HHS Public Access

Author manuscript

IEEE Trans Pattern Anal Mach Intell. Author manuscript; available in PMC 2023 February 01.

Published in final edited form as:

IEEE Trans Pattern Anal Mach Intell. 2022 February ; 44(2): 1002–1019. doi:10.1109/TPAMI.2020.3015859.

Model-Protected Multi-Task Learning

Jian Liang,
Department of Automation, Tsinghua University, State Key Laboratory of Intelligent Technologies and Systems, Tsinghua National Laboratory for Information Science and Technology, Beijing, 100084, China

Ziqi Liu,
Department of Computer Science, Xi'an Jiaotong University, Xi'an, 710049, China

Jiayu Zhou,
Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824, USA

Xiaoqian Jiang,
Department of Biomedical Informatics, University of California, San Diego, La Jolla, CA, 92093, USA

Changshui Zhang [Fellow, IEEE],
Department of Automation, Tsinghua University, State Key Laboratory of Intelligent Technologies and Systems, Tsinghua National Laboratory for Information Science and Technology, Beijing, 100084, China

Fei Wang
Department of Healthcare Policy and Research, Weill Cornell Medical College, New York City, NY, 10065, USA

Abstract

Multi-task learning (MTL) refers to the paradigm of learning multiple related tasks together. In contrast, in single-task learning (STL) each individual task is learned independently. MTL often leads to better trained models because they can leverage the commonalities among related tasks. However, because MTL algorithms can "leak" information from different models across different tasks, MTL poses a potential security risk. Specifically, an adversary may participate in the MTL process through one task and thereby acquire the model information for another task. The previously proposed privacy-preserving MTL methods protect data instances rather than models, and some of them may underperform in comparison with STL methods. In this paper, we propose a privacy-preserving MTL framework to prevent information from each model leaking to other models based on a perturbation of the covariance matrix of the model matrix. We study two popular MTL approaches for instantiation, namely, learning the low-rank and group-sparse patterns of the model matrix. Our algorithms can be guaranteed not to underperform compared with STL methods. We build our methods based upon tools for differential privacy, and privacy guarantees, utility bounds are provided, and heterogeneous privacy budgets are considered. The

liangjian12@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn.

experiments demonstrate that our algorithms outperform the baseline methods constructed by existing privacy-preserving MTL methods on the proposed model-protection problem.

Keywords

Multi-Task Learning; Model Protection; Differential Privacy; Covariance Matrix; Low-rank Subspace Learning

1 Introduction

Multi-task learning (MTL) [12] refers to the paradigm of learning multiple related tasks together. In contrast, single-task learning (STL) refers to the paradigm of learning each individual task independently. MTL often leads to better trained models because the commonalities among related tasks may assist in the learning process for each specific task. For example, an infant's ability to recognize a cat might help in developing the ability to recognize a dog. In recent years, MTL has received considerable interest in a broad range of application areas, including computer vision [48, 75], natural language processing [2] and health informatics [64, 68]. The key to MTL is to relate learning tasks via a shared representation, which, in turn, benefits the tasks to be learned. Each possible shared representation encodes certain assumptions regarding task relatedness. Because MTL approaches explore and leverage the commonalities among related tasks within the learning process, either explicitly or implicitly, they pose a potential security risk. Specifically, an adversary may participate in the MTL process through a participating task, thereby acquiring the model information for another task. A predictive model may identify a causality between system inputs and outputs. Knowledge of the causality makes it possible or easier for an adversary to change a system input to trigger an irrational or even harmful output, which can be regarded as a *generalized adversarial attack*. The system could be a predictive system for traffic-sign recognition or face identification, as studied by recent adversarial-attack approaches [65, 51, 46, 72, 57, 56, 52]. As noted by Finlayson et al. [26], adversarial attacks on medical machine learning are increasingly rampant and easy to implement (e.g., by simply rotating a picture to upload to a specific angle), especially in medical fraud which is a $250 billion industry. Therefore, model-information leakage during an MTL process could realize or escalate such adversarial attacks to increase fraudulent medical costs or insurance claims. In addition, the aforementioned system could well be a real human body. For example, consider *personalized predictive modeling* [53, 74], which has become a fundamental methodology in health informatics. This type of modeling builds a custom-made model for each patient. In modern health informatics, such a model may include patient disease conditions/causes (e.g., which foods can induce an allergic reaction in a patient). If such information were to be leaked, an adversary might use the information to deliberately introduce the food into a patient meal to trigger an allergic reaction.

Because of the concerns discussed above, a secure training strategy must be developed for MTL approaches to prevent information from each model leaking to other models. In this paper, we propose a model-protected multi-task learning (MP-MTL) approach that enables the joint learning of multiple related tasks while simultaneously preventing model leakage

for each task. Our approach is based on *differential privacy* [22] which provides a strong, cryptographically motivated definition of privacy based on rigorous mathematical theory and has recently received significant research attention due to their robustness to known attacks [16, 27]. This scheme is useful when one wishes to prevent potential attackers from acquiring information on any element of the input dataset based on a change in the output distribution.

To focus on the main issue, our MP-MTL method is designed simply based on linear multi-task models [39, 49]. We assume that the model parameters are learned by minimizing an objective that combines an average empirical prediction loss and a regularization term. The regularization term captures the commonalities among the different tasks and couples their model parameters. The solution process for this type of MTL method can be viewed as a recursive two-step procedure. The first step is a decoupled learning procedure in which the model parameters of each task are estimated independently using some precomputed shared information among tasks. The second step is a centralized transfer procedure in which some of the information shared among tasks is extracted for distribution to each task for the decoupled learning procedure in the next step. Our MP-MTL mechanism protects the models by adding perturbations during the second step. Note that we assume a curator that collects models for joint learning but never needs to collect task data. We develop a rigorous mathematical definition of the MP-MTL problem and propose an algorithmic framework to obtain the solution. We add perturbations to the covariance matrix of the parameter matrix because the tasks' covariance matrix is widely used as a fundamental source from which to extract useful knowledge to share among tasks [49, 77, 39, 5, 79, 64], which is the key observation that enables the proposed framework. The usage of the perturbed covariance matrix depends on the specific MTL method applied. Consequently, our technique can cover a wide range of MTL algorithms and is generally applicable for many optimization schemes, such as proximal gradient methods [39, 49], alternating methods [5] and Frank-Wolfe methods [37]. We introduce Wishart noise into the covariance matrix to ensure model security. Fig. 1 illustrates the key ideas of the main framework.

We further develop two concrete approaches as instantiations of our framework, each of which transforms an existing MTL algorithm into a private version. Specifically, we consider two popular types of basic MTL models: 1) a model that learns a low-rank subspace by means of a trace norm penalty [39] and 2) a model that performs shared feature selection by means of a group-ℓ_1 ($\ell_{2,1}$ norm) penalty [49]. We first choose to learn a low-rank subspace of the model matrix because it is typical to learn a shared representation, which is the key to relating tasks in MTL. In addition, it is also typical to learn correlated but different parameters for multiple models that share the same model structure, which is also commonly encountered in MTL. This is a typical/mainstream approach in MTL, as stated by Zhang and Yang [76] in their MTL survey as well as by Su et al. [63] and Gu et al. [31]; (see, e.g., Ando and Zhang [4], Chen et al. [17], Xu and Lafferty [71], Han and Zhang [34], and Zhen et al. [78]). On the other hand, learning a shared feature selection is also typical in MTL and can be regarded as learning a specific type of low-rank subspace. In both cases, we instantiate our framework by approximating *proximal gradient descent methods*, which were presented by Ji and Ye [39] and Liu et al. [49]. The covariance matrix is used to build a linear transform matrix used to project the models into new feature

subspaces; then, the most useful subspaces are selected. The projection matrix is related to the generalized inverse of the singular value matrix (or the diagonal matrix) of the perturbed covariance matrix for the instantiation with a trace norm penalty (or the instantiation with a group-ℓ_1 penalty). Wishart noise is positive definite; thus, it renders the singular values of the perturbed covariance matrix "larger" and those of the generalized inverse "smaller". Consequently, under a high noise level, the projection matrix tends to be an identity matrix that shares no information between models but keeps the models intact. This means that our algorithms will not underperform in comparison with STL methods under high noise levels; hence, participation in the joint learning process has no negative effect on training any task model. Approximating the proximal operators with Wishart noise makes it possible for the added noise to destroy the covariance matrix without destroying the projected models, which is a key observation that enables our instantiated algorithms.

We provide privacy guarantees. Utility analyses are also presented for both convex and strongly convex prediction loss functions and for both the basic and accelerated proximal-gradient methods. Furthermore, we consider heterogeneous privacy budgets for different iterations of our algorithms and present a utility analysis for privacy-budget allocation. We also validate the effectiveness of our approach on both benchmark and real-world datasets.

Our proposed MP-MTL algorithms fall into a larger scope of MTL algorithms based on differential privacy (i.e., MTL algorithms with randomness added using differential privacy tools). Within this scope, Pathak et al. [58] proposed a differentially private aggregation (DP-AGGR) method and provided the associated privacy guarantee and utility analysis. However, DP-AGGR works for homogeneous tasks only, i.e., the coding procedures for both features and targets are the same for different tasks. In contrast, Gupta et al. [33] proposed a differentially private multi-task relationship learning (DP-MTRL) method to handle heterogeneous tasks. However, they did not provide utility analyses. In addition, they provided the privacy guarantee for each iteration only, neglecting the additional privacy loss due to multiple iterations of their algorithm (see Kairouz et al. [42]). Moreover, their privacy guarantee holds only if the adopted loss function has a closed-form solution (e.g., the least-square loss). Both DP-AGGR and DP-MTRL protect a single data instance instead of the model of each task, and neither of them considers privacy-budget allocation or the associated utility analyses. Therefore, to our knowledge, we are the first to consider allocation strategies for heterogeneous privacy budgets and the associated utility analyses. In addition, we are also the first to present a utility analysis for heterogeneous tasks. Moreover, we are also the first to provide privacy guarantees for heterogeneous tasks with loss functions that may have no closed-form solutions (e.g., logistic loss).

Since our method is the first to address the model-protected problem in the MTL setting, we construct baseline MP-MTL methods for comparison by exploiting existing privacy-preserving MTL methods, which are referred to as instance-protected multi-task learning (IP-MTL) methods because they protect the security only of data instances rather than that of models. The IP-MTL methods are transformed into their respective MP-MTL methods by directly enforcing the *group privacy* [22] of the entire dataset coming from a single learning task. The experimental results demonstrate that our proposed algorithms outperform the constructed baseline MP-MTL methods.

Our contributions are highlighted as follows.

