Differential Privacy for Tensor-Valued Queries

Jungang Yang, Liyao Xiang, *Member, IEEE*, Ruidong Chen, Weiting Li, and Baochun Li, *Fellow, IEEE*

***Abstract*— Private individual information are increasingly exposed through high-dimensional and high-order data, with the wide deployment of learning techniques. These data are typically expressed in form of tensors, but there is no principled way to guarantee privacy for tensor-valued queries. Conventional differential privacy is typically applied to scalar values without a precise definition on the shape of the queried data. Realizing that the conventional mechanisms do not take the data structural information into account, we propose *Tensor Variate Gaussian* (TVG), a new (ϵ, δ)-differential privacy mechanism for tensor-valued queries. We further introduce two mechanisms based on TVG with an improved utility by imposing the unimodal differentially-private noise. With the utility space available, the proposed mechanisms can be instantiated with an optimized utility, and the optimization problem has a closed-form solution scalable to large-scale problems. Finally, we experimentally test our mechanisms on a variety of datasets and models, demonstrating that TVG is superior than other state-of-the-art mechanisms on tensor-valued queries.**

***Index Terms*— Differential privacy, deep learning, stochastic gradient descent.**

I. Introduction

STORED and processed in the form of tensors, high-dimensional and high-order data are growingly demanding in a wide range of scenarios such as spatio-temporal user behavior modeling, social network analysis, and particularly big data applications. Tensors often preserve natural representation of data such as multimedia data, graph data, data cube, etc. Videos consist of correlated images over time which is counter-intuitive to transform into matrices. Graph data with multimodal features often contain multiple dimensions. Data cube would lose too much information if flattened into plain vectors. Obviously, different shapes of the tensors would yield different interpretations on the data, and thus demands special attention in handling its privacy condition.

Theoretical and software tools are rapidly emerging to identify or solve problems expressed in tensor-forms. However, these promising applications pose great threats to individual privacy, as the query results may single an individual out from a population of data records. For example, through tensor-valued features, an attacker is able to infer if a participant is in the training set. Typically, differential privacy constrains an adversary's capability to deduce anything about an individual with the released public information, but there is no principled way to guarantee differential privacy for tensors up to now.

The conventional way of handling tensor-valued data is to treat the tensor as a collection of its elements, or as a collection of vectors to guarantee differential privacy. Hence the tensor's potential structural information and relation among elements may be lost. For example, a symmetric tensor, of which elements remain constant under any permutation of the indices, would have different properties from others. In this work, we specify the definition of differential privacy on tensors, and design mechanisms to meet the definition.

Designing tensor-valued differential privacy mechanisms is particularly challenging, not only because the high-dimensional/order distribution is extremely complicated, but also due to a lack of general privacy mechanisms applicable to all forms of data. There are differential privacy mechanisms for scalars [1], [2], vectors [3], or matrices [4], [5], but there is no unified mechanism for different forms of data. However, tensor forms are all-inclusive: a first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are high-order tensors. We try to address this problem by designing a unified differential privacy mechanism despite the specific shape of tensors.

Preserving utility and privacy at the same time for high-dimensional/order data is also difficult. This originates from the fundamental trade-off between data privacy and utility. For tensor-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 [3], [5]–[7]. As most of the solutions are heuristic, there are no utility guaranteed, nor scalable approaches to large-scale applications. We show our mechanism has a natural form that is easy to optimize in terms of utility, and readily to deploy in large scale.

In this work, we formalize the study of tensor-valued differential privacy and innovatively propose a mechanism called *Tensor Variate Gaussian* (TVG). Preserving data's original structure, TVG adds differentially-private, tensor-valued noise to the data. The idea is to utilize the tensor variate Gaussian distribution to guarantee (ϵ, δ)-differential privacy, and the guarantee only depends on the covariance matrices of the noise. We rigorously prove that TVG 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.

Manuscript received December 7, 2020; revised April 24, 2021; accepted May 20, 2021. Date of publication June 16, 2021; date of current version December 29, 2021. This work was supported in part by the NSF, China, under Grant 61902245, Grant 62032020, Grant 61960206002, Grant 61822206, Grant 62020106005, and Grant 61829201; and in part by the Science and Technology Innovation Program of Shanghai under Grant 19YF1424500. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mohamed Ali Kaafar. *(Jungang Yang and Liyao Xiang are co-first authors.) (Corresponding author: Liyao Xiang.)*

Jungang Yang, Liyao Xiang, Ruidong Chen, and Weiting Li are with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiangliyao08@sjtu.edu.cn).

Baochun Li is with the University of Toronto, Toronto, ON M5S 3G4, Canada.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2021.3089884, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2021.3089884

We novelly found that TVG has many intriguing properties. Even if the tensor-valued data is reshaped, the same (ϵ, δ)-differential privacy may be guaranteed under TVG. That allows flexibility in applying differential privacy regardless of the specific tensor shape. We further discuss special cases of TVG where the tensor-valued noise is set to be unimodal. When the data and noise meet certain criteria, we apply unimodal-noise mechanisms that guarantee differential privacy at even tighter noise bounds and thus yielding higher data utility.

Taking into account the utility subspace of data in practical scenarios, we propose optimized differential privacy schemes based on TVG. Actually, TVG implies a set of mechanisms meeting the same differential privacy guarantee, which leaves much design space for us to manipulate for achieving better data utility. From the utility aspect, we observe different modes or dimensions of a tensor may have different impacts to the final results, depending on how the tensor-valued data is involved in the tasks. For example, in a facial image, the hair feature is more important than the background pattern feature to a face recognition task. 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 optimized schemes for TVG by imposing different weights on different parts of the tensor. Closed-form solutions for these optimization problems are derived, rendering TVG readily be deployed in large-scale scenarios.

We summarize highlights of our contributions as follows.

1) We propose a (ϵ, δ)-differential privacy mechanism TVG for tensor-valued queries. Regardless of the specific shape of the tensor, TVG and its two variants enjoy tighter noise bounds and higher utility than previous works.
2) We introduce utility optimization schemes based on TVG. Closed-form solutions and utility certificates are derived for the optimized noise distribution in TVG.
3) A series of experiments were conducted on different tensor-valued query functions, models and datasets. Results show that TVG is scalable, and achieves better utility than other mechanisms at the same privacy level.

II. Related Work

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

A. 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 the Gaussian mechanism [8], Laplace mechanism [2], Exponential mechanism [8], Johnson-Lindenstrauss (JL) Transform [4], Matrix Variate Gaussian (MVG) [5], and Matrix Mechanism (MM) [3].

Although none of the mechanisms are applicable to tensor-valued queries, our work is still 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. 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, where the vector data x is queried as Wx and W is a query matrix. It adds a vector-valued (Gaussian or Laplace) noise to the data and seeks an optimal transform to minimize the impact of the noise on the data. MVG is proposed for matrix-valued queries, and adds matrix-valued noise to guarantee (ϵ, δ)-differential privacy. It defines l_2-sensitivity on the Frobenius norm of the difference between two adjacent matrices. Our work is also an additive noise mechanism, but is about the tensor-valued queries. We prove that our mechanism achieves (ϵ, δ)-differential privacy based on the l_2-sensitivity of adjacent datasets.

Johnson-Lindenstrauss (JL) Transform is a multiplicative noise mechanism, most often used in the covariance query or covariance estimation. It generates a randomized noise matrix and multiplies the sensitive matrix-valued data by the generated noise matrix. The JL transform publishes a sanitized covariance matrix that preserves differential privacy w.r.t. bounded changes.

Apart from these basic mechanisms, we notice that a number of advanced schemes have also been proposed. Zero-concentrated differential privacy [9] imposes a bound on the moment generating function of the privacy loss, and enjoys a nice composition property than conventional differential privacy. Balle and Wang [1] improves the conventional Gaussian mechanism by directly using the Gaussian cumulative density function instead of a tail bound approximation.

Our work is also aligned with works addressing the utility of additive noise such as [5]–[7]. The optimal noise distribution is found by Geng and Viswanath [6] in terms of the magnitude of the noise, but has restriction on data dimensions. Similar to [7], 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 [5].

B. Sampling and Composition

There are mechanisms whose privacy guarantee is deducted from the primitive mechanisms. Examples include composition schemes [8], [10]–[14], privacy amplification by sampling [15], etc. Abadi *et al.* [10] introduce a new accounting method to compose the Gaussian mechanism, which reduces the total amount of additive noise with the same privacy guarantee, whereas Kairouz *et al.* put forward an optimal composition scheme for the general distribution of noise. Works of [12], [14] offer dynamic accounting methods based on the runtime convergence of the algorithm. And Lécuyer *et al.* [13] enforce a global differential privacy guarantee in a continuously growing data regime. Our work is orthogonal to these works but certainly can be applied together with these mechanisms.

C. Learning With Differential Privacy

There are a wide range of works applying differential privacy mechanisms to machine learning algorithms. Depending on different privacy-preserving goals, we have differentially-private inputs [5], [16], outputs [17], [18], gradients [12]–[14], [19]–[22], and objective functions [23]–[25], etc. We pay particular attention to the machine learning applications as they

are most likely to deal with data of high dimensions or orders. Our mechanism can be applied to the inputs, outputs and gradients wherever the objects in protection are of tensor form.

