–428.

[18] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not just privacy: Improving performance of private deep learning in mobile cloud," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2407–2416.

[19] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 304–317.

[20] X. Pan, M. Zhang, S. Ji, and M. Yang, "Privacy risks of general-purpose language models," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1314–1331.

[21] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially Private Empirical Risk Minimization," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.

[22] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 245–248.

[23] R. Bassily, A. Smith, and A. Thakurta, "Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds," in *Proc. of the IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2014, pp. 464–473.

[24] D. Wang, M. Ye, and J. Xu, "Differentially Private Empirical Risk Minimization Revisited: Faster and More General," in *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2719–2728.

[25] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[26] M. Lécuyer, R. Spahn, K. Vodrahalli, R. Geambasu, and D. Hsu, "Privacy accounting and quality control in the sage differentially private ml platform," in *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, 2019, pp. 181–195.

[27] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. of the 2019 IEEE Symposium on Security and Privacy (S&P),*, 2019.

[28] Z. Xu, S. Shi, A. X. Liu, J. Zhao, and L. Chen, "An adaptive and fast convergent approach to differentially private deep learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1867–1876.

[29] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional Mechanism: Regression Analysis under Differential Privacy," *Proc. of the VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012.

[30] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction," in *Proc. of the 30th AAAI Conference on Artificial Intelligence*, vol. 16, 2016, pp. 1309–1316.

[31] J. Zhang, K. Zheng, W. Mou, and L. Wang, "Efficient Private ERM for Smooth Objectives," in *Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 2017, pp. 3922–3928.

[32] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2587–2596.

[33] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: federated learning with local differential privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 61–66.

[34] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, 2020.

[35] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[36] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[37] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[38] D. Dua and C. Graff, "Uci machine learning repository," 2017.

[39] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings*

of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[40] P. Kairouz, S. Oh, and P. Viswanath, "The Composition Theorem for Differential Privacy," *IEEE Transactions on Information Theory (TIT)*, vol. 63, no. 6, pp. 4037–4049, 2017.

[41] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *Theory of Cryptography Conference*. Springer, 2010, pp. 437–454.

Jungang Yang received the Bachelor's degree in Information and Computing Science from Nanjing University, Nanjing, China, in 2019. He is currently pursuing the Ph.D. degree in computer science, Shanghai Jiao Tong University. His reasearch interests include privacy analysis and adversarial learning.

Liyao Xiang received the B.Eng. degree in Electrical and Computer Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the Ph.D. degree in Computer Engineering from the University of Toronto, Toronto, ON, Canada, in 2018. She is currently an assistant professor at Shanghai Jiao Tong University. Her research interests include security and privacy, privacy analysis in data mining, and mobile computing.

Jiahao Yu is pursing his Bachelor degree at the School of Electronic Information and Electrical Engineering of Shanghai Jiao Tong University. y. He focuses primarily on the areas of machine learning and security, specifically exploring the robustness of machine learning models against various adversarial attacks.

Xinbing Wang received the B.S. degree (with hons.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received the Ph.D. degree, major in the Department of electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor in the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. Dr. Wang has been an associate editor for IEEE/ACM Transactions on Networking and IEEE Transactions on Mobile Computing, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, ACM MobiHoc 2012-2014, IEEE INFOCOM 2009-2017.

Bin Guo received the Ph.D. degree in computer science from Keio University, Japan, in 2009, and then was a postdoc researcher with Institut Télécom SudParis, France. He is a professor with Northwestern Polytechnical University, China. His research interests include ubiquitous computing, mobile crowd sensing, and HCI. He has served as an associate editor of the IEEE Communications Magazine and the IEEE Transactions on Human-Machine-Systems, the guest editor of the ACM Transactions on Intelligent Systems and Technology and the IEEE Internet of Things, the general cochair of the 12th IEEE International Conference on Ubiquitous Intelligence and Computing (IEEE UIC'15), and the program chair of IEEE CPSCom'16, ANT'14, and UIC'13. He has published more than 90 papers in refereed journals, conference proceedings, and book chapters. He is a senior member of the IEEE

Zhetao Li received the B.Eng. degree in Electrical Information Engineering from Xiangtan University in 2002, the M.Eng. degree in Pattern Recognition and Intelligent System from Beihang University in 2005, and the Ph.D. degree in Computer Application Technology from Hunan University in 2010. He earned the certificate of Predeparture English Language Training from Beijing Language and Culture University in 2011. Dr. Li is currently an associate professor of College of Information Engineering, Xiangtan University. He was a visiting researcher at Ajou University from May to Aug 2012. From Feb 2013 to Dec 2013, he was a clerk fellow at the Department of Science & Technology, Ministry of Education of the People's Republic of China.

Baochun Li received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 1995, and the M.S. and Ph.D. degrees from the Department of Computer Science, University of Illinois at Urbana– Champaign, Champaign, in 1997 and 2000, respectively. He held the Nortel Networks Junior Chair with Network Architecture and Services from 2003 to 2005. He has been the Bell Canada Endowed Chair in computer engineering since 2005. Since 2000, he has been with the Department of Electrical and Computer Engineering, University of Toronto, where he is currently a Professor. His research interests include cloud computing, largescale data processing, computer networking, and distributed systems. He is a member of ACM. He was a recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems in 2000. He was a recipient of the Multimedia Communications Best Paper Award from the IEEE Communications Society in 2009, and a recipient of the University of Toronto McLean Award.