- We are the first to propose and address the model-protection problem in an MTL setting.
- We develop a general algorithmic framework to solve the MP-MTL problem to obtain secure estimates of the model parameters. We derive concrete instantiations of our algorithmic framework for two popular types of MTL models, namely, models that learn the low-rank and group-sparse patterns of the model matrix. By approximating the proximal operators with Wishart noise, we can guarantee that our algorithms will not underperform in comparison with STL methods under high noise levels.
- We provide privacy guarantees. We also present utility analyses for both convex and strongly convex prediction loss functions and for both the basic and accelerated proximal-gradient methods. Heterogeneous privacy budgets are considered for different iterations of our algorithms, and a utility analysis for privacy-budget allocation is presented.
- Within the larger scope of MTL algorithms based on differential privacy, to the best of our knowledge, we are the first to 1) consider allocation strategies for heterogeneous privacy budgets and the associated utility analyses, 2) present a utility analysis for heterogeneous tasks, and 3) provide privacy guarantees for heterogeneous tasks with loss functions that may have no closed-form solutions.
- For comparison, we construct baseline MP-MTL methods using IP-MTL methods (i.e., existing privacy-preserving MTL methods). The experiments demonstrate that our proposed algorithms significantly outperform the baseline methods.

The remainder of this paper is organized as follows. Section 2 discusses related works and definitions of differential privacy. Section 3 introduces the background on MTL problems and the definition of the proposed model-protection problem. The algorithmic framework and concrete instantiations of the proposed MP-MTL method are presented in Section 4, along with the analyses of utility and privacy-budget allocation. Section 5 presents an empirical evaluation of the proposed approaches, and Section 6 provides conclusions.

2 Related Works

2.1 Privacy-preserving MTL Approaches

Few privacy-preserving MTL approaches have been proposed to date [50, 9, 58, 61, 33]. Moreover, such approaches protect only the security of data instances rather than that of models. A typical focus of research is distributed learning [50, 9], in which the datasets for different tasks are distributively located. The local task models are trained independently using their own datasets before being aggregated and injected with useful knowledge shared across tasks. Such a procedure mitigates the privacy problem by updating each local model independently. However, these methods do not provide theoretical privacy guarantees.

In contrast, Pathak et al. [58] proposed the DP-AGGR method in a distributed learning scheme with privacy guarantees in which they first trained local models distributively and then averaged the models of the tasks before adding noise based on the output perturbation method of [20]. The final solution for each task is the averaged model. However, because this method performs only averaging, it has a limited ability to address more complicated task relations such as low-rank [4], group-sparse [66], clustered [30] or graph-based [77] task structures. Considering the task relationships, Gupta et al. [33] proposed the DP-MTRL method to transform the multi-task relationship learning proposed by Zhang and Yeung [77] into a differentially private version. They adopt the output perturbation method. In addition to their limitations mentioned in the introduction, their privacy-preserving MTL methods underperformed on their synthetic datasets compared with non-private STL methods (which can guarantee optimal privacy against cross-task information leakage), which suggests that there is no reason to use their proposed methods. Both DP-AGGR and DP-MTRL add noise directly to the models, which is unnecessary to avoid information leakage across tasks and may jeopardize the utility of the algorithms.

2.2 Related Works of Differential Privacy

Several related definitions of privacy are listed as follows.

Joint differential privacy.—In a game theory setting, Kearns et al. [45] and Kearns et al. [44] proposed to guarantee that for each player, the output to other players reveals little input information about that player.

One-analyst-to-many-analyst privacy.—In a database query setting, Hsu et al. [36] proposed a method for protecting the privacy of all the queries of one analyst from other analysts.

Both joint differential privacy and one-analyst-to-many-analyst privacy are similar to our proposed MP-MTL (defined in Definition 6) from a very high-level perspective: when the input of one object is replaced, the joint distribution of the output of other objects will not be significantly affected. However, both definitions are different from our proposed MP-MTL in both the modeling objects and the input characteristics. We discuss the detailed differences in the supplementary material (Appendix C).

Differential privacy for streams.—This definition considers continual/iterative observations, and was proposed by Chan et al. [13] and Dwork et al. [21]. Because machine learning algorithms are generally iterative, this paper also involves the concept of iteration in the definitions of MP-MTL and IP-MTL algorithms, and it simply aims to directly use composition theorems of differential privacy to bound the total privacy-budgets.

Local private learning algorithms.—This definition was proposed by Kasiviswanathan et al. [43] to characterize that each individual’s data are added independent randomness before further processing. Algorithms that accomplish this task are referred to as input perturbation. The idea can be adopted to propose possible solutions to the MP-MTL problem. For example, independent randomness can be added to each task model. Both DP-MTRL and DP-AGGR can be regarded as examples, although they protect data

instances rather than models. However, because local private learning algorithms have some limitations (e.g., they may require exponentially more data than do general private algorithms [43]), we did not adopt this idea in this paper.

Secure multi-party computation (SMC).—In Section 1, we assume the use of a trusted curator that collects the task models, and this assumption raises privacy concerns in untrusted curator settings. Such concerns are related to the demand for SMC [58, 28], the purpose of which is to avoid the leakage of data instances to the curator. We present an extended framework that considers SMC in the supplementary material (Appendix B).

In addition to the above related definitions, the sample-aggregate framework proposed by Nissim et al. [55] is also related. This framework first randomly partitions a database into multiple small databases, executes the same algorithms on all the sub-databases, aggregates all the outputs, and finally adds randomness according to the smooth sensitivity of the aggregation function. For model-protection, this framework may be applicable for homogeneous tasks (which is the setting considered by DP-AGGR) to extend their method for empirical risk minimization: instead of data instances, tasks can be randomly partitioned into groups to perform the above procedures. However, applying this framework to heterogeneous tasks is not trivial.

2.3 Methods that Privately Release the Covariance Matrix

Several methods have been proposed to privately release the covariance matrix [41, 23, 10]. Considering an additive noise matrix, based on our utility analysis, the overall utility of the MTL algorithm depends on the spectral norm of the noise matrix. A list of the bounds on the spectral norms of additive noise matrices can be found in Jiang et al. [41]. We choose to add Wishart noise [41] to the covariance matrix for four reasons: (1) For a given privacy budget, this type of noise matrix has a better spectral-norm bound than does a Laplace noise matrix [41]. (2) Unlike a Gaussian noise matrix, which enables an (ϵ, δ)-private method with a positive δ, this approach enables an $(\epsilon, 0)$-private method and can be used to build an iterative algorithm that is entirely $(\epsilon, 0)$-private, which provides a stronger privacy guarantee. (3) Unlike the Gaussian and Laplace noise matrices cases, the Wishart noise matrix is positive definite and can be exploited to guarantee that our method will not underperform compared with STL methods under high noise levels. (4) This approach allows arbitrary changes to any task, unlike the method of Blocki et al. [10].

3 Preliminaries and the Proposed Problem

In this section, we introduce the MTL background and the definition of model-protection problems for MTL. The notations and symbols that will be used throughout the paper are summarized in Table 1.

Extensive MTL studies have been conducted on linear models using regularized approaches. The basic MTL algorithm that we consider in this paper is as follows:

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{i=1}^{m} \mathscr{L}_i(\mathbf{X}_i \mathbf{w}_i, \mathbf{y}_i) + \lambda g(\mathbf{W}), \tag{1}$$

where m is the number of tasks. The datasets for the tasks are denoted by $\mathscr{D}^m = (\mathbf{X}^m, \mathbf{y}^m) = \{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_m, \mathbf{y}_m)\}$ where for each $i \in [m], \mathscr{D}_i = (\mathbf{X}_i, \mathbf{y}_i)$, where $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ and $\mathbf{y}_i \in \mathbb{R}^{n_i \times 1}$ denote the data matrix and target vector of the i-th task with n_i samples and dimensionality d, respectively. $\mathscr{L}_i$ is the prediction loss function for the i-th task. In this paper, we focus on linear MTL models in which $\mathbf{w}_i$ denotes the model/predictor for task i and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ is the model parameter matrix. $g(\cdot)$ is a regularization term that represents the structure of the information shared among the tasks, for which λ is a pre-fixed hyper-parameter. As a special case, STL can be described by (1) with $\lambda = 0$.

The key to MTL is to relate the tasks via a shared representation, which, in turn, benefits the tasks to be learned. Each possible shared representation encodes certain assumptions regarding task relatedness.

A typical/mainstream assumption is that the tasks share a latent low-rank subspace [76, 63, 31, 4, 17, 71, 34, 78]. The formulation leads to a low-rank structure of the model matrix. Because optimization problems involving rank functions are intractable, a trace-norm penalty is typically used [3, 39, 59], as in the following problem, which will be referred to as the *trace-norm-regularized MTL problem*.

$$\min_{\mathbf{W}} \sum_{i=1}^{m} \mathscr{L}_i(\mathbf{X}_i \mathbf{w}_i, \mathbf{y}_i) + \lambda \|\mathbf{W}\|_* . \tag{2}$$

Another typical assumption is that all tasks share a subset of important features. Such task relatedness can be captured by imposing a group-sparse penalty on the predictor matrix to select shared features across tasks [66, 70, 49]. One commonly used group-sparse penalty is the group ℓ_1 penalty [49, 54], as in the following problem, which will be referred to as the *group-ℓ_1-regularized MTL problem*.

$$\min_{\mathbf{W}} \sum_{i=1}^{m} \mathscr{L}_i(\mathbf{X}_i \mathbf{w}_i, \mathbf{y}_i) + \lambda \|\mathbf{W}\|_{2,1} . \tag{3}$$

Next, we present a compact definition of the model-protection problem in the context of MTL and discuss the general approach without differential privacy. As (1) shows, as a result of the joint learning process, $\widehat{\mathbf{w}}_j$ may contain some information on $\widehat{\mathbf{w}}_j$, for $i, j \in [m]$ and $i \neq j$, making it possible for the owner of task j to use such information to attack task i. Thus, we define the model-protection problem:

Definition 1 (Model-protection Problem for MTL).

The model-protection problem for MTL has three objectives:

1. *to minimize the information on $\widehat{\mathbf{w}}_i$ that can be inferred from $\widehat{\mathbf{w}}_{[-i]}$, for all $i \in [m]$*
2. *to maximize the prediction performance of $\widehat{\mathbf{w}}_i$, for all $i \in [m]$; and*

3. *to share useful predictive information among tasks.*

Now, consider such a procedure in which a trusted curator collects independently-trained models, denoted by $\mathbf{w}_1, \ldots, \mathbf{w}_m$, for all tasks *without their associated data* to be used as input. After the joint learning process, the curator outputs the updated models, denoted by $\widehat{\mathbf{W}}$, and sends each updated model to each task privately. The model collection and joint learning processes are performed alternately.