D. Tensor Decomposition With Differential Privacy

In big data analysis, tensor decomposition is also an important tool. In decomposing the tensor, it is necessary to use differential privacy for sensitive data protection, such as [26], [27]. These works mainly focus on preserving the sensitive eigenvectors and eigenvalues generated by tensor decomposition, as well as improving their effectiveness. And these works can apply one of the mechanisms in Sec. II-A to guarantee differential privacy. Therefore, tensor decomposition with DP is mainly an application of the DP mechanism, but rather a DP primitive specific to tensors. Our mechanism can also be applied to tensor decomposition as a primitive mechanism.

Overall, we consider our work proposes a primitive differential privacy mechanism, which is orthogonal to sampling and composition schemes but certainly can be combined with any of these. The major application of this work is where the tensor-valued data is sensitive and requires to be protected.

III. Preliminaries

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

A. Differential 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 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 $\mathcal{O}$,

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

In the special case of $\delta = 0$ we call $\mathbf{M}$ ϵ-differentially private.

B. Relevant Definitions and Lemmas

As we mainly focus on tensors, we clarify some of the tensor-related definitions and lemmas adopted in this paper.

Definition 2 (Orders, Fibers, and Slices): Following the convention of [28], we define:

- *The order of a tensor is the number of its dimensions, also known as ways or modes.*
- *Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one.*
- *Slices are two-dimensional sections of a tensor, defined by fixing all but two indices.*

We give an example for ease of understanding. The dataset of CIFAR-10 consisting of images of $32 \times 32 \times 3$, which are 3-order tensors. Assume that $\mathcal{X} \in \mathbb{R}^{32 \times 32 \times 3}$ contains elements x_{ijk}, $i \in [32]$, $j \in [32]$, $k \in [3]$. In particular, $(x_{111}, x_{112}, x_{113})$ is a fiber.

Definition 3 (n-Mode Matrix Product): The n-mode (matrix) product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n U$, where $n \in [N]$, and is of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. Elementwise, we have

$$(\mathcal{X} \times_n U)_{i_1 \ldots i_{n-1}\ j\ i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}. \tag{2}$$

Definition 4 (Tensor Inner Product): The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N} y_{i_1 i_2 \ldots i_N}. \tag{3}$$

It follows immediately that $\langle \mathcal{X}, \mathcal{X} \rangle = \|\mathcal{X}\|^2$.

Definition 5 (Matricization: transforming a tensor into a matrix:) Matricization is the process of reordering the elements of an N-way array into a matrix. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathcal{X}_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting matrix. Tensor element $(i_1, i_2, \ldots, i_N)$ maps to matrix element (i_n, j), where

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k \quad with \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m. \tag{4}$$

A more general treatment of matricization can be found in [29].

Definition 6 (Standard Normal Distribution (SND)): If a tensor-valued random variable $\mathcal{N} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ follows a standard normal distribution (SND), its probability density function is

$$\Pr(\mathcal{N}) = \frac{1}{(2\pi)^{\frac{I}{2}}} \exp\left\{ -\frac{1}{2} \langle \mathcal{N}, \mathcal{N} \rangle \right\}, \tag{5}$$

where $I = I_1 I_2 \cdots I_N$.

Note that if a tensor follows SND, each element of the tensor $\mathcal{N}_{i_1 \ldots i_N}$ follows the normal distribution $\mathcal{N}(0, 1)$. Besides the above definitions, we introduce some tensor-related lemmas concerning our work.

Lemma 1 (n-Mode Matrix Product to Kronecker Products [29]): Let $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be tensors and $U_n \in \mathbb{R}^{J \times I_n}$ for all $n \in \{1, \ldots, N\}$. Then for any $n \in [N]$, we have

$$\begin{aligned} &\mathcal{Y} = \mathcal{X} \times_1 U_1 \times_2 U_2 \ldots \times_N U_N \Leftrightarrow \\ &\mathcal{Y}_{(n)} = U_n \mathcal{X}_{(n)} \left(U_N \otimes \cdots \otimes U_{n+1} \otimes U_{n-1} \otimes \cdots \otimes U_1 \right)^{\top}, \end{aligned} \tag{6}$$

where $\otimes$ means the Kronecker product and the $\mathcal{Y}_{(n)}$ is the mode-n matricization of $\mathcal{Y}$, which is defined in Def. 5.

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

TABLE I
NOTATIONS

$\mathcal{X}$	a tensor
U	a matrix
$\mathcal{X} \times_n U$	n-Mode Matrix Product (Def. 3)
$\langle \mathcal{X}, \mathcal{Y} \rangle$	tensor inner product (Def. 4)
$\mathcal{X}_{(n)}$	matricization of tensor (Def. 5)
$\mathcal{N}$	a Standard Normal Distribution (SND) variable
$s_2(f)$	l_2-sensitivity of $f(\mathcal{X})$ (Def. 7)
$\mathcal{TVG}(\mu, \Sigma_1, \ldots, \Sigma_N)$	Tensor Variate Gaussian distribution (Def. 8)
$\mu,\ \Sigma_k$	the mean and the covariance matrix of $\mathcal{TVG}$
U_k	$U_k U_k^\top = \Sigma_k$
$\zeta(\delta)^2$	$-2\ln\delta + 2\sqrt{-I_1 I_2 \cdots I_N \ln\delta} + I_1 I_2 \cdots I_N$
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 $\Sigma_k = W_{U_k} S_k W_{U_k}^\top$
S_{U_k}	$S_{U_k} S_{U_k}^\top = S_k$
E_k	the identity matrix of dimension I_k
W_k	the utility subspace for mode k
P_{ki}	$\sum_{j=1}^{J_k} (W_k W_{U_k})_{ji}^2$

IV. TENSOR VARIATE GAUSSIAN MECHANISM

In this section, we first introduce the tensor-valued differential privacy mechanism called Tensor Variate Gaussian (TVG), and give the main theorem along with its sketch proof. The tensor-valued differential privacy mechanism is mostly different from the scalar-valued one in that the data or query are in tensor form, and we need to guarantee differential privacy on high-dimensional/order distributions regardless of the specific shape of the tensor.

We first define l_2-sensitivity on a pair of adjacent tensors as follows:

Definition 7 (l_2-sensitivity): The l_2-sensitivity of the query function $f(\mathcal{X}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as

$$s_2(f) = \sup_{d(\mathcal{X}, \mathcal{X}')=1} \|f(\mathcal{X}) - f(\mathcal{X}')\|, \tag{7}$$

where $\|\cdot\|$ is the tensor norm, $\mathcal{X}$ and $\mathcal{X}'$ are datasets expressed as tensors, whose fibers are data records. Then $d(\mathcal{X}, \mathcal{X}') = 1$ means that $\mathcal{X}$ and $\mathcal{X}'$ are neighboring datasets differing by only a single record.

Based on the standard normal distribution on tensors (Def. 6), we define tensor variate Gaussian distribution $\mathcal{TVG}$ and its variable $\mathcal{Z}$ as below:

Definition 8 (Tensor Variate Gaussian): A tensor variate Gaussian distribution $\mathcal{TVG}(\mu, \Sigma_1, \ldots, \Sigma_N)$ has mean $\mu \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and covariance matrices $\Sigma_k \in \mathbb{R}^{I_k \times I_k}$ for $k \in [N]$ and each Σ_k is a positive semidefinite matrix. If an N-order tensor-valued random variable $\mathcal{Z} \sim \mathcal{TVG}(\mu, \Sigma_1, \ldots, \Sigma_N)$, there exists a SND tensor $\mathcal{N} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ such that

$$\mathcal{Z} = \mu + \mathcal{N} \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_N U_N, \tag{8}$$

where $U_k \in \mathbb{R}^{I_k \times I_k}$ satisfies $U_k U_k^\top = \Sigma_k$ for each $k \in [N]$. And the probability density function for $\mathcal{Z}$ is

$$\Pr(\mathcal{Z}) = \frac{1}{(2\pi)^{\frac{I}{2}} |U_1|^{I/I_1} |U_2|^{I/I_2} \cdots |U_N|^{I/I_N}} \exp\left\{-\frac{1}{2}\langle \mathcal{N}, \mathcal{N} \rangle\right\}, \tag{9}$$

where $I = I_1 I_2 \cdots I_N$.

Note that $\mathcal{N} = (\mathcal{Z} - \mu) \times_1 U_1^{-1} \times_2 U_2^{-1} \times_3 \ldots \times_N U_N^{-1}$, and $\mathcal{N}$ is a special case of tensor variate Gaussian that $\mathcal{N} \sim \mathcal{TVG}(\mathbf{0}, E_1, \ldots, E_N)$, where $E_k \in \mathbb{R}^{I_k \times I_k},\ \forall k \in [N]$ is the identity matrix. By Eq. 8, it is clear that the tensor variate Gaussian is a linear transformation of the SND. With the above definition, we apply additive tensor-valued noise following $\mathcal{TVG}$ distribution in the TVG mechanism stated in the following.

Definition 9 (TVG Mechanism): For a given query function $f(\mathcal{X}) \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and a tensor variate Gaussian $\mathcal{Z} \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, \ldots, \Sigma_N)$, the TVG mechanism is defined as:

$$\mathbf{TVG}(f(\mathcal{X})) = f(\mathcal{X}) + \mathcal{Z}. \tag{10}$$

Similar to the Gaussian mechanism [8] and MVG [5], TVG adds zero-mean randomized noise to the query result. Note that $\Sigma_k \in \mathbb{R}^{I_k \times I_k}$ for $k \in [N]$ are the covariance matrices of different modes for $\mathcal{Z}$, 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. Before introducing our main theorem, we first present lemmas used in the proof of the theorem. Due to space constraint, we only illustrate a sketch proof of our theorem, and leave the complete proofs of the lemmas and theorem in supplemental materials.

Lemma 2 (Tensor norm inequality): Let $U_k \in \mathbb{R}^{I_k \times I_k}$ for $k \in [N]$, and $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be a pair of tensors which satisfy:

$$\mathcal{X} = \mathcal{Y} \times_1 U_1 \times_2 U_2 \times_3 \ldots \times_N U_N. \tag{11}$$

Then we have the tensor norm inequality that

$$\|\mathcal{X}\| \leq \|\mathcal{Y}\| \|U_1\|_F \|U_2\|_F \ldots \|U_N\|_F. \tag{12}$$

See Appendix A-A for the proof.

Lemma 3 (The bound of the SND tensor): For a tensor $\mathcal{N} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ following the standard normal distribution, $\delta \in (0, 1)$ and $\zeta(\delta)^2 = -2\ln\delta + 2\sqrt{-I_1\ I_2 \ldots I_N \ln\delta} + I_1\ I_2 \ldots I_N$, we have

$$\Pr[\|\mathcal{N}\|_F^2 \leq \zeta(\delta)^2] \geq 1 - \delta. \tag{13}$$

By Lemma 2 and 3, we can prove the main theorem on TVG mechanism defined in Def. 9:

Theorem 1 (Tensor Variate Gaussian): We have a query function $f(\mathcal{X}) \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, and a tensor variate Gaussian noise $\mathcal{Z} \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, \ldots, \Sigma_N) \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. $\Sigma_1, \ldots, \Sigma_N$ are the covariance matrices and $U_k \in \mathbb{R}^{I_k \times I_k}$ satisfies $U_k U_k^\top = \Sigma_k$ for each $k \in [N]$. The TVG mechanism guarantees (ϵ, δ)-differential privacy if $U_1, \ldots, U_N$ satisfy

$$\|U_1^{-1}\|_F^2 \ldots \|U_N^{-1}\|_F^2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4\alpha^2}. \tag{14}$$

$\alpha = s_2^2(f)$, and $\beta = 2\zeta(\delta) s_2(f)$, where $s_2(f)$ is the l_2-sensitivity of $f(\mathcal{X})$ and $\zeta(\delta)$ is defined in Lemma 3.

Note that the right side of Eq. (14) is a constant once the privacy parameters ϵ and δ are given. Hence the theorem shows that to guarantee differential privacy for tensor-valued data, one only needs to satisfy the constraint on Frobenius norms concerning covariance matrices of the additive noise $\mathcal{Z}$. We include a sketch proof here, and for the full proof, please refer to Appendix A-B.

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

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

which can be rewritten as

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

We express $\mathcal{Z}$ in terms of a SND tensor by Eq. (8) and define the following events:

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

where $\zeta^2(\delta)$ is defined in Lemma 3. By the definition of $\zeta^2(\delta)$ and Lemma 3, we have $\Pr(\{\mathcal{Z} \in \mathcal{O} - f(\mathcal{X})\} \cap \mathbf{R}_2) \leq \Pr(\mathbf{R}_2) \leq \delta$. And thus we only needs to find the sufficient conditions for the following inequality to hold:

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

Letting $\mathcal{O}' = \mathcal{O} - f(\mathcal{X})$ and $\Delta = f(\mathcal{X}) - f(\mathcal{X}')$, we find that

$$\begin{aligned}
&\Pr(\mathcal{Z} \in \mathcal{O}' \cap \mathbf{R}_1) \\
&\leq e^{\epsilon} \cdot \Pr(\mathcal{Z} \in (\mathcal{O}' + \Delta) \cap \mathbf{R}_1) \\
&\Leftrightarrow \frac{\int_{\mathcal{O}' \cap \mathbf{R}_1} \exp(-\frac{1}{2}\|\mathcal{Z} \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}\|^2) d\mathcal{Z}}{\int_{(\mathcal{O}'+\Delta) \cap \mathbf{R}_1} \exp(-\frac{1}{2}\|\mathcal{Z} \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}\|^2) d\mathcal{Z}} \\
&\leq e^{\epsilon} \Leftrightarrow \frac{1}{2}\|\Delta'\|^2 + \langle \Delta', \mathcal{Q}' \rangle \leq \epsilon,
\end{aligned}$$

where $\Delta' = \Delta \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}$, and $\mathcal{Q}' = \mathcal{Q} \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}$, $\forall \mathcal{Q} \in \mathcal{O}' \cap \mathbf{R}_1$. It is obvious that the last inequality consists of two parts and we will prove the bound for each.

For conciseness, we define $\phi = \|U_1^{-1}\|_F \|U_2^{-1}\|_F \ldots \|U_N^{-1}\|_F$. By Lemma 2, it can be proved that the first part satisfies

$$\begin{aligned}
\|\Delta'\|^2 &= \|\Delta \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}\|^2 && \text{(15a)} \\
&\leq s_2^2(f)\phi^2. && \text{(15b)}
\end{aligned}$$

Similarly, we derive the bound for the second part:

$$\langle \Delta', \mathcal{Q}' \rangle \leq s_2(f)\zeta(\delta)\phi. \tag{16}$$

By combining two Eq. (15b)(16), the sufficient condition is

$$s_2(f)^2\phi^2 + 2s_2(f)\zeta(\delta)\phi \leq 2\epsilon. \tag{17}$$

Note that ϕ can only be non-negative. By solving inequality Eq. (17), 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. ∎

By Thm. 1, one only need to satisfy the constraint on the covariance matrices of the additive noise to meet the differential privacy guarantee. Being aware that the constraint only deals with the Frobenius norm, we further provide the following corollary concerning the shape of the additive noise. The proof can be found in supplemental materials.

Corollary 1: TVG mechanism $\mathbf{TVG}_1(f(\mathcal{X})) = f(\mathcal{X}) + \mathcal{Z}_1$ *where* $\mathcal{Z}_1 \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, \ldots, \Sigma_N) \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *is* (ϵ, δ)*-differentially private. By reshaping* $f(\mathcal{X})$ *to* $f'(\mathcal{X})$*, and* $\mathcal{Z}_1$ *to* $\mathcal{Z}_2$*, where* $f'(\mathcal{X}), \mathcal{Z}_2 \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ *for any* $J_1 \ldots J_M = I_1 \ldots I_N$*, a new TVG mechanism can be defined:* $\mathbf{TVG}_2(f'(\mathcal{X})) = f'(\mathcal{X}) + \mathcal{Z}_2$*. If* $\mathcal{Z}_2 \sim \mathcal{TVG}(\mathbf{0}, \Gamma_1, \ldots, \Gamma_M)$ *where* $\Gamma_m \in \mathbb{R}^{J_m \times J_m}$, $\forall m \in [M]$*, then* $\mathbf{TVG}_2$ *satisfies the same* (ϵ, δ)*-differential privacy with* $\mathbf{TVG}_1$.

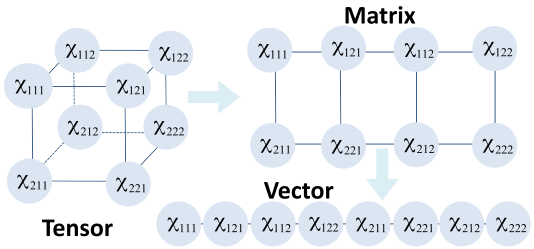

Fig. 1. An example of reshaping tensors.

See Appendix C-A for the proof. If the reshaped noise follows the tensor variate Gaussian distribution of zero mean, and the tensor-valued data can be reshaped accordingly, the resulting TVG mechanism satisfies the same differential privacy guarantee as the original one. We give an example of reshaping a tensor of the form $2 \times 2 \times 2$ into a 2×4 matrix, and then to a vector of length 8 in Fig. 1. This property builds a connection between different forms of tensors under the same differential privacy guarantee, and indicates that however the tensor is reshaped, the same differential privacy may be met with the same total amount of noise (when the reshaped noise also satisfies the TVG distribution).

Theorem 1 gives the condition that TVG mechanism 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.

Consider the following use case of TVG. In the task of sensitive image classification, convolutional neural networks are adopted as the model. The shape of the gradients of the first convolutional layer is $5 \times 5 \times 6$, representing 6 convolution kernels with each kernel of size 5×5. Queries to the gradients are adopted to update the neural network parameters during training. Since these gradients are generated by the sensitive training data, it requires TVG for preserving privacy at each release.