We note that the *trace-norm-regularized MTL problem* and the *group-ℓ_1-regularized MTL problem* are unified in the multi-task feature learning framework, which is based on the covariance matrix of the tasks' predictors [6, 24, 7]. Many other MTL methods also fall under this framework, such as learning clustered structures among tasks [30, 79] and inferring task relations [77, 25, 11]. As such, we note that the tasks' covariance matrix constitutes a major source of shared knowledge in MTL methods; hence, it is regarded as the primary target for model protection.

Therefore, we address the model-protection problem by rephrasing the first objective in Definition 1 as follows: to minimize the changes in $\widehat{w}_{[-i]}$ and the tasks' covariance matrix ($\widehat{\mathbf{W}}\widehat{\mathbf{W}}^{\mathrm{T}}$ or $\widehat{\mathbf{W}}^{\mathrm{T}}\widehat{\mathbf{W}}$) when task i participates in the joint learning process for all $i \in [m]$. Thus, the model for this new task is protected. Then, we find that the concept of differential privacy (minimizing the change in the output distribution) can be adopted to further rephrase this objective as follows: to minimize the changes in *the distribution of* $\widehat{w}_{[-i]}$ and the tasks' covariance matrix when task i participates in the joint learning process for all $i \in [m]$.

In differential privacy, algorithms are randomized by introducing some type of perturbation.

Definition 2 (Randomized Algorithm).

A randomized algorithm $\mathscr{A}: \mathscr{D} \to \theta \in \mathscr{C}$ is built by introducing some type of perturbation into some mapping $\mathscr{D} \to \theta \in \mathscr{C}$. Algorithm $\mathscr{A}$ outputs $\mathscr{A}(\mathscr{D}) = \theta$ with a density $p(\mathscr{A}(\mathscr{D}) = \theta)$ for each $\theta \in \mathscr{C}$. The probability space is over the perturbation introduced into algorithm $\mathscr{A}$.

In this paper, $\mathscr{A}$ denotes a randomized machine learning estimator, and θ denotes the view of potential adversaries, which includes the model parameters that we wish to estimate. Perturbations can be introduced into the original learning system via the (1) input data [47, 8], (2) model parameters [14, 40], (3) objective function [15, 73], or (4) optimization process [62, 69].

The formal definition of differential privacy is as follows.

Definition 3 (Dwork et al. [22]).

A randomized algorithm $\mathscr{A}$ provides (ϵ, δ)-differential privacy if, for any two adjacent datasets $\mathscr{D}$ and $\mathscr{D}'$ that differ by a single entry and for any set $\mathscr{S}$,

$$\mathbb{P}(\mathscr{A}(\mathscr{D}) \in \mathscr{S}) \leq \exp(\epsilon)\mathbb{P}(\mathscr{A}(\mathscr{D}') \in \mathscr{S}) + \delta,$$

where $\mathscr{A}(\mathscr{D})$ and $\mathscr{A}(\mathscr{D}')$ are the outputs of $\mathscr{A}$ on the inputs $\mathscr{D}$ and $\mathscr{D}'$, respectively.

The privacy loss pair (ϵ, δ) is referred to as the privacy budget/loss, and it quantifies the privacy risk of algorithm $\mathscr{A}$. The intuition is that it is difficult for a potential attacker to infer whether a certain data point has been changed in (or added to) the dataset $\mathscr{D}$ based on a change in the output distribution. Consequently, the information of any single data point is protected.

Furthermore, note that differential privacy is defined in terms of application-specific adjacent input databases. In our setting, these are each task's model and dataset pair, which are treated as a "single entry" by Definition 3.

Several mechanisms exist for introducing a specific type of perturbation. A typical type is calibrated to the sensitivity of the original "unrandomized" machine learning estimator $f: \mathscr{D} \to \theta \in \mathbb{R}^d$. The sensitivity of an estimator is defined as the maximum change in its output due to a replacement of any single data instance.

Definition 4 (Dwork et al. [22]).

The sensitivity of a function $f: \mathscr{D} \to \mathbb{R}^d$ *is defined as*

$$S(f) = \max_{\mathscr{D}, \mathscr{D}'} \|f(\mathscr{D}) - f(\mathscr{D}')\|$$

for all datasets $\mathscr{D}$ *and* $\mathscr{D}'$ *that differ by at most one instance, where* $\|\cdot\|$ *is specified by a particular mechanism. For example, the Gaussian mechanism [23] requires the* ℓ_2 *norm, and the Laplace mechanism [22] requires the* ℓ_1 *norm.*

The use of additive noise such as Laplace [22] or Gaussian noise [23] with a standard deviation proportional to $S(f)$ is a common practice for guaranteeing private learning. In this paper, we adopt the Wishart noise for covariance matrices [41], which is defined as follows.

Definition 5 (Gupta and Nagar [32]).

A $d \times d$ *random symmetric positive definite matrix* $\mathbf{E}$ *is said to have a Wishart distribution* $\mathbf{E} \sim W_d(\nu, \mathbf{V})$ *if its probability density function is*

$$p(\mathbf{E}) = \frac{\left|\mathbf{E}\right|^{(\nu - d - 1)/2} \exp\left(-\mathrm{tr}\left(\mathbf{V}^{-1}\mathbf{E}\right)/2\right)}{2^{\frac{\nu d}{2}} |\mathbf{V}|^{1/2} \Gamma_d(\nu/2)},$$

where $\nu > d - 1$ *and* $\mathbf{V}$ *is a* $d \times d$ *positive definite matrix.*

Because machine learning schemes are usually presented as sequential paradigms with multiple iterations and usually output multiple variables simultaneously, several differential privacy properties are particularly useful for ensuring privacy in machine learning, such as post-processing immunity, group privacy, composition properties and adaptive composition. The details of these properties are introduced in the supplementary material (Appendix F).

4 Methodology

We present our methodology in this section: the modeling of and rationale for our MP-MTL framework, two instantiations and utility analyses. Regarding the theoretical results, we present only the main results; the detailed derivations are included in the provided supplementary material (Appendix H).

4.1 The General MP-MTL Framework

Consider an MTL algorithm $\mathscr{A}$ with T iterations. For $t = 1, \ldots, T$, a trusted curator collects the models of m tasks, respectively, denoted by $\mathbf{w}_1^{(t-1)}, \ldots, \mathbf{w}_m^{(t-1)}$. Then, a model-protection and shared-information-extraction procedure is performed, and the updated models $\mathbf{w}_1^{(t)}, \ldots, \mathbf{w}_m^{(t)}$ are output and sent back to their respective tasks.

Remark 1.—In each iteration, the curator collects only the models. The dataset for each task can be regarded as the input for the entire MTL algorithm, but it is not the input for the curator.

In such a setting, for each $i \in [m]$, we wish to protect the dataset $\mathscr{D}_i = (\mathbf{X}_i, \mathbf{y}_i)$ of task i and its *entire input model-sequence* $(\mathbf{w}_i^{(0)}, \ldots, \mathbf{w}_i^{(T-1)})$ (denoted by $\mathbf{w}_i^{(0:T-1)}$ for short). *For the i-th task*, the *entire output model-sequence* of other tasks, $\mathbf{w}_{[-i]}^{(1:T)}$, belongs to the *view* of a potential adversary (i.e., the information that the adversary can acquire to infer the unique information of task i). Note that although the output model-sequence of each task is what we ultimately wish to protect, the unique information within each task is contained in the task's dataset and input model-sequence, which are actually protected.

The idea for using differental privacy tools is as follows. For simpilicity, we assume that $T = 1$ and omit the iteration-step indices. Let $\widetilde{\mathscr{D}} = \{(\mathbf{w}_1^{(0)}, \mathscr{D}_1), \ldots, (\mathbf{w}_m^{(0)}, \mathscr{D}_m)\}$ be an augmented dataset; i.e., let $(\mathbf{w}_i^{(0)}, \mathscr{D}_i)$ be treated as the i-th "data instance" of the augmented dataset $\widetilde{\mathscr{D}}$, for all $i \in [m]$. Thus, the m datasets and m models associated with the m tasks are transformed into a single dataset $\widetilde{\mathscr{D}}$ with m "data instances". Then, we define m outputs $\theta = (\theta_1, \ldots, \theta_m)$ such that for each $i \in [m]$, $\theta_i \in \mathscr{C}_i$ denotes the *view* of an adversary for task i, which includes $w_{[-i]}^{(1)}$. Thus, an (ϵ, δ) - MP-MTL algorithm $\mathscr{A}(\mathscr{B})$ should satisfy the following m inequalities. For each $i \in [m]$, for all neighboring datasets $\widetilde{\mathscr{D}}$ and $\widetilde{\mathscr{D}}'$ that differ by the i-th "data instance", and for any set $\mathscr{S}_i \subseteq C_i$, we have

$$\mathbb{P}(\theta_i \in \mathscr{S}_i \mid \mathscr{B} = \widetilde{\mathscr{D}}) \leq e^{\epsilon}\mathbb{P}(\theta_i \in \mathscr{S}_i \mid \mathscr{B} = \widetilde{\mathscr{D}}') + \delta\,. \tag{4}$$

We formally define an MP-MTL algorithm as follows.