V. Unimodal Gaussian Noise

In this section, we present two special forms of TVG where noise bounds can be further improved, and thus a better utility can be achieved at the same privacy guarantee. In particular, we assume the additive noise $\mathcal{Z} \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, \ldots, \Sigma_N)$ is unimodal, which means one mode of the noise is directional noise, and all other modes are set to be i.i.d.

We first show an improvement over the general TVG by adding unimodal directional noise $\mathcal{Z}$. W.l.o.g., we assume mode-1 of $\mathcal{Z}$ is the directional noise and the rest are i.i.d., *i.e.,* $U_k = E_k, \forall k = 2, \ldots, N$ (E_k represents the identity matrix). Note that the result is not a special case of Thm. 1 by simply

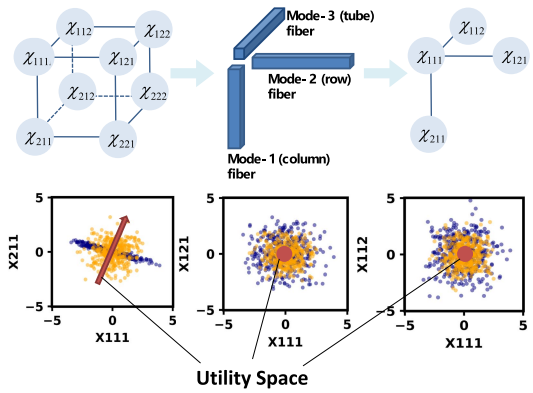

Fig. 2. The diagram of unimodal directional noise. Upper: a 3-order tensor $\mathcal{X} \in \mathbb{R}^{2 \times 2 \times 2}$ with column, row, and tube fibers. Lower: the three subfigures are projections of the same tensor in three different modes. The yellow dots represent a SND variable $\mathcal{N}$ with all its modes being i.i.d. The blue dots denote a variable $\mathcal{X} = \mathcal{N} \times_1 U_1$ with one mode being directional noise and the rest being i.i.d. The distribution of its column fiber is decided by the covariance matrix $\Sigma_1 = U_1 U_1^\top$. The red line (or dot) indicates the utility subspace of the data.

substituting U_k with E_k. We derive a new noise bound which is $I_2\ I_3 \ldots I_N$ times tighter than that of Thm. 1.

A. Unimodal Directional Noise

The unimodal directional noise (UDN) means that we select one mode of the noise to be directional noise. For example, for data records with different features, we can sample noise such that the noise applied to data records are independently drawn from the same distribution, whereas the noise added to features are correlated.

We further use a toy example in Fig. 2 to clarify the point. The upper figure shows a 3-order tensor-valued noise decomposed into 3 rank-1 tensors, which are respectively fibers of different modes. The lower figure describes the tensor noise by different modes. The yellow dots represent the SND tensor $\mathcal{N}$ whereas the blue dots denote $\mathcal{X} = \mathcal{N} \times_1 U_1$. In practice, such unimodal noise can be generated by applying mode-n matrix product to the SND tensor, which only changes the noise directions of mode-n fibers.

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 $U_n = E_n, \forall n = 2, \ldots, N$ to the left-hand side of the inequality (14), we get

$$\|{U_1}^{-1}\|_F^2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4 I_2 \ldots I_N \alpha^2}. \tag{18}$$

By our new theorem, the right-hand side bound can be improved by $I_2\ I_3 \ldots I_N$:

Theorem 2 (Unimodal Directional Noise): Given $U_k = E_k, \forall k = 2, \ldots, N$, TVG mechanism guarantees (ϵ, δ)-differential privacy if

$$\|{U_1}^{-1}\|_F^2 \leq \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4\alpha^2}. \tag{19}$$

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

Please refer to Appendix A-C for the proof. 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. (19). 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. We found that in the particular case where U_1 is a diagonal matrix, we can further improve the noise bound by I_1 times. Based on that, we derive the second special form of TVG with independent directional noise.

B. Independent Directional Noise

The independent directional noise (IDN) suggests that one mode of the noise is independent-directional, and other modes are i.i.d. For example, for data records with different features, independent directional noise indicates that the noise applied to different feature dimensions are independent while the noise applied to different data records are sampled independently from the same distribution. For most applications, the assumption is valid as we do not have any prior knowledge about the correlation between different features but can only assume independence.

W.l.o.g., we assume U_1 is a diagonal matrix and $U_k = E_k$, $\forall k = 2, \ldots, N$. At the first glance, independent directional noise is a special case of the unimodal directional noise, and Thm. 2 should be applied accordingly. However, we found that the conclusion of Thm. 2 is yet suboptimal and we can obtain a tighter bound under certain constraints. Note that by choosing U_1 to be diagonal, we in fact let the row-wise noise $W_{U_1} = E_1$, where E_1 is the identity matrix with the same size of U_1. Moreover, we assume that the data to be protected can be scaled to the same range, *i.e.,* each element of $f(\mathcal{X})$ is in range $[a, b]$.

Particularly, letting $U_1 = diag[\sigma_1, \ldots, \sigma_{I_1}] \in \mathbb{R}^{I_1 \times I_1}$ be a diagonal matrix, the probability density function (pdf) for $\mathcal{Z} \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, \ldots, \Sigma_N)$ can be written as

$$\Pr(\mathcal{Z}) = \frac{1}{(2\pi)^{I/2} |U_1|^{I/I_1}} \cdot \exp\left(-\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \frac{z_{i_1 i_2 \cdots i_N}^2}{2\sigma_{i_1}^2}\right), \tag{20}$$

where $I = I_1 I_2 \cdots I_N$. Given the pdf above, we can improve the bound for $\|U_1^{-1}\|_F$ with the following theorem.

Theorem 3 (Independent Directional Noise): Let $\mathcal{Z} \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, E_2 \ldots, E_N) \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ where $\Sigma_1 = U_1 U_1^\top$ and $U_1 = diag[\sigma_1, \ldots, \sigma_{I_1}] \in \mathbb{R}^{I_1 \times I_1}$. If we normalize each element of $f(\mathcal{X})$ to the same range $[a, b]$, TVG mechanism $\mathbf{TVG}(f(\mathcal{X})) = f(\mathcal{X}) + \mathcal{Z}$ guarantees (ϵ, δ)-differential privacy if

$$\|U_1^{-1}\|_F^2 \leq \frac{I_1}{\hat{s}_2^2(f)} \left(-\zeta(\delta) + \sqrt{\zeta^2(\delta) + 2\epsilon}\right)^2, \tag{21}$$

with $\zeta(\delta)$ defined in the Lemma 3 and $\hat{s}_2(f) = (b - a)\sqrt{I_1\ I_2 \ldots I_N}$.

The proof is based on the pdf of $\mathcal{Z}$ and can be found in Appendix A-D. It is worth mentioning that the proof utilizes the fact the tensor noise only uses one mode-1 matrix product, and thus eliminates many unnecessary relaxation of the bound. Hence it results in a tighter noise bound.

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

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

Obviously, we have improved the noise bound by I_1 times. An interesting observation is that, when we intentionally ignore some structural information of the tensor-valued noise, and rewrite the probability density function as an element-wise function Eq. (20), the noise bound actually improves.

VI. Privacy and Utility

We show in this section that TVG has a natural form to be optimized w.r.t. the utility subspace. If we know some dimensions/directions/modes of the tensor are more important than others, we can select noise directions/distributions such that less noise is inserted to the more important part of the tensor, at the cost of a higher level of noise adding to the less important part. By treating the TVG 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. Moreover, we discuss the value of the objective function w.r.t. the shape of the tensor.

A. Utility Objective

Depending on how data is applied in the downstream task, the utility subspace of data can be taken into account when choosing the specific noise distribution in TVG. Such utility subspace can be obtained directly from the downstream task, for example, it is known a prior some features of the data are critical to the task while others are not. Or in some cases, the relative importance can be drawn by analyzing the data itself, *i.e.,* a differentially-private SVD performed on the data to differentiate different directions. We give an example to explain this in Fig. 2.

In Fig. 2, we depict the utility subspace projected on the three modes in red. In the middle and right bottom figures, the utility subspace is a dot since its direction is vertical to mode-2 and mode-3. That indicates that mode-2 and mode-3 of the noise have little impact on the data utility. In the left bottom figure, the blue dots represent a directional noise which is vertical to the utility direction. The choice of directional noise (blue dots) is better than the i.i.d. one (yellow dots) in terms of the utility, as it applies less noise to the utility subspace. The example suggests that, we can create directional noise with consideration of utility.

Following the idea, we propose a scheme that, given the utility subspace, the noise distribution can be designed accordingly to bring minimum impact to the utility while satisfying the differential privacy constraint. The problem is formulated as a constrained optimization problem of which the objective is the expected amount of noise projected to the utility subspace, under the differential privacy constraint. For simplicity, we assume the utility subspace is linear, and other cases can be discussed accordingly.

Assume the task has linear utility subspace in all modes such that:

$$\mathcal{Y} = f(\mathcal{X}) \times_1 W_1 \times_2 W_2 \ldots \times_N W_N,$$

where $f(\mathcal{X}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $W_k \in \mathbb{R}^{J_k \times I_k}$ represents the utility subspace. $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is the output. To preserve privacy, we sample a noise $\mathcal{Z} \sim \mathcal{TVG}(\mathbf{0}, \Sigma_1, \Sigma_2 \ldots, \Sigma_N)$ and make predictions on the perturbed query result:

$$\hat{\mathcal{Y}} = [f(\mathcal{X}) + \mathcal{Z}] \times_1 W_1 \times_2 W_2 \ldots \times_N W_N.$$

According to [3], [6] 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:

$$\min_{U_1 \ldots U_N} \mathbb{E}\|\mathcal{Y} - \hat{\mathcal{Y}}\|^2 \Leftrightarrow \min_{U_1 \ldots U_N} \mathbb{E}\|\mathcal{Z} \times_1 W_1 \times_2 W_2 \ldots \times_N W_N\|^2. \quad (22)$$

By applying Lemma 1, we could transform the tensor to a matrix form. We turn the objective into:

$$\begin{aligned}
&\min_{U_1 \ldots U_N} \mathbb{E}\|\mathcal{Z} \times_1 W_1 \times_2 W_2 \ldots \times_N W_N\|^2 \\
&\quad \Leftrightarrow \min_{U_1 \ldots U_N} \mathbb{E}\|W_1 U_1 \mathcal{N}_{(1)} (W_N U_N \otimes \cdots \otimes W_2 U_2)^\top\|_F^2 \\
&\quad \Leftrightarrow \min_{U_1 \ldots U_N} \|W_1\ U_1\|_F^2 \|W_2\ U_2\|_F^2 \ldots \|W_N U_N\|_F^2 \\
&\quad \Leftrightarrow \min_{U_1 \ldots U_N} \|W_1\ W_{U_1} S_{U_1}\|_F^2 \ldots \|W_N W_{U_N} S_{U_N}\|_F^2,
\end{aligned}$$

where $U_k = W_{U_k} S_{U_k}$ is defined by the SVD of Σ_k in Sec. V-A and $S_{U_k} = diag(\sigma_{k1}, \ldots, \sigma_{kI_k})$. By letting $P_{ki} = \sum_{j=1}^{J_k} (W_k W_{U_k})_{ji}^2$, we can write our objective as

$$\min_{U_1 \ldots U_N} \prod_{k=1}^{N} \sum_{i=1}^{I_k} P_{ki} \sigma_{ki}^2. \quad (23)$$

B. Privacy Constrained Optimization

In light of the utility objective of Eq. (23), we propose optimized schemes which seek noise distributions that minimize the error while satisfying the differential privacy constraint. The privacy constraints are given by Eq. (14). If we substitute $U_k = W_{U_k} S_{U_k}$ and $S_{U_k} = diag(\sigma_{k1}, \ldots, \sigma_{kI_k})$, $k \in [N]$ to Eq. (14), and since W_{U_k} is a unitary matrix, we can rewrite the constraint and formulate the problem as the following optimization problem:

$$\begin{aligned}
&\min_{U_1 \ldots U_N} \prod_{k=1}^{N} \sum_{i=1}^{I_k} P_{ki} \sigma_{ki}^2, \\
&\text{s.t.} \prod_{k=1}^{N} \sum_{i=1}^{I_k} \frac{1}{\sigma_{ki}^2} \leq B, \quad (24)
\end{aligned}$$

where α, β, B are defined in Table. I.

The problem follows the form of a geometric program that can be solved efficiently. KKT conditions [30] can be applied and we obtain optimal solutions:

$$\prod_{k=1}^{N} \sigma_{ki_k}^2 = \frac{\prod_{k=1}^{N} \sum_{i=1}^{I_k} \sqrt{P_{ki}}}{\prod_{k=1}^{N} \sqrt{P_{ki_k}} B}, \quad \forall i_k \in [I_k],\ k \in [N], \tag{25}$$

We consider all σ_{ki} satisfying the above equation are optimal solutions to the problem. Given the optimal solutions, we can calculate the minimum value of the error as:

$$\text{Error}_{\text{TVG}}(\mathcal{Y}, \epsilon, \delta) = \frac{\left(\prod_{k=1}^{N} \sum_{i=1}^{I_k} \sqrt{P_{ki}}\right)^2}{B}. \tag{26}$$

The detailed derivation can be found in Appendix B.

1) Unimodal Directional Noise (UDN): For a straightforward illustration, we follow the assumption in Thm. 2 to add unimodal noise of which the mode-1 noise is directional noise. And $U_k = E_k, \forall k = 2, \ldots, N$. We could formulate the optimization problem w.r.t. UDN similarly:

$$\min_{U_1} \sum_{i=1}^{I_1} P_i \sigma_i^2, \quad \text{s.t.} \ \sum_{i=1}^{I_1} \frac{1}{\sigma_i^2} \leq B. \tag{27}$$

By Eq. (25), we could get the optimal solution:

$$\sigma_i^2 = \frac{\sum_{j=1}^{I_1} \sqrt{P_j}}{\sqrt{P_i} B}, \quad \forall i = 1, \ldots, I_1. \tag{28}$$

We can also obtain the optimal objective value by Eq. (26):

$$\text{Error}_{\text{UDN}}(\mathcal{Y}, \epsilon, \delta) = \frac{\left(\sum_{i=1}^{I_1} \sqrt{P_i}\right)^2}{B}. \tag{29}$$

To generate the unimodal Gaussian noise, we need the covariance matrix Σ_1. Differential privacy guarantee is already satisfied by calculating σ_i^2 from Eq. 28, and we still need to design W_{U_1}. We can simply adopt differentially-private SVD schemes [31], [32] to obtain the principal components of W as W_{U_1}. Then we compute the covariance matrix Σ_1 by $U_1 = W_{U_1} S_{U_1}$. Finally, a tensor-valued noise $\mathcal{Z}$ is sampled from $\mathcal{TVG}(\mathbf{0}, \Sigma_1, E_2 \ldots, E_N)$. The scheme is summarized in Alg. 1.

2) Independent Directional Noise (IDN): In the case of independent directional noise, we have $W_{U_1} = E_1$. Given Thm. 3 and Eq. (24), the optimized distribution of noise can be obtained by replacing the constraint of (27) with Eq. (21). The algorithm based on the independent directional noise is almost the same with Alg. 1 except that

$$B = \frac{I_1}{\hat{s}_2^2(f)} \left(-\zeta(\delta) + \sqrt{\zeta^2(\delta) + 2\epsilon}\right)^2,$$

and $W_{U_1} = E_1$.

3) Error and Tensor Shape: In our deduction, we found that in spite of the tensor shape, if the same differential privacy is guaranteed, TVG always yields the same error in the optimal case. That means, by TVG, the same differential privacy guarantee on the tensor always leads to the same utility, which has nothing to do with the shape of the tensor. Specifically, we have the following corollary, and the proofs are provided in Appendix C-B.

Algorithm 1 Generating Optimized Tensor Noise

Input: (a) privacy parameters ϵ, δ, (b) l_2 sensitivity $s_2(f)$, (c) the utility subspace $W \in \mathbb{R}^{J \times I_1}$, (d) the directions of the noise in mode-1 fibers $W_{U_1} \in \mathbb{R}^{I_1 \times I_1}$

Output: $f(\mathcal{X}) + \mathcal{Z}$

1: compute $\alpha,\ \beta$ as $\alpha = s_2^2(f)$, and $\beta = 2\zeta(\delta) s_2(f)$
2: compute $B = \frac{(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon})^2}{4\alpha^2}$
3: **for** $i \in \{1, \ldots, I_1\}$ **do**
4: $\quad P_i = \sum_{j=1}^{J} (W W_{U_1})_{ji}^2$
5: $\quad \sigma_i^2 = \frac{\sum_{j=1}^{I_1} \sqrt{P_j}}{\sqrt{P_j} B}$
6: **end for**
7: compute the diagonal matrix $S_{U_1} = diag(\sigma_1, \ldots, \sigma_{I_1})$
8: compute $U_1 = W_{U_1} S_1$
9: sampling $\mathcal{N}_{i_1\ i_2 \ldots i_N}$ from $\mathcal{N}(0, 1)$ for all $i_1, i_2, \ldots, i_N$
10: compute $\mathcal{Z} = \mathcal{N} \times_1 U_1$
11: **return** $f(\mathcal{X}) + \mathcal{Z}$

Corollary 2: Consider two linear models $\mathcal{Y} = f(\mathcal{X}) \times_1 W_1 \times_2 W_2 \ldots \times_N W_N \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\mathcal{Y}' = f'(\mathcal{X}) \times_1 W_1' \times_2 W_2' \ldots \times_N W_N' \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, where $f'(\mathcal{X})$ is reshaped from $f(\mathcal{X})$, and $W_k', k \in [N]$ are reshaped from $W_k, k \in [N]$, ensuring each element in $f'(\mathcal{X})$ is multiplied by the same coefficient as it is in $f(\mathcal{X})$. When (ϵ, δ)-differentially private TVG mechanism is respectively applied, we have $Error_{TVG}(\mathcal{Y}, \epsilon, \delta) = Error_{TVG}(\mathcal{Y}', \epsilon, \delta)$.