Definition 6 (MP-MTL).—*Let $\mathscr{A}$ be a randomized MTL algorithm with a number of iterations T. In the first iteration, $\mathscr{A}$ performs the mapping* $\left(\mathbf{W}^{(0)} \in \mathbb{R}^{d \times m}, \mathscr{D}^m\right) \to \theta_1 \in \mathscr{C}_1$,

where θ_1 *includes* $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times m}$. *For* $t = 2, \ldots, T$, *in the t-th iteration,* $\mathscr{A}$ *performs the mapping* $\left(\mathbf{W}^{(t-1)} \in \mathbb{R}^{d \times m}, \mathscr{D}^m, \theta_1, \ldots, \theta_{t-1}\right) \to \theta_t \in \mathscr{C}_t$, *where* θ_t *includes* $\mathbf{W}^{(t)} \in \mathbb{R}^{d \times m}$. *Then,* $\mathscr{A}$ *is an* (ϵ, δ) - MP-MTL algorithm *if for all* $i \in [m]$, *for all* $t \in [T]$, *and for neighboring input pairs* $\left(\mathbf{W}^{(t-1)}, \mathscr{D}^m\right)$ *and* $((\mathbf{W}')^{(t-1)}, (\mathscr{D}')^m)$ *that differ only by the i-th task such that* $\mathbf{w}_i^{(t-1)} \neq (\mathbf{w}_i')^{(t-1)}$ *or* $\mathscr{D}_i \neq \mathscr{D}_i'$, *the following holds for some constants* $\epsilon, \delta \geq 0$ *and for any set* $\mathscr{S} \subseteq \mathbb{R}^{d \times (m-1) \times T}$:

$$\mathbb{P}(\mathbf{w}_{[-i]}^{(1:T)} \in \mathscr{S} \mid \bigcap_{t=1}^{T} \mathscr{B}_t) \leq e^{\epsilon} \mathbb{P}(\mathbf{w}_{[-i]}^{(1:T)} \in \mathscr{S} \mid \bigcap_{t=1}^{T} \mathscr{B}_t') + \delta, \tag{5}$$

where for all $t \in [T]$, $\mathscr{B}_t$, $\mathscr{B}_t'$ *denote the inputs for the t-th iteration:*

$$\mathscr{B}_t = \left(\mathbf{W}^{(t-1)}, \mathscr{D}^m, \boldsymbol{\theta}_{1:t-1}\right), \mathscr{B}_t' = ((\mathbf{W}')^{(t-1)}, (\mathscr{D}')^m, \boldsymbol{\theta}_{1:t-1}),$$

and

$$\theta_{1:t-1} = \begin{cases} \varnothing, & t = 1 \\ \theta_1, \theta_2, \cdots, \theta_{t-1}, & t \geq 2. \end{cases}$$

Note that in Definition 6, we view the model sequence $\mathbf{w}_{[-i]}^{(1:T)}$ as a single output of the algorithm for each task $i \in [m]$. The definition of neighboring inputs allows the model and dataset for any task to change in *all rounds* of the iterative optimization rather than in only a single round.

Definition 6 is defined based on differential privacy. Specifically, an algorithm is always first proven to satisfy the constraints of differential privacy and then proven to satisfy the constraints of our proposed MP-MTL. Thus, some results (e.g., composition) from differential privacy are also applicable to prove whether an algorithm is an MP-MTL algorithm.

STL can easily be shown to be optimal for avoiding information leakage across tasks because the individual task models are learned independently.

Claim 1.—*Any STL algorithm that learns each task independently is a* (0, 0) - *MP-MTL algorithm.*

From this claim, we learn that when no information is shared across tasks, no leakage across tasks will occur.

Our MP-MTL framework is elaborated in Algorithm 1, which considers heterogeneous privacy budgets for different iteration steps. To maintain the total privacy budget below a specified value using the adaptive composition theorem provided by Kairouz et al. [42], we define a composition bound of a series of privacy budgets (the equation is taken directly from Theorem 3.5 of Kairouz et al. [42]):

Definition 7 (Composition Bound of Privacy Budgets).—*For an integer* $T \geq 1$, *a series of privacy budgets,* $\epsilon_1, \ldots, \epsilon_T \geq 0$, *and a specified privacy loss* $\delta \geq 0$, *the composition bound of* $\{\epsilon_t\}$ *is defined as* $CB(\{\epsilon_t\}, \delta)$, *which equals*

$$\min\left\{\sum_{t=1}^{T}\epsilon_t, \sum_{t=1}^{T}\frac{\left(e^{\epsilon_t}-1\right)\epsilon_t}{\left(e^{\epsilon_t}+1\right)}+\sqrt{\sum_{t=1}^{T}2\epsilon_t^2\log\left(\frac{1}{\delta}\right)}, \sum_{t=1}^{T}\frac{\left(e^{\epsilon_t}-1\right)\epsilon_t}{\left(e^{\epsilon_t}+1\right)}+\sqrt{\sum_{t=1}^{T}2\epsilon_t^2\log\left(e+\frac{\sqrt{\sum_{t=1}^{T}\epsilon_t^2}}{\delta}\right)}\right\}. \tag{6}$$

In Algorithm 1, as mentioned in Section 3, we choose to protect the tasks' covariance matrix, which is denoted by $\mathbf{\Sigma} = \mathbf{W}\mathbf{W}^{\mathrm{T}}$ or $\mathbf{\Sigma} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$, depending on the MTL method selected. As previously stated, Wishart noise [41] is added. Fig. 1 illustrates the key concepts of the framework. In detail, Step 1 of Algorithm 1 ensures that the total privacy budgets satisfy the specified values ϵ and δ, respectively. The purpose of norm clipping in Step 3 is simply to render the models in a bounded space, which helps us compute a proper noise scale to add to satisfy the privacy constraint defined in Definition 6. Step 4 extracts the shared information between tasks—the tasks' covariance matrix. Step 5 adds a perturbation into the shared information. Step 6 further extracts useful information from the perturbed covariance matrix. Step 7 sends the extracted useful information to each task to perform decoupled learning. If no noise is added, Steps 4–7 could be a proximal gradient descent step, i.e., first performing a proximal operator step and then taking a gradient descent step; see, e.g., Ji and Ye [39] and Liu et al. [49]. This framework is applicable for many optimization schemes, such as proximal gradient methods [39, 49], alternating methods [5] and Frank-Wolfe methods [37].

Note that we mainly provided theoretical and experimental results for the $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ type of covariance matrix. Nonetheless, the $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ type of covariance matrix can be regarded as a natural alternative to include in our framework, since it was successfully used to learn relationships between tasks [77, 33, 76].

Remark 2.—*In Algorithm 1, a curator who collects models and performs centralized transfer needs to run only Steps 4–6 and does not need to collect the datasets* $(\mathbf{X}^m, \mathbf{y}^m)$, *which are used only in STL algorithms.*

Algorithm 1

MP-MTL framework

Input: Datasets $\left(\mathbf{X}^m, \mathbf{y}^m\right) = \{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_m, \mathbf{y}_m)\}$, where $\forall i \in [m]$, $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ and $\mathbf{y}_i \in \mathbb{R}^{n_i \times 1}$. Privacy loss ϵ, $\delta \geq 0$. Number of iterations T. Initial shared information matrix $\mathbf{M}^{(0)}$. Initial task models $\mathbf{W}^{(0)}$, which can be acquired via arbitrary STL methods.

Output: $\mathbf{W}^{(1:T)}$.

1: Set $\{\epsilon_t\}$ such that $CB(\{\epsilon_t\}, \delta) \leq \epsilon$, where $CB(\{\epsilon_t\}, \delta)$ is the composition bound of $\{\epsilon_t\}$.

2: **for** $t = 1 : T$ **do**

3: Norm clipping: $\widetilde{\mathbf{w}}_i^{(t-1)} = \mathbf{w}_i^{(t-1)} / \max(1, \frac{\left\|\mathbf{w}_i^{(t-1)}\right\|_2}{K})$, for all $i \in [m]$. Let $\mathbf{W}^{(0)} = \widetilde{\mathbf{W}}^{(0)}$.

4: $\widetilde{\mathbf{\Sigma}}^{(t)} = \widetilde{\mathbf{W}}^{(t-1)} (\widetilde{\mathbf{W}}^{(t-1)})^{\mathrm{T}}$ (or $\widetilde{\mathbf{\Sigma}}^{(t)} = (\widetilde{\mathbf{W}}^{(t-1)})^{\mathrm{T}} \widetilde{\mathbf{W}}^{(t-1)}$).

5: $\Sigma^{(t)} = \widetilde{\Sigma}^{(t)} + \mathrm{E}$, where $\mathbf{E} \sim \mathcal{W}_d(d+1, \frac{K^2}{2\epsilon_t}\mathbf{I}_d)$ (or $\mathbf{E} \sim \mathcal{W}_m(m+1, \frac{K^2}{2\epsilon_t}\mathbf{I}_m)$) is a sample from the Wishart distribution, $\mathbf{I}_d$ denotes the $d \times d$ identity matrix, and diag$(\cdot)$ transforms a vector into a diagonal matrix.

6: Perform an arbitrary mapping $f: \Sigma^{(1:t)} \rightarrow \mathbf{M}^{(t)}$, e.g., take the diagonal elements of $\mathbf{\Sigma}^{(t)}$ or the singular value decomposition of $\mathbf{\Sigma}^{(t)}$.

7: $\mathbf{w}_i^{(t)} = \mathscr{A}_{\mathrm{st},i}(\mathbf{M}^{(t)}, \widetilde{\mathbf{w}}_i^{(0:t-1)}, \mathbf{X}_i, \mathbf{y}_i)$, for all $i \in [m]$, where $\mathscr{A}_{\mathrm{st},i}$ is an arbitrary STL algorithm for the i-th task and the $\widetilde{\mathbf{w}}_i^{(0:t-1)}$ are used for initialization.

8: **end for**

4.2 Instantiations of the MP-MTL Framework

In this section, we instantiate our MP-MTL framework (described in Algorithm 1) by approximating the *proximal gradient descent methods* presented by Ji and Ye [39] and Liu et al. [49] for the *trace-norm-regularized MTL problem* and the *group-ℓ_1-regularized MTL problem*, respectively. Both proximal gradient descent methods solve the respective MTL problems by alternately performing a proximal operator step and a gradient descent step. Taking the *trace-norm-regularized MTL problem* as an example, the loss function, $\sum_i \mathscr{L}_i$, is minimized by the gradient descent steps, while the regularization term, the trace-norm, is minimized by the proximal operator steps. The proximal operator minimizes the regularization term, keeping the variable near the result of a previous gradient descent step. Specifically, we instantiate Steps 4–7 of Algorithm 1 by approximating a proximal gradient descent step, i.e., first performing a proximal operator step and then taking a gradient descent step. It is similar for the *group-ℓ_1-regularized MTL problem*, but the difference lies in the instantiations of Step 6 of Algorithm 1 because different regularization terms lead to different optimal solutions for the proximal operators. Note that both instantiations use the $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ type of covariance matrix, which is required by the optimal solutions [39, 49].

First, we instantiate the MP-MTL framework for the *trace-norm-regularized MTL problem*, as shown in Algorithm 2. Generally speaking, the algorithm uses an accelerated proximal gradient method. Steps 4–9 approximate the following proximal operator [39]:

$$\widehat{\mathbf{W}}^{(t-1)} = \operatorname{argmin}_{\mathbf{W}} \frac{1}{2\eta} \|\mathbf{W} - \widetilde{\mathbf{W}}^{(t-1)}\|_F^2 + \lambda \|\mathbf{W}\|_*, \tag{7}$$

where $\widetilde{\mathbf{W}}^{(t-1)}$ can be regarded as the result of the gradient descent step in the previous iteration, assuming K is sufficiently large. In detail, Steps 6–8 of Algorithm 2 instantiate Step 6 of Algorithm 1 by constructing a projection matrix, $\mathbf{M}^{(t)} = \mathbf{U}\mathbf{S}_{\eta\lambda}\mathbf{U}^{\mathrm{T}}$, from the result of singular vector decomposition of the perturbed covariance matrix. Steps 9–11 of Algorithm 2 instantiate Step 7 of Algorithm 1 by first projecting the models (in Step 9) and then performing accelerated gradient descent.