VII. Comparison With Other Mechanisms

For a better understanding of the position of this work in the current literature, we compare TVG with existing differential privacy mechanisms on high-dimensional data, mainly Matrix Mechanism [3] and Matrix Variate Gaussian [5]. Moreover, we show the applicability of TVG to Rényi differential privacy and general composition rules.

Different from mechanisms adding homogeneous noise in all directions, TVG inserts heterogeneous noise in different directions. Directional noise are also considered in MVG, however, TVG takes the downstream task into account in generating the directional noise, reducing the impact of noise on utility.

A. Comparison With Matrix Mechanism

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

Definition 10 (Matrix mechanism [3]): Given an $m \times n$ workload itmatrix 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:

$$\mathcal{M}_{\mathcal{K},A}(W, x) = W A^{+} \mathcal{K}(A, x)$$

where

$$\mathcal{K}(A, x) = Ax + \|A\| \tilde{b},$$

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

Note that W above is analogous to the linear utility subspace in TVG. 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 TVG is defined by Eq. (26) and the error minimization problem can be transformed into a convex one with a closed-form solution, which has a lower time complexity of $O(mn^2)$. While it is common for MM and TVG to search for an optimal noise direction, TVG has a closed-form solution and has broader application to tensor of all shapes.

B. 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:

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

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 TVG, MVG is also an additive noise scheme:

Definition 11 (MVG [5]): 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 constraint on Σ and Ψ:

Theorem 4 (MVG [5]): Let

$$\sigma(\Sigma^{-1}) = [\sigma_1(\Sigma^{-1}), \ldots, \sigma_m(\Sigma^{-1})]^T,$$
$$\sigma(\Psi^{-1}) = [\sigma_1(\Psi^{-1}), \ldots, \sigma_n(\Psi^{-1})]^T,$$

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:

$$\|\sigma(\Sigma^{-1})\|_2\|\sigma(\Psi^{-1})\|_2 \le \frac{(-\beta_0 + \sqrt{\beta_0^2 + 8\alpha_0\epsilon})^2}{4\alpha_0^2}, \quad (30)$$

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

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

$$\|\sigma(\Sigma^{-1})\|_2\|\sigma(\Psi^{-1})\|_2 \le \frac{\left(-\beta + \sqrt{\beta^2 + 8\alpha\epsilon}\right)^2}{4\alpha^2\sqrt{mn}} = \frac{16\epsilon^2}{(\beta + \sqrt{\beta^2 + 8\alpha\epsilon})^2\sqrt{mn}} \quad (31)$$

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. (30), α is reduced by $\frac{1}{s_2^2(f)}[(H_r + H_{r,1/2})\gamma^2 + 2\ H_r\gamma\, s_2(f)]$. And β is reduced by $(mn)^{\frac{1}{4}}H_r$ comparing with β_0 in Eq. (30). Overall, the right-hand side of inequality (31) is about H_r^2 times larger than that of inequality (30). 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 2 rather than the harmonic numbers in the proof.

C. Rényi Differential Privacy and Composition

We shortly prove that TVG is a general mechanism which also satisfies (α, ϵ)-Rényi Differential Privacy (RDP) [33], and can be directly applied composition as a primitive mechanism.

For any pair of adjacent datasets $\mathcal{X}$ and $\mathcal{X}'$, the Rényi divergence between query results on the two datasets is

$$\begin{aligned} &D_\alpha(\Pr(f(\mathcal{X}) + \mathcal{Z} \in \mathcal{O})\|\Pr(f(\mathcal{X}') + \mathcal{Z} \in \mathcal{O})) \\ &= \frac{1}{\alpha - 1}\mathbb{E}_{\Pr(f(\mathcal{X}'))}\left(\frac{\Pr(f(\mathcal{X}) + \mathcal{Z} \in \mathcal{O})}{\Pr(f(\mathcal{X}') + \mathcal{Z} \in \mathcal{O})}\right)^\alpha \\ &= \frac{1}{\alpha - 1}\int_{\mathbb{R}} \exp(-\frac{\alpha}{2}\|\mathcal{Z} \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}\|^2 \\ &\quad - \frac{1-\alpha}{2}\|\mathcal{Z} + \Delta \times_1 U_1^{-1} \times_2 \ldots \times_N U_N^{-1}\|^2)d\mathcal{Z} \\ &\le \alpha\|\Delta\|_F^2\|U_1^{-1}\|_F^2 \ldots \|U_N^{-1}\|_F^2 \end{aligned} \quad (32)$$

Therefore if $\alpha\|\Delta\|_F^2\|U_1^{-1}\|_F^2 \ldots \|U_N^{-1}\|_F^2 \le \epsilon$, TVG satisfies (α, ϵ)-RDP.

The TVG mechanism could also take advantage of advanced composition approaches such as the moments accountant technique [10] and optimal composition [11].

VIII. Evaluations

For fair evaluations of our proposed mechanisms, we conduct a series of experiments on a variety of query functions, models and datasets, and compare the results against a number of existing mechanisms.

A. Setup

1) Datasets and Tasks: We select typical learning tasks from multiple areas such as computer vision, data mining, text mining where the data are likely to be sensitive. Data can be private and proprietary to the data owner so that its release should preserve privacy. For example, the testing data can be sensitive personal images, and the neural network features computed on these data should not reflect any personal information. Another example is that the training data is private and the model trained on these data is supposed to preserve individual privacy.

For computer vision, we select three image classification tasks, respectively on MNIST [34], CIFAR-10 [35], and SVHN [36]. MNIST contains gray-scale images of handwritten digits, while CIFAR-10 and SVHN consist of real-world RGB images. Each dataset has 10 categories of images. For data mining tasks, we run classification tasks respectively on Adult [37], Credit [1]. Adult is a small-scale dataset on

[1]Credit dataset: https://www.kaggle.com/mlg-ulb/creditcardfraud.

which one can predict whether income exceeds a threshold. Credit is a highly unbalanced dataset that aims to recognize fraudulent credit card transactions. For text mining, we choose a binary classification task on IMDB [38] dataset. We also have CTG [37] to run the covariance estimation task.

2) Baselines and Metrics: We compare TVG with other differential privacy mechanisms dealing with high-dimensional data. The baselines include: i.i.d. Gaussian mechanism (Gaussian), Optimized mechanism (Optimized) [7], Matrix Variate Gaussian (MVG) [5], Matrix Mechanism (MM) [3], and Johnson-Lindenstrauss (JL) Transform [4]. We implement UDN and IDN as solutions to the optimized scheme (Sec. VI). In fact, Gaussian, Optimized and MM are designed for scalar-valued or vectorized queries, not specific to tensors or matrices. We follow the convention of [10] to flatten each tensor to a collection of its elements and apply the schemes. And we reshape tensors to matrices to apply matrix mechanism MVG. However, it should be noted that all these baselines do not have precise DP definition on tensors, instead their DP guarantee holds on the collection of elements, but not the original tensor values. Nevertheless, we still adopt them to show the practicality of our proposal. Particularly in the experiments, differentially-private SVD/PCA is implemented to seek directional noise in MVG. MM is only applied to small-scale datasets due to its high complexity. JL transform is only used in the covariance estimation.

For most of the experiments, we use testing accuracy as the utility metric to compare different mechanisms. For unbalanced dataset Credit, Area Under Curve(AUC) is adopted as the metric. In the experiment of covariance estimation, residual sum of square (RSS) is used as the metric. Let X be the clean data matrix and $\hat{X}$ be the perturbation result. $\hat{S} = \frac{1}{N}\hat{X}\hat{X}^\top$ is the covariance estimate for $S = \frac{1}{N}XX^\top$. Let $\{\hat{v}_i\}$ be the eigenvector of $\hat{S}$, and $\rho(\hat{v}_i) = \hat{v}_i^\top S \hat{v}_i$. RSS is defined as

$$RSS(\hat{S}) = \sum_i (\lambda_i - \rho(\hat{v}_i))^2, \tag{33}$$

where λ_i is the i^{th} eigenvalue of $\hat{S}$.

3) Privacy-Preserving Targets: For each task, we can have different privacy-preserving targets. We categorize the targets into three types: model gradients, training features, and testing data. For the first type, we assume we are training models on private training datasets and thus the model should be learned in a differentially-private manner. Specifically, we adopt the noisy stochastic gradient descent [10] as our optimization algorithm, and the query function returns differentially-private gradients. For the second type, we consider feature release on private training datasets. This often happens when the model is trained distributedly on more than one parties where one party extracts features from the raw data, and the other party trains on the released features. The query function releases differentially-private intermediate features, and the features would be further trained. For the third type, since the testing data is sensitive, the query function returns a differentially-private version of the prediction output on the testing data. We state the implementation detail by types as follows.

TABLE II
SETUP FOR STOCHASTIC GRADIENT DESCENT (SGD)

Dataset	MNIST	Credit	Adult
Model	LeNet	FCN	MLP
Training Data	55,000	199,364	32,561
Testing Data	5,000	85,443	16,281
Batch size	128	32	128
Lot size	128	256	256
Gradient Shape	$3,136 \times 512$	28×10	105×12
Clip Value	10^{-4}	0.1	1

B. Implementation Details

1) Type I: Private Stochastic Gradient Descent (SGD): We deploy experiments on datasets including MNIST, Credit and Adult. The setup is in Table II. Noisy stochastic gradient descent [10], [19] is adopted as the differentially-private optimization algorithm. We reiterate the procedures as follows:

1) Take a random sample from the training set with sampling probability q.

2) Compute the gradients on this sample.

3) Clip each gradient by its l_∞ norm. The clip value is C.

4) Average the gradients for a batch of samples, and **apply perturbation** to the averaged gradients.

5) Update the corresponding model parameters with the perturbed average gradients. And go back to 1).

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

$$f(\mathcal{X}) = \sum_{i=1}^{L} g(x_i), \tag{34}$$

where $g(x_i)$ is the gradient of the example x_i and L is the batch size. The shape of $f(\mathcal{X})$ is the same as the network. For example, in LeNet, the shape of gradients of the first convolution layer is $5 \times 5 \times 6$. For neighboring datasets $\{\mathcal{X}, \mathcal{X}'\}$, the l_2-sensitivity is

$$s_2(f) = \sup_{\mathcal{X},\mathcal{X}'} \left\| f(\mathcal{X}) - f(\mathcal{X}') \right\|_F = 2C\sqrt{I_1 I_2 \dots I_N}, \tag{35}$$

where C is the clip value in Table II, and only the gradient of the largest shape is reported. Actually, we are perturbing more gradients than that. Following the convention, we perturb gradients per batch and group several batches into a lot for adding noise. The average provides an unbiased estimator, the variance of which quickly decreases with the size of the group. We use the same composition scheme [11] for IDN, UDN and MVG to compose differentially-private gradients of each lot. Since the moments accountant method [10] is claimed as the state-of-the-art composition for the Gaussian mechanism, we adopt it respectively for Gaussian and MM. We also amplify the privacy guarantee [15] in the sampling step for all mechanisms in comparison. Please find the detailed composition and amplification scheme in Appendix D. Step **4)** is where we implement different privacy mechanisms. In UDN, MVG and MM, we set W to identity matrix E since the utility subspace is unknown. Directional noise can be considered in the future if the sensitivity of each gradient is known.

TABLE III
SETUP FOR PRIVATE TRAINING FEATURE AND TESTING DATA

Dataset	MNIST	CIFAR-10	SVHN	IMDB	CTG
Model	LeNet	VGG-16	AlexNet	BiLSTM	Covariance Estimation
Training Data	50,000	50,000	73,257	25,000	–
Testing Data	10,000	10,000	26,032	25,000	2126
Feature Tensor Shape	$256 \times 16 \times 5 \times 5$	$256 \times 512 \times 1 \times 1$	$256 \times 256 \times 4 \times 4$	256×150	2126×21

TABLE IV
COMPARISON OF ERROR

Method	$\mathbb{E} \quad \|\mathcal{Z}\|^2$				
	Gaussian	MVG	UDN	IDN	Optimized
Type I	–	$I_1 I/B_{MVG} \approx O(I_1 I^3 \ln^2(I_1+1))$	$I_1 I/B_{UDN} \approx O(I_1 I^3)$	$I_1 I/B_{IDN} \approx O(I^3)$	$I\sigma_g^2 \approx O(I^2)$
Type II	$I\sigma^2 \approx O(I^3)$	$I_1 I/B_{MVG} \approx O(I_1 I^3 \ln^2(I_1+1))$	$I_1 I/B_{UDN} \approx O(I_1 I^3)$	$I_1 I/B_{IDN} \approx O(I^3)$	-
Type III	$I\sigma^2 \approx O(I^3)$	$I_1 I/B_{MVG} \approx O(I_1 I^3 \ln^2(I_1+1))$	$\frac{I\left(\sum_{i=1}^{I_1}\sqrt{P_i}\right)^2}{(I_1 B_{UDN})} \approx O(I_1 I^3)$	$\frac{I\left(\sum_{i=1}^{I_1}\sqrt{P_i}\right)^2}{(I_1 B_{IDN})} \approx O(I^3)$	-

2) Type II: Private Training Features: Four datasets are adopted: MNIST, CIFAR-10, SVHN, and IMDB. We consider all of their training data as private and aim to protect the training data privacy. We modify each model by replacing its activation function with $\tanh(\cdot)$ to normalize the released intermediate-layer feature to the range of $(-1, 1)$. For all datasets, stochastic gradient descent is adopted as the optimizer and in each iteration, a batch of 256 input instances are randomly selected to train the optimizer. Configuration details are given in Table III.

a) Query function: We set the query function as the identity query $f(\mathcal{X}) = \mathcal{X}$. $\mathcal{X}$ is the training features in the experiment. For example, the feature tensor shape over MNIST dataset is $256 \times 16 \times 5 \times 5$, which is a 4-order tensor. 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

$$s_2(f) = \sup_{\mathcal{X},\mathcal{X}'} \left\|\mathcal{X} - \mathcal{X}'\right\|_F = 2\sqrt{I_1 I_2 \dots I_N} \tag{36}$$

The released features are perturbed once before training. For UDN, IDN, MVG and MM, we set the utility subspace W to E since no prior knowledge about the training data is known. For the i.i.d. Gaussian and MM, l_2-sensitivity is computed for each element and for UDN, IDN and MVG, their l_2-sensitivity is computed directly on the feature tensor. This is due to the composition of different mechanism, which could know more detail theories in supplemental materials Appendix D.

3) Type III: Private Testing Data: Datasets include MNIST, CIFAR-10, SVHN, IMDB, and CTG. The experimental setup is the same as Type II, shown in Table III. Each model is trained on the unperturbed training dataset and tested on private testing data. We choose the output from the last convolutional layer of each model as the private testing features. The query is also the identity function.

a) Query function: Similarly, we set the query function as the identity query $f(\mathcal{X}) = \mathcal{X}$ where $\mathcal{X}$ is the testing feature in the experiment. The l_2 sensitivity is the same with Type II experiments. On Cora, we set the query function as the covariance matrix query $f(\mathcal{X}) = \frac{1}{N}\mathcal{X}\mathcal{X}^T$, where $\mathcal{X} \in [-1, 1]^{26 \times 2126}$. For neighboring datasets $\{\mathcal{X}, \mathcal{X}'\}$, the l_2 -sensitivity is

$$s_2(f) = \sup_{\mathcal{X},\mathcal{X}'} \frac{\left\|x_j x_j^T - x'_j x'^T_j\right\|_F}{2126} = \frac{2\sqrt{\sum_{i=1}^{26^2} x_j(i)^2}}{2126} = \frac{52}{2126}.$$

Since it is known how each feature is processed in the model, we set the utility subspace in UDN, IDN, MVG and MM as the model weights associated with the released features. We use Alg. 1 to generate noise in UDN and IDN. As the directional matrix W_{U_1} of UDN and MVG could be any orthogonal matrix, here we choose the right-singular matrix of W as W_{U_1}. In MVG, we adopt the *binary precision allocation strategy* from [5] to decide the importance of different directions. Due to the time complexity issue, we only implement MM on IMDB and CTG. We calculate the l_2-sensitivity for each mechanism in the same way as in Type II.

C. Experimental Results

Before delving into the experimental results, we first present theoretical analysis of the expected error of each differential privacy mechanism. The theoretical results are presented in Table IV. In the table, $I = I_1\ I_2 \dots I_N$ is the product of each dimension of each order, and I_1 is the dimension of the first order of the tensor where we apply directional noise. For Type I Gaussian, we adopt the same σ_g in the differentially-private SGD in [10]. For Type II and III Gaussian, σ is the standard deviation of the i.i.d Gaussian distribution in the scalar-valued Gaussian mechanism [8], *i.e.,* $\sigma \geq c\Delta_2(f)/\epsilon$ and $c^2 > 2\ln(1.25/\delta)$. $B_{MVG}, B_{UDN}, B_{IDN}$ are respectively calculated by the right-hand side of Eq. (30), (19) and (21).