We provide a running example for model leakage and model protection under different settings of Algorithm 2, as shown in Fig. 2. We generate models for $m = 10$ tasks, in which the data dimension is $d = 5$. The 10th task (the rightmost task), is an anomaly task that requires privacy protection. In Fig. 2 (a), the matrix denoted by $\mathbf{W}^{(0)}$ is first generated from an i.i.d. uniform distribution $\mathcal{U}(0, 1)$. Then, the rightmost column is multiplied by 100. For MTL with model leakage, we execute Algorithm 2, setting $T = 1$, $\eta = 1$, $\epsilon_1 = \epsilon = 1e40$, $\delta = 0$, $K = 100\sqrt{5}$ and $\lambda = 50$. Because ϵ is sufficiently large, the sampled noise matrix $\mathbf{E}$ added to the covariance matrix is approximately a zero matrix. Then, it can be regarded that the MTL runs using a normal proximal operator. The output model matrix $\widehat{\mathbf{W}}^{(1)}$ is shown in Fig. 2 (b), in which the 10th task results in significant influences on the parameters of other models: other models' parameters are similar to those of the 10th task, e.g., for each task, the first feature is the largest, and the fifth feature is the smallest. For MTL with model protection, we execute Algorithm 2 with the same setting as above except that we set $\epsilon_1 = \epsilon = 0.1$. The output model matrix $\widehat{\mathbf{W}}^{(1)}$ is shown in Fig. 2 (c), in which the influences from the 10th task are not significant: other models' parameters are not similar to those of the 10th task. Meanwhile, for $\mathbf{W}^{(0)}$, shown in Fig. 2 (a), for tasks 1–9, the ℓ_2 norms of the second and fifth rows are the two largest ones; these are clearly shown in Fig. 2 (c). This result means that the shared information between tasks is to use the second and fifth features, which is successfully extracted by our method.

Algorithm 2

MP-MTL Low-rank Estimator

Input: Datasets $(\mathbf{X}^m, \mathbf{y}^m) = \{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_m, \mathbf{y}_m)\}$, where $\forall i \in [m]$, $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ and $\mathbf{y}_i \in \mathbb{R}^{n_i \times 1}$. Privacy loss ϵ, $\delta \geq 0$. Number of iterations T. Step size η. Regularization parameter $\lambda > 0$. Norm clipping parameter $K > 0$. Acceleration parameters $\{\beta_t\}$. Initial task models $\mathbf{W}^{(0)}$.

Output: $\widehat{\mathbf{W}}^{(1:T)}$.

1: Set $\{\epsilon_t\}$ such that $CB(\{\epsilon_t\}, \delta) \leq \epsilon$, where $CB(\{\epsilon_t\}, \delta)$ is the composition bound of $\{\epsilon_t\}$.

2: **for** $t = 1 : T$ **do**

3: Norm clipping: $\widetilde{\mathbf{w}}_i^{(t-1)} = \mathbf{w}_i^{(t-1)} / \max(1, \frac{\left\|\mathbf{w}_i^{(t-1)}\right\|_2}{K})$, for all $i \in [m]$. Let $\widehat{\mathbf{W}}^{(0)} = \widetilde{\mathbf{W}}^{(0)}$.

4: $\widetilde{\mathbf{\Sigma}}^{(t)} = \widetilde{\mathbf{W}}^{(t-1)} (\widetilde{\mathbf{W}}^{(t-1)})^{\mathrm{T}}$.

5: $\mathbf{\Sigma}^{(t)} = \widetilde{\mathbf{\Sigma}}^{(t)} + \mathbf{E}$, where $\mathbf{E} \sim W_d(d+1, \frac{K^2}{2\epsilon_t}\mathbf{I}_d)$ is a sample from the Wishart distribution.

6: Perform singular vector decomposition: $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}} = \mathbf{\Sigma}^{(t)}$.

7: Let $\mathbf{S}_{\eta\lambda}$ be a diagonal matrix, and let $\mathbf{S}_{\eta\lambda, ii} = \max\{0, 1 - \eta\lambda/\sqrt{\Lambda_{ii}}\}$, for $i = 1, \ldots, \min\{d, m\}$.

8: $\mathbf{M}^{(t)} = \mathbf{U}\mathbf{S}_{\eta\lambda}\mathbf{U}^{\mathrm{T}}$.

9: Let $\widehat{\mathbf{w}}_i^{(t)} = \mathbf{M}^{(t)} \widetilde{\mathbf{w}}_i^{(t-1)}$, for all $i \in [m]$.

10: Let $\mathbf{z}_i^{(t)} = \widehat{\mathbf{w}}_i^{(t)} + \beta_t (\widehat{\mathbf{w}}_i^{(t)} - \widehat{\mathbf{w}}_i^{(t-1)})$, for all $i \in [m]$.

11: Let $\mathbf{w}_i^{(t)} = \mathbf{z}_i^{(t)} - \eta \frac{\partial \mathscr{L}_i(\mathbf{X}_i \mathbf{z}_i^{(t)}, \mathbf{y}_i)}{\partial \mathbf{z}_i^{(t)}}$, for all $i \in [m]$.

12: **end for**

Second, we instantiate the MP-MTL framework for the *group-ℓ_1-regularized MTL problem* defined in (3), as shown in Algorithm 3. Steps 4–8 approximate the following proximal operator [49]:

$$\widehat{\mathbf{W}}^{(t-1)} = \underset{\mathbf{W}}{\operatorname{argmin}} \frac{1}{2\eta} \left\| \mathbf{W} - \widetilde{\mathbf{W}}^{(t-1)} \right\|_F^2 + \lambda \|\mathbf{W}\|_{2,1}. \tag{8}$$

The only difference between Algorithm 3 and Algorithm 2 is the way they obtain the projection matrix $\mathbf{M}^{(t)}$ for the models (see the differences between Steps 6–8 of Algorithm 2 and Steps 6–7 of Algorithm 3). Because Algorithm 3 minimizes the group-sparse penalty, it focuses on only the diagonal elements of the perturbed covariance matrix.

The error bounds for the proximal operator approximations are provided in Section 4.4.

We use the following result to show that under high noise levels, our algorithms share no information between models but keep the models intact; thus, they degrade to STL methods but in such a way they do not underperform compared with STL methods.

Algorithm 3

MP-MTL Group-sparse Estimator

Input: Datasets $(\mathbf{X}^m, \mathbf{y}^m) = \{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_m, \mathbf{y}_m)\}$, where $\forall i \in [m]$, $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ and $\mathbf{y}_i \in \mathbb{R}^{n_i \times 1}$. Privacy loss ϵ, $\delta \geq 0$. Number of iterations T. Step size η. Regularization parameter $\lambda > 0$. Norm clipping parameter $K > 0$. Acceleration parameters $\{\beta_t\}$. Initial task models $\mathbf{W}^{(0)}$.

Output: $\widehat{\mathbf{W}}^{(1:T)}$.

1: Set $\{\epsilon_t\}$ such that $CB(\{\epsilon_t\}, \delta) \leq \epsilon$, where $CB(\{\epsilon_t\}, \delta)$ is the composition bound of $\{\epsilon_t\}$.

2: **for** $t = 1 : T$ **do**

3: Norm clipping: $\widetilde{\mathbf{w}}_i^{(t-1)} \equiv \mathbf{w}_i^{(t-1)} / \max\left(1, \frac{\left\|\mathbf{w}_i^{(t-1)}\right\|_2}{K}\right)$, for all $i \in [m]$. Let $\widehat{\mathbf{W}}^{(0)} = \widetilde{\mathbf{W}}^{(0)}$.

4: $\widetilde{\mathbf{\Sigma}}^{(t)} = \widetilde{\mathbf{W}}^{(t-1)} (\widetilde{\mathbf{W}}^{(t-1)})^{\mathrm{T}}$.

5: $\Sigma^{(t)} = \widetilde{\Sigma}^{(t)} + \mathbf{E}$, where $\mathbf{E} \sim W_d(d+1, \frac{K^2}{2\epsilon_t} \mathbf{I}_d)$ is a sample of the Wishart distribution.

6: Let $\mathbf{S}_{\eta\lambda}$ be a diagonal matrix, where for $i = 1, \ldots, d$, $\mathbf{S}_{\eta\lambda, ii} = \max\{0, 1 - \eta\lambda / \sqrt{\left|\Sigma_{ii}^{(t)}\right|}\}$.

7: $\mathrm{M}^{(t)} = \mathbf{S}_{\eta\lambda}$.

8: Let $\widehat{\mathbf{w}}_i^{(t)} = \mathbf{M}^{(t)} \widetilde{\mathbf{w}}_i^{(t-1)}$, for all $i \in [m]$.

9: Let $\mathbf{z}_i^{(t)} = \widehat{\mathbf{w}}_i^{(t)} + \beta_t (\widehat{\mathbf{w}}_i^{(t)} - \widehat{\mathbf{w}}_i^{(t-1)})$, for all $i \in [m]$.

10: Let $\mathbf{w}_i^{(t)} = \mathbf{z}_i^{(t)} - \eta \frac{\partial \mathscr{L}_i(\mathbf{X}_i \mathbf{z}_i^{(t)}, \mathbf{y}_i)}{\partial \mathbf{z}_i^{(t)}}$, for all $i \in [m]$.

11: **end for**

Proposition 1.—*For Algorithm 2, the projection matrix* $\mathbf{U}\mathbf{S}_{\eta\lambda}\mathbf{U}^{\mathrm{T}}$ *degrades to an identity matrix, and the algorithm degrades to an STL algorithm with no random perturbation if the smallest singular value of* $\mathbf{E}$ *satisfies* $\sigma_d(\mathbf{E}) = C\lambda^2$ *for a sufficiently large* $C > 0$.

For Algorithm 3, the projection matrix $\mathbf{S}_{\eta\lambda}$ *degrades to an identity matrix, and the algorithm degrades to an STL algorithm with no random perturbation if the smallest diagonal element of* $\mathbf{E}$ *satisfies* $\min_j \mathbf{E}_{jj} = C\lambda^2$ *for sufficiently large* $C > 0$.

We also consider other complex MTL frameworks for instantiation. For example, Gong et al. [29], Chen et al. [18], Jalali et al. [38] and Chen et al. [19] considered a decomposed parameter/model matrix to handle heterogeneities among tasks, e.g., detecting entry-wise outliers in the parameter matrix [38, 19] and detecting anomalous tasks [29, 18]. These detection procedures are claimed to be beneficial for the knowledge sharing process in cases of heterogeneous tasks. Our MP-MTL framework can be naturally extended to such a model-decomposed setting because the additional procedures are still STL algorithms; hence, the privacy loss will not increase. Please see the supplementary material (Appendix A) for additional details.

4.3 Privacy Guarantees

The following two results show that our proposed framework and the two instantiated algorithms satisfy the privacy constraint defined in Definition 6.

Theorem 1.—*Algorithm 1 is an* (ϵ, δ) *- MP-MTL algorithm.*

The proof of Theorem 1 can be found in the supplementary material (Appendix H.3).

Corollary 1.—*Algorithm 2 and Algorithm 3 are* (ϵ, δ) *- MP-MTL algorithms.*

Corollary 1 follows Theorem 1 because Algorithm 2 and 3 are instantiations of Algorithm 1. The necessary property to be satisfied is that the instantiation of Step 7 of Algorithm 1 is an STL algorithm. The proof of Corollary 1 can be found in the supplementary material (Appendix H.4).

4.4 Utility Analyses

We build utility analyses specifically for our instantiations, i.e., Algorithms 2 and 3 instead of Algorithm 1, because 1) Algorithm 1 is a framework that allows the minimization of a variety of regularization terms and many optimization schemes. Specifically, Steps 6 and 7 of Algorithm 1 include arbitrary mappings and arbitrary STL algorithms, respectively. Therefore, the analysis is not trivial and requires additional assumptions. 2) Algorithms

2 and 3 correspond to trace-norm and group-ℓ_1-norm regularization, respectively, which correspond to two mainstream MTL approaches.

Our utility analyses are built upon the matrix perturbation error bounds of Wishart noise presented by Jiang et al. [41], the error bounds with arbitrary heterogeneous residues of inexact proximal-gradient descent presented by Schmidt et al. [60], and the two optimal solutions for proximal operators presented by Ji and Ye [39] and Liu et al. [49]. The following parts of the utility analyses are novel: 1) the derivations of the approximation error bounds for both the proximal operators in (7) and (8); 2) the derivations of runtime and utility bounds, considering three cases of composition bounds of privacy budgets, two privacy-budget allocation strategies, two specific regularization terms, both convex and strongly convex prediction loss functions, and both the basic and accelerated proximal-gradient methods, subject to the elaborate composition theorem of privacy; 3) the optimizations of the utility bounds with respect to the parameters of privacy-budget allocation strategies.

We studied the utility bounds for three cases of the composition bound of $\{\epsilon_t\}$ defined in (6). It can be perceived that three composition theorems bound the total privacy loss given $\{\epsilon_t\}$. For different $\{\epsilon_t\}$, the best bound may be different. Specifically, the results corresponding to the bound for the low-privacy budget regime (e.g., $\epsilon + e\delta \leq 1$) are presented in the main paper, while the results corresponding to the other two bounds are similar and reported in the supplementary material (Appendix D) because the low-privacy budget regime may receive the most attention. Under such a regime, the bound is as follows.

$$\sum_{t=1}^{T} \frac{\left(e^{\epsilon_t}-1\right)\epsilon_t}{\left(e^{\epsilon_t}+1\right)} + \sqrt{\sum_{t=1}^{T} 2\epsilon_t^2 \log\left(e + \frac{\sqrt{\sum_{t=1}^{T}\epsilon_t^2}}{\delta}\right)}.$$

First, we make some assumptions.

Parameter space.—A bounded parameter space is assumed for model matrices:

$$\mathscr{W} = \{\mathbf{W} \in \mathbb{R}^{d \times m} : \max_{i \in [m]} \|\mathbf{w}_i\|_2 \leq K\},$$

where K is the norm clipping parameter.

Properties of objective functions.—We consider the loss function $f(\mathbf{W}) = \frac{1}{m}\sum_{i=1}^{m} \mathscr{L}_i(\mathbf{X}_i\mathbf{w}_i, \mathbf{y}_i)$ and assume that $mf(\mathbf{W})$ is convex and has an L-Lipschitz-continuous gradient (as defined in Schmidt et al. [60]). Let $W_* = \text{argmin}_{\mathbf{W}} mf(\mathbf{W}) + \lambda g(\mathbf{W})$, where $g(\cdot) = \|\cdot\|_*$ for Algorithm 2 and $g(\cdot) = \|\cdot\|_{2,1}$ for Algorithm 3. Without loss of generality, we assume that $\mathbf{W}_* \in \mathscr{W}$ and $f(\widetilde{\mathbf{W}}^{(0)}) - f(\mathbf{W}_*) = O\left(K^2 Lm\right)$. We define $q = \min\{d, m\}$.

The number of tasks.—The number of tasks is assumed to be sufficient as follows.

Assumption 1.: *For Algorithm 2, we assume that for sufficiently large* $C > 0$,

$$m > CK^2 d^2 \log^2(d)(\log(e + \epsilon/\sqrt{2}\delta) + 2\epsilon)/\epsilon^2 .$$

For Algorithm 3, we assume that for sufficiently large $C > 0$,

$$m > C\log(d)\sqrt{\log(e + \epsilon/\sqrt{2}\delta) + 2\epsilon}\big/\epsilon .$$

Then, we present the results. Before reporting the utility bounds, we report two intermediate results: the approximation error bounds for proximal operators with trace-norm (low-rank) and group-ℓ_1 (group-sparse) penalties, respectively. Now the noise matrix $\mathbf{E}$ is allowed to be arbitrary.

Lemma 1 (Low rank).: *Consider Algorithm 2. For* $t \in [T]$, *in the t-th iteration, let* $\mathbf{C} = \widetilde{\mathbf{W}}^{(t-1)}$. *Let* $r_c = rank(\mathbf{C}) \le q$ *be the rank of* $\mathbf{C}$. *Suppose that an index* $k \le q$ *exists such that* $\sigma_k(\mathbf{C}) > \eta\lambda$ and $\sigma_{k+1}(\mathbf{C}) \le \eta\lambda$. *Assume that* $2\sigma_1(E) \le \sqrt{\sigma_j(\mathbf{C})} - \sqrt{\sigma_{j+1}(\mathbf{C})}$ *for* $j \in [k]$. *Then for any* $\mathrm{E} \in \mathbb{R}^{d\times d}$:

$$\begin{aligned} &\frac{1}{2\eta}\|\widehat{\mathbf{W}}^{(t)} - \mathbf{C}\|_F^2 + \lambda\|\widehat{\mathbf{W}}^{(t)}\|_* \\ &- \left\{\min_{\mathbf{W}} \frac{1}{2\eta}\|\mathbf{W} - \mathbf{C}\|_F^2 + \lambda\|\mathbf{W}\|_*\right\} \\ &\le \frac{1}{\eta}\left(\frac{\sigma_1^2(\mathbf{C})}{\eta\lambda} + \sigma_1(\mathbf{C})\right) \\ &\cdot \left[\left(\frac{k^2}{\eta\lambda} + 2k\right)\sigma_1(\mathbf{E}) + \max(0, r_c - k)\sqrt{\sigma_1(\mathbf{E})}\right]. \end{aligned} \tag{9}$$

Lemma 2 (Group Sparse).: *Consider Algorithm 3. For* $t \in [T]$, *in the t-th iteration, let* $\mathbf{C} = \widetilde{\mathbf{W}}^{(t-1)}$. *Let the indices of the non-zero rows of* $\mathbf{C}$ *be denoted by* $\mathcal{I}_c = \left\{j : \mathbf{C}^j \ne \mathbf{0}\right\}$, *and let* $r_{c,s} = |\mathcal{I}_c| \le d$. *Let* $\mathbf{\Sigma}_0 = \mathbf{C}\mathbf{C}^{\mathrm{T}}$. *Suppose that an integer* $k \le d$ *exists such that* $\sum_{j=1}^{d} I\left(\sqrt{\Sigma_{jj,0}} \ge \eta\lambda\right) = k$, *where* $I(\cdot)$ *is the indicator function. Then for any* $\mathbf{E} \in \mathbb{R}^{d\times d}$, *we have:*

$$\begin{aligned} &\frac{1}{2\eta}\|\widehat{\mathbf{W}}^{(t)} - \mathbf{C}\|_F^2 + \lambda\|\widehat{\mathbf{W}}^{(t)}\|_{2,1} \\ &- \left\{\min_{\mathbf{W}} \frac{1}{2\eta}\|\mathbf{W} - \mathbf{C}\|_F^2 + \lambda\|\mathbf{W}\|_{2,1}\right\} \\ &\le \frac{1}{\eta}\left[\frac{r_{c,s}}{\eta\lambda}\left(\max_{j\in[d]}\|\mathbf{C}^j\|_2\right)^2 + \left(\max_{j\in[d]}\|\mathbf{C}^j\|_2\right)\right] \\ &\cdot \left[\frac{k}{2\eta\lambda}\max_{j\in[d]}|\mathbf{E}_{jj}| + \max(0, r_{c,s} - k)\max_{j\in[d]}\sqrt{|\mathbf{E}_{jj}|}\right]. \end{aligned} \tag{10}$$

We find that the approximation error bounds both depend on $\sigma_1(\mathbf{E})$ (note that $\max_j |\mathbf{E}_{jj}| \le \sigma_1(\mathbf{E})$).

Note that Lemma 1 requires $\eta\lambda$ to fall between the k-th and the $(k+1)$-th singular values in every iteration for the same k. Under Assumption 1 (the number of tasks is sufficiently large), when the initial task models in $W^{(0)}$ are acquired via proper STL methods, a significant margin will always exist between the k-th and the $(k+1)$-th singular values of the normalized model matrix $\mathbf{C}$ for the same k. Therefore, the above requirement is easily satisfied. It is similar for the requirement of k in Lemma 2.

Now, we present guarantees regarding both utility and runtime. In the following, $\mathbf{E}$ is assumed to be a Wishart random matrix. The privacy budgets $\{\epsilon_t\}$ are considered heterogeneous, i.e., different with respect to t.

We consider two cases for the loss function $f(\mathbf{W})$: convex and strongly convex. For each case, we report the results of both Algorithms 2 (the low-rank estimator) and 3 (the group-sparse estimator). For each algorithm, we present the results for both the basic and accelerated proximal gradient descent methods.