For ease of understanding the theoretical results, we give approximation of the expected noise magnitude in each case besides the exact value. As we found, for private SGD, Optimized has the least perturbation error among all but without an accurate DP guarantee on tensors. This is because 'flattening' operation on tensors obfuscates tensors of the same collection of elements but different shapes. And for the private training and testing data, IDN and Gaussian have the same order of expected error. However, the results are given without considering directional noise. If the directional

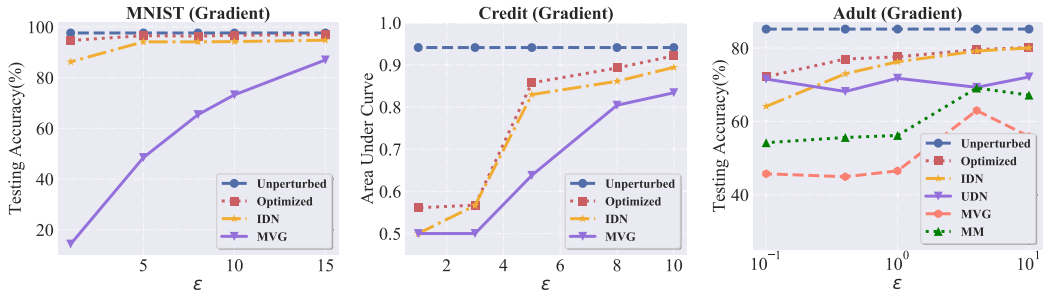

Fig. 3. Type I: Private stochastic gradient descent (SGD). IDN shows close performance to Optimized which does not have precise DP guarantee on tensor data. UDN is slightly inferior to IDN but still better than MM and MVG.

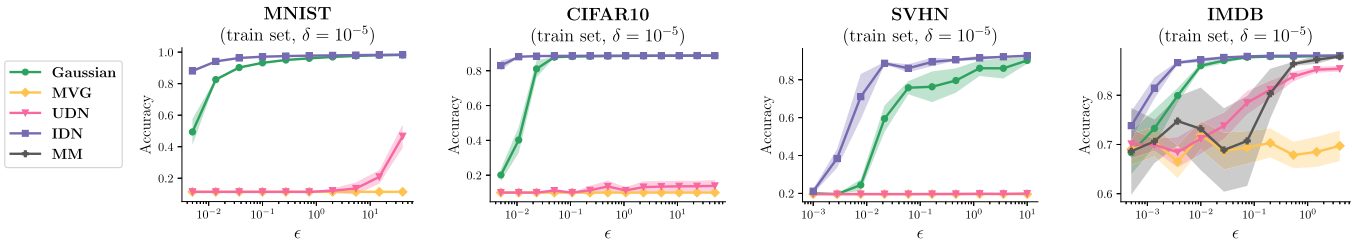

Fig. 4. Type II: Private training features. IDN performs best overall, followed by the Gaussian mechanism. Their performance is close to each other except that IDN yields higher accuracies in the high privacy regime (ϵ is small). UDN and MVG have worse performance. As ϵ grows, UDN gradually gains advantage over MVG.

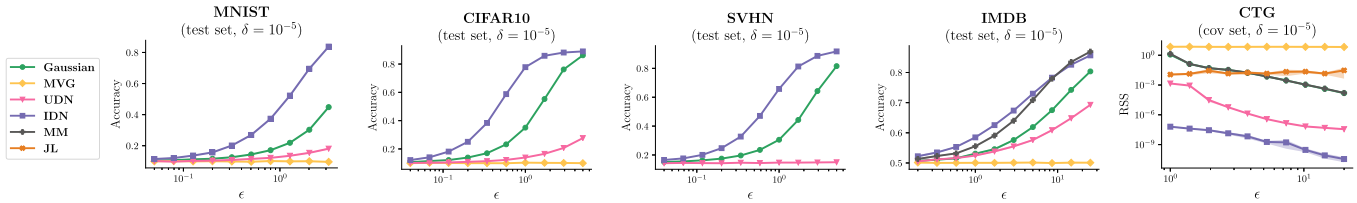

Fig. 5. Type III: Private testing data. IDN has the best performance overall, followed by the i.i.d. Gaussian on image classification datasets. MM has an inferior performance to IDN overall. UDN is worse than Gaussian but no worse than MVG. The barely visible error bars indicate that the experimental results are highly consistent.

noise is applied, IDN is supposed to incur less error. JL and MM are missing from Table. IV since the error is data-dependent or not deterministic.

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

1) Type I Results: Fig. 3 reports accuracies under different ϵs when fixing $\delta = 10^{-5}$. '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 agrees with the theoretical error analysis. In the experiment on Adult, the performance of MM is a little better than MVG, but is still worse than UDN. The performance of IDN degrades slightly from Optimized whereas both IDN and UDN outperform the rest of the baselines.

2) Type II Results: Fig. 4 reports accuracies under different ϵs when fixing $\delta = 10^{-5}$. Compared with Type I results, the relative performance of UDN is worse in Type II since the l_2-sensitivity is computed on each element, rather than the entire tensor, and thus leading to a looser noise bound. Actually, the performance of all mechanisms is in accords with the theoretical results in Table IV. The i.i.d. Gaussian has similar accuracy performance with IDN, since their noise is of the same order of magnitude. But IDN still has superior performance than Gaussian in the regime where ϵ is small. In the experiment on IMDB, we could observe that the performance of MM is poorer than IDN and Gaussian mechanism, but comparable with UDN. Moreover, the variation of accuracy on MM is quite substantial, indicating unstable performance across different runs of experiments.

3) Type III Results: Fig. 5 reports accuracies under different ϵs when fixing $\delta = 10^{-5}$. Different from Type II, the accuracy results on private testing data are less fluctuated than that on private training features, and thus reveal the general trend better. Overall, the trend is consistent with Table IV. The accuracy gap between IDN and Gaussian is larger than that of Type II results, and it may be because directional noise from the optimized scheme is taken into account. Although UDN's performance is inferior to IDN and Gaussian, it still has better performance than MVG, especially when $\epsilon \geq 1$.

On the IMDB dataset, we found that MM performs exceedingly well when ϵ is large, almost equivalent to IDN. But on CTG, MM only performs as good as the Gaussian mechanism,

which is worse than IDN. We reckon that the performance of MM largely depends on a carefully chosen utility subspace W, *i.e.,* how well W describes the linear utility subspace. Otherwise, MM has a similar performance with the Gaussian. The results on CTG clearly distinguish different mechanisms for all levels of privacy. JL or MM are worse than UDN and IDN in terms of RSS.

IX. Conclusion

In this paper, we propose a new differential privacy mechanism for tensor-valued queries, called TVG. We show that TVG enjoys a tighter noise bound than previous works, and thus has better utility. Two special forms of TVG are discussed: unimodal directional noise (UDN) and independent directional noise (IDN). We found that by removing some structural information, UDN and IDN progressively achieve better utility under the same differential privacy guarantee. In practical settings where the utility subspace of the data is known, we further improve the utility by implementing optimized schemes over differential privacy constraints. Closed-form solutions to the optimization problems are derived. Experimental results under a variety of settings have shown the practicality of TVG mechanisms.

References

[1] B. Balle and Y.-X. Wang, "Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 394–403.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2006, pp. 265–284.

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

[4] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "The Johnson-Lindenstrauss transform itself preserves differential privacy," in *Proc. IEEE 53rd Annu. Symp. Found. Comput. Sci.*, Oct. 2012, pp. 410–419.

[5] T. Chanyaswad, A. Dytso, H. V. Poor, and P. Mittal, "MVG mechanism: Differential privacy under matrix-valued query," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 230–246.

[6] Q. Geng and P. Viswanath, "Optimal noise adding mechanisms for approximate differential privacy," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 952–969, Feb. 2016.

[7] L. Xiang, J. Yang, and B. Li, "Differentially-private deep learning from an optimization perspective," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 559–567.

[8] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[9] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2016, pp. 635–658.

[10] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 308–318.

[11] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, Jun. 2017.

[12] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1656–1665.

[13] M. Lécuyer, R. Spahn, K. Vodrahalli, R. Geambasu, and D. Hsu, "Privacy accounting and quality control in the sage differentially private ML platform," in *Proc. 27th ACM Symp. Operating Syst. Princ.*, Oct. 2019, pp. 181–195.

[14] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 332–349.

[15] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2010, pp. 437–454.

[16] 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 *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2407–2416.

[17] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[18] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Scalable private learning with PATE," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018.

[19] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 245–248.

[20] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.

[21] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci.*, Oct. 2014, pp. 464–473.

[22] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 2719–2728.

[23] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, Jul. 2012.

[24] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proc. 30th AAAI Conf. Artif. Intell.*, vol. 16, Feb. 2016, pp. 1309–1316.

[25] J. Zhang, K. Zheng, W. Mou, and L. Wang, "Efficient private ERM for smooth objectives," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*. Menlo Park, CA, USA: AAAI Press, 2017, pp. 3922–3928.

[26] Y. Wang and A. Anandkumar, "Online and differentially-private tensor decomposition," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 29, 2016.

[27] H. Imtia and A. D. Sarwate, "Improved algorithms for differentially private orthogonal tensor decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2201–2205.

[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[29] T. G. Kolda, "Multilinear operators for higher-order decompositions," Sandia Nat. Laboratories, Albuquerque, NM, USA, Tech. Rep., 2006.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[31] A. Blum, C. Dwork, F. McSherry, and K. Nissim, "Practical privacy: The SuLQ framework," in *Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst. (PODS)*, 2005, pp. 128–138.

[32] K. Chaudhuri, A. Sarwate, and K. Sinha, "Near-optimal differentially private principal components," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 989–997.

[33] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 263–275.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

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

[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," Tech. Rep., 2011.

[37] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[38] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Portland, OR, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015