For the convex case of the loss function $f(\mathbf{W})$, according to Propositions 1 and 2 of Schmidt et al. [60], the weight for the approximation error of each iteration grows polynomially with respect to t. To minimize the bound for the total error, the approximation error should decrease polynomially. Considering that the approximation error bounds depend on $\sigma_1(\mathbf{E})$ and that $\sigma_1(\mathbf{E}) = O(1/\epsilon_t)$ (shown by Jiang et al. [41]), we should let ϵ_t grow polynomially. Therefore, we set $\epsilon_t = \Theta(t^\alpha)$ for $\alpha \in \mathbb{R}$ and $t \in [T]$, and define

$$M_0 = \sqrt{\log(e + \epsilon/\sqrt{2}\delta) + 2\epsilon}/\sqrt{|2\alpha + 1|}\epsilon,$$

which is used for both Theorems 2 and 3.

Theorem 2 (Low rank - Convexity).: *Consider Algorithm 2. For an index $k \leq q$ that satisfies the conditions given in Lemma 1 for all $t \in [T]$, $\eta = 1/L$, and $\lambda = \Theta(LK\sqrt{m})$, assume that $\epsilon_t \leq 4Kk^2d(\log d)/q^2$ for $t \in [T]$. Define*

$$M = M_0 Kkd\log d/\sqrt{m}.$$

No acceleration: *If we set $\beta_t = 0$ for $t \in [m]$ and then also set $T = \Theta(((\alpha/2 - 2)^2)^{\phi(\alpha)/2})$ for $\mathscr{E} = f(\widehat{\mathbf{W}}^{(T)}) - f(\mathbf{W}_*)$, we have, with high probability,*

$$\mathscr{E} = O(K^2L(M/(\alpha/2 - 1)^2)^{\phi(\alpha)}), \tag{11}$$

where

$$\phi(\alpha) = \begin{cases} 2/(2\alpha + 1), & \alpha > 2; \\ 2/5, & -1/2 < \alpha < 2; \\ 1/(2 - \alpha), & \alpha < -1/2; \end{cases} \tag{12}$$

Use acceleration: *If we set* $\beta_t = (t-1)/(t+2)$ *for* $t \in [m]$ *and then also set* $T = \Theta(((\alpha/2-2)^2)^{\phi(\alpha)/2})$ *for* $\mathscr{E} = f\left(\widehat{\mathbf{W}}^{(T)}\right) - f(\mathbf{W}_*)$, *we have, with high probability,*

$$\mathscr{E} = O(K^2 L(M/(\alpha/2-2)^2)^{\phi(\alpha)}), \tag{13}$$

where

$$\phi(\alpha) = \begin{cases} 4/(2\alpha+1), & \alpha > 4; \\ 4/9, & -1/2 < \alpha < 4; \\ 2/(4-\alpha), & \alpha < -1/2; \end{cases} \tag{14}$$

Theorem 3 (Group sparse - Convexity).: *Consider Algorithm 3. For an index set* $k \leq d$ *that satisfies the condition given in Lemma 2 for all* $t \in [T]$, $\eta = 1/L$, *and* $\lambda = \Theta(LKd\sqrt{m})$, *assume that* $\epsilon_t \leq k^2 \log(d)/4Kd(d-k)^2 m$ *for* $t \in [T]$. *Define*

$$M = M_0 k \log d/m .$$

No acceleration: *If we set* $\beta_t = 0$ *for* $t \in [m]$ *and then also set* $T = \Theta(((\alpha/2-2)^2)^{\phi(\alpha)})$ *for* $\mathscr{E} = f(\frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{W}}^{(t)}) - f(\mathbf{W}_*)$, *we have, with high probability,*

$$\mathscr{E} = O(K^2 L(M/(\alpha/2-1)^2)^{\phi(\alpha)}), \tag{15}$$

where $\phi(\alpha)$ *is defined in* (12).

Use acceleration: *If we set* $\beta_t = (t-1)/(t+2)$ *for* $t \in [m]$ *and then also set* $T = \Theta(((\alpha/2-2)^2)^{\phi(\alpha)/2})$ *for* $\mathscr{E} = f(\widehat{\mathbf{W}}^{(T)}) - f(\mathbf{W}_*)$, *we have, with high probability,*

$$\mathscr{E} = O(K^2 L(M/(\alpha/2-2)^2)^{\phi(\alpha)}), \tag{16}$$

where $\phi((\alpha)$ *is defined in* (14).

It can be observed that the group sparsity leads to better/smaller bounds compared with the low-rank case. This is because the group-sparse estimator only uses the diagonal elements of the perturbed covariance matrix and therefore introduces only a small part of the noise from the noisy matrix **E**. In contrast, the low-rank estimator exploits the whole perturbed covariance matrix, hence introducing all the noise.

Next, we assume that $mf(\mathbf{W})$ is μ-strongly convex and has an L-Lipschitz-continuous gradient, where $\mu < L$. In this case, according to Propositions 3 and 4 of Schmidt et al. [60], the weight for the approximation error of each iteration grows *exponentially* with respect to t. Therefore, we set $\epsilon_t = \Theta(Q^{-t})$ for $Q > 0$ and $t \in [T]$. Define

$$M_0' = \sqrt{\log(e + \epsilon/\sqrt{2}\delta) + 2\epsilon}/\sqrt{\left|1 - Q^2\right|}\epsilon,$$

which is used for both Theorems 4 and 5.

Theorem 4 (Low rank - Strong convexity).: *Consider Algorithm 2. For an index $k \leq q$ that satisfies the conditions given in Lemma 1 for all $t \in [T]$, $\eta = 1/L$, and $\lambda = \Theta(LK\sqrt{m})$, assume that $\epsilon_t \leq 4Kk^2d(\log d)/q^2$ for $t \in [T]$, denoted by*

$$M = M_0' Kkd \log d/\sqrt{m}.$$

No acceleration: *If we set $\beta_t = 0$ for $t \in [m]$ and then let $Q_0 = 1 - \mu/L$ and set $T = \Theta(\log_{1/\psi(Q,Q_0^2)}((Q_0/\sqrt{Q}-1)^2/M))$ for $\mathscr{E} = \frac{1}{\sqrt{m}}\|\widehat{\mathbf{W}}^{(T)} - \mathbf{W}_*\|_F$, we have, with high probability,*

$$\mathscr{E} = O(K(M/(Q_0/\sqrt{Q}-1)^2)^{\log_{\psi(Q,Q_0^2)}Q_0}), \tag{17}$$

where for any $\widetilde{Q} \in (0,1)$,

$$\psi(Q,\widetilde{Q}) = \begin{cases} Q, & 0 < Q < \widetilde{Q}; \\ \widetilde{Q}, & \widetilde{Q} < Q < 1; \\ \widetilde{Q}/Q, & Q > 1. \end{cases} \tag{18}$$

Use acceleration: *If we set $\beta_t = (1-\sqrt{\mu/L})/(1+\sqrt{\mu/L})$ for $t \in [m]$ and then let $Q_0' = 1 - \sqrt{\mu/L}$ and set $T = \Theta(\log_{1/\psi(Q,Q_0')}((\sqrt{Q_0'}/\sqrt{Q}-1)^2/M))$ for $\mathscr{E} = f(\widehat{\mathbf{W}}^{(T)}) - f(\mathbf{W}_*)$, we have, with high probability,*

$$\mathscr{E} = O(K(M/(\sqrt{Q_0'}/\sqrt{Q}-1)^2)^{\log_{\psi(Q,Q_0')}Q_0'}), \tag{19}$$

where $\psi(\cdot,\cdot)$ is defined in (18).

Theorem 5 (Group sparse - Strong convexity).: *Consider Algorithm 3. For an index $k \leq d$ that satisfies the condition given in Lemma 2 for all $t \in [T]$, $\eta = 1/L$, and $\lambda = \Theta(LKd\sqrt{m})$, assume that $\epsilon_t \leq k^2 \log(d)/4Kd(d-k)^2m$ for $t \in [T]$. Define*

$$M = M_0' k \log d/m.$$

No acceleration: *If we set $\beta_t = 0$ for $t \in [m]$ and then let $Q_0 = 1 - \mu/L$ and set $T = \Theta(\log_{1/\psi(Q,Q_0^2)}((Q_0/\sqrt{Q}-1)^2/M))$ for $\mathscr{E} = \frac{1}{\sqrt{m}}\|\widehat{\mathbf{W}}^{(T)} - \mathbf{W}_*\|_F$, we have, with high probability,*

$$\mathscr{E} = O(K(M/(Q_0/\sqrt{Q}-1)^2)^{\log_{\psi(Q,Q_0^2)}Q_0}), \tag{20}$$

where $\psi(\cdot,\cdot)$ *is defined in* (18).

Use acceleration: *If we set* $\beta_t = (1-\sqrt{\mu/L})/(1+\sqrt{\mu/L})$ *for* $t \in [m]$ *and then let* $Q_0' = 1-\sqrt{\mu/L}$ *and set* $T = \Theta(\log_{1/\psi(Q,Q_0')}((\sqrt{Q_0'}/\sqrt{Q}-1)^2/M))$ *for* $\mathscr{E} = f(\widehat{\mathbf{W}}^{(T)}) - f(\mathbf{W}_*)$, *we have, with high probability,*

$$\mathscr{E} = O(K(M/(\sqrt{Q_0'}/\sqrt{Q}-1)^2)^{\log_{\psi(Q,Q_0')}Q_0'}), \tag{21}$$

where $\psi(\cdot,\cdot)$ *is defined in* (18).

4.5 Privacy Budget Allocation

In this section, we optimize the utility bounds presented in Theorems 2–5 with respect to α and Q, respectively, which results in optimized privacy-budget allocation strategies. Then, we discuss the optimized results.

Theorem 6.—Consider Algorithms 2 and 3.

For a convex *f, we use Theorems 2 and 3.*

1. No acceleration*: The bounds for the low-rank and group-sparse estimators both reach their respective minima w.r.t.* α *at* $\alpha = 0$. *Meanwhile,* $\phi(\alpha) = 2/5$.
2. Use acceleration*: The bounds for low-rank and group-sparse estimators both reach their respective minima w.r.t.* α *at* $\alpha = 2/5$. *Meanwhile,* $\phi(\alpha) = 4/9$.

For a strongly convex *f, we use Theorems 4 and 5.*

1. No acceleration*: The bounds for the low-rank and group-sparse estimators both reach their respective minima w.r.t.* Q *at* $Q = Q_0^{2/5}$. *Meanwhile,* $\log_{\psi(Q,Q_0^2)}Q_0 = 1/2$.
2. Use acceleration*: The bounds for low-rank and group-sparse estimators both reach their respective minima w.r.t.* Q *at* $Q = (Q_0')^{1/5}$. *Meanwhile,* $\log_{\psi(Q,Q_0')}Q_0' = 1$.

the results correspondinging to the optimized privacy-budget allocation strategies (with $\delta > 0$) are summarized in Table 2, where the terms with respect to K, L, k, and $\sqrt{\log(e+\epsilon/\sqrt{2\delta})+2\epsilon}$ are omitted, and the results associated with the setting $\epsilon = \sum_{t=1}^{T}\epsilon_t$ are included, providing $(\epsilon, 0)$ - MP-MTL algorithms.

We learn from Theorem 6 that 1) for all four settings, a non-decreasing series of $\{\epsilon_t\}$ results in a good utility bound, since the best $\alpha = 0, 2/5 \geq 0$ for $\epsilon_t = \Theta(t^{\alpha})$ and the best $Q = Q_0^{2/5}$, $(Q_0')^{1/5} < 1$ for $\epsilon_t = \Theta(Q^{-t})$. Intuitively, this means adding non-increasing noise over the iterations—which is reasonable because the initial iterations may move quickly in the parameter space while the last iterations may only fine-tune the model slightly. 2) Both the strong-convexity condition and the acceleration strategy improve the utility bounds: both

increase the powers of those bounds that are far less than 1 under Assumption 1. 3) By setting α and Q to their optimized values, the acceleration strategy improves the runtime, as shown in Claim 2.

Claim 2.—*Assume Assumption 1. Consider Theorems 2–5 and set α and Q to the optimized values in Theorem 6, respectively. Assume $\mu/L < 0.3819$. The values for T are smaller when using the acceleration strategy compared to those with no acceleration.*

Now, we introduce a concrete strategy to set $\{\epsilon_t\}$ in both Algorithm 2 and Algorithm 3. We assume that T, ϵ and δ are given. Note that this strategy is optimal if α and Q are set according to the optimal settings stated by Theorem 6.

For a *convex f*, if no acceleration is to be used, then set $\beta_t = 0$ for $t \in [m]$ and set $\alpha \in \mathbb{R}$ (e.g., $\alpha = 0$); otherwise, set $\beta_t = (t-1)/(t+2)$ for $t \in [m]$ and set $\alpha \in \mathbb{R}$ (e.g., $\alpha = 2/5$). Then, for $t \in [T]$, let $\epsilon_t = \epsilon_0 t^{\alpha}$ and find the largest ϵ_0 that satisfies $CB(\{\epsilon_t\}, \delta) \leq \epsilon$, where $CB(\{\epsilon_t\}, \delta)$ is the composition bound of $\{\epsilon_t\}$ defined in (6).

For a *μ-strongly convex mf*$(\mathbf{W})$ with a known value of μ (e.g., $\frac{\mu}{2}\|\mathbf{w}_i\|_2^2$ is added to each $\mathscr{L}_i$), if no acceleration is to be used, then set $\beta_t = 0$ for $t \in [m]$ and set $Q > 0$ (e.g., $Q = (1 - \mu/L)^{2/5}$, if L is known); otherwise, if L is known, set $\beta_t = (1 - \sqrt{\mu/L})/(1 + \sqrt{\mu/L})$ for $t \in [m]$ and set $Q > 0$ (e.g., $Q = (1 - \sqrt{\mu/L})^{1/5}$). Then, for $t \in [T]$, let $\epsilon_t = \epsilon_0 Q^{-t}$ and find the largest ϵ_0 that satisfies $CB(\{\epsilon_t\}, \delta) \leq \epsilon$, where $CB(\{\epsilon_t\}, \delta)$ is the composition bound of $\{\epsilon_t\}$ defined in (6).

4.6 Baseline MP-MTL Constructed by IP-MTL

IP-MTL algorithms *prevent a single data instance* in one task from leaking to other tasks and are formally defined:

Definition 8 (IP-MTL).—*Let $\mathscr{A}$ be a randomized MTL algorithm with a number of iterations T. In the first iteration, $\mathscr{A}$ performs the mapping $(\mathbf{W}^{(0)} \in \mathbb{R}^{d \times m}, \mathscr{D}^m) \to \theta_1 \in \mathscr{C}_1$, where θ_1 includes $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times m}$. For $t = 2, \ldots, T$, in the t-th iteration, $\mathscr{A}$ performs the mapping $(\mathbf{W}^{(t-1)} \in \mathbb{R}^{d \times m}, \mathscr{D}^m, \theta_1, \ldots, \theta_{t-1}) \to \theta_t \in \mathscr{C}_t$, where θ_t includes $\mathbf{W}^{(t)} \in \mathbb{R}^{d \times m}$. Here, $\mathscr{A}$ is an (ϵ, δ)* - IP-MTL *algorithm if—for all $i \in [m]$ and for all neighboring datasets $\mathscr{D}^m$ and $(\mathscr{D}')^m$ that differ by a single data instance for the i-th task—the following holds for some constants ϵ, $\delta \geq 0$ and for any set $\mathscr{S} \subseteq \mathbb{R}^{d \times (m-1) \times T}$:*

$$\mathbb{P}(\mathbf{w}_{[-i]}^{(1:T)} \in \mathscr{S} \mid \bigcap_{t=1}^{T} \mathscr{B}_t) \leq e^{\epsilon}\mathbb{P}(\mathbf{w}_{[-i]}^{(1:T)} \in \mathscr{S} \mid \bigcap_{t=1}^{T} \mathscr{B}'_t) + \delta, \tag{22}$$

where for all $t \in [T]$, $\mathscr{B}_t$, $\mathscr{B}'_t$ denote the inputs for the t-th iteration:

$$\mathscr{B}_t = (\mathbf{W}^{(t-1)}, \mathscr{D}^m, \theta_{1:t-1}), \mathscr{B}'_t = ((\mathbf{W}')^{(t-1)}, (\mathscr{D}')^m, \theta_{1:t-1}),$$

and

$$\theta_{1:t-1} = \begin{cases} \varnothing, & t = 1 \\ \theta_1, \theta_2, \cdots, \theta_{t-1}, & t \geq 2, \end{cases}$$

and $(\mathbf{W}')^{(t-1)}$ *is associated with the case where a single data instance for the i-th task has been replaced.*

IP-MTL is similar to the one-query-to-many-analyst privacy defined by Hsu et al. [36]. However, these two definitions are different in both the modeling objects and the input characteristics.

We present examples of IP-MTL as follows.

Proposition 2.—*Both DP-AGGR [58] and DP-MTRL [33] are IP-MTL algorithms with* $T = 1$ *and* $T \geq 1$, *respectively.*

Now, we can construct baseline MP-MTL methods by IP-MTL methods based on the result of Proposition 3: to guarantee an (ϵ, δ) - MP-MTL algorithm, one can use an $(\epsilon/n, \delta/(n\exp(\epsilon))$ - IP-MTL algorithm.

Proposition 3.—*For task sample sizes of* $n_1, \ldots, n_m$, *any* (ϵ, δ) - *IP-MTL algorithm is a* $(n\epsilon, n\exp(n\epsilon)\delta)$ - *MP-MTL algorithm when* $n = \max_{i\in[m]} n_i$.

The proof of Proposition 3 can be found in the supplementary material (Appendix H.14), directly following the proof of the group privacy Lemma stated by Lemma 2.2 of Vadhan [67]. Therefore, Proposition 3 is regarded as the group privacy property of differential privacy applied to a “group” of the entire dataset for a single task.

5 Experiments

In this section, we evaluate the proposed MP-MTL method. We evaluate two instantiations of our method, Algorithm 2 and Algorithm 3 with respect to their ability to capture the low-rank and group-sparse patterns, respectively, in the model matrix. We use both synthetic and real-world datasets to evaluate these algorithms. All the algorithms were implemented in MATLAB.

5.1 Methods for Comparison

We use least-square loss and logistic loss for the least-square regression and binary classification problems, respectively.

For each setting, we evaluate three types of methods: 1) non-private STL methods, in which each task is learned independently without the introduction of any perturbation; 2) MP-MTL methods, including our proposed MP-MTL methods and baseline MP-MTL methods constructed by IP-MTL methods; and 3) non-private MTL methods, which correspond to the original MTL methods without the introduction of any perturbation.

To select the IP-MTL methods for constructing the baseline MP-MTL methods, because few such approaches have been proposed, we first consider DP-MTRL [33]. Since DP-MTRL does not consider the privacy-loss increase resulting from its iterative update procedure, we adopted the same composition technique as in our method. We also corrected DP-MTRL to consider the Lipschitz constants of the loss functions when computing the sensitivities in its 4th step, which were omitted. For all $i \in [m]$, the Lipschitz constant L_i of the loss function $\mathscr{L}_i$ is estimated as $L_i = \max_{j \in [n_i]} \left| \mathscr{L}_i'(\mathbf{x}_{ij}\mathbf{w}_i, y_{ij}) \right|$, which is smaller than the true value. Thus, intuitively, we add less noise to DP-MTRL than would otherwise be added. For the binary classification case, we still let DP-MTRL minimize the least-square loss because, in each of its outer iterations, DP-MTRL requires a closed-form solution to guarantee the theoretical privacy results. However, if the logistic loss is used, an iterative optimization is required in each outer iteration; consequently, the requirement of a closed-form solution cannot be satisfied. Therefore, DP-MTRL provides no privacy guarantee for the logistic loss. Moreover, it is not trivial to modify DP-MTRL for loss functions that require an iterative optimization in each outer iteration because additional leakage will occur in each inner iteration.

The DP-AGGR method proposed by Pathak et al. [58] which outputs an averaged model as the final solution, is also considered to be an IP-MTL method that transforms into a baseline MP-MTL method.

Remark 3.—We continue to refer to the baseline MP-MTL methods constructed by IP-MTL methods (DP-MTRL and DP-AGGR) using their respective names.

Differentially private STL methods are not considered because 1) empirically, they are always outperformed by non-private STL methods [16, 69], and 2) our MP-MTL method always outperforms STL methods, as will be demonstrated later.

5.2 Experimental Setting

For the non-private methods, the regularization parameters and the numbers of iterations were optimized via 5-fold cross-validation on the training data, and acceleration was used without considering the strong convexity of the loss function f. For the private methods, the regularization parameters, the number of iterations, the optimization strategy (whether to use acceleration and whether to consider strong convexity via adding ℓ_2^2 norm penalties), and the privacy-budget allocation hyper-parameters (a and Q) under each privacy loss ϵ were optimized via 5-fold cross-validation on the training data. In the case considering strong convexity, $\frac{\mu}{2}\|\mathbf{w}_i\|_2^2$ was added to each $\mathscr{L}_i$ with $\mu = 1e-3$.

Note that the parameter tuning step using cross-validation was not included in the privacy budget for the algorithms. In this paper, we regarded the hyper-parameters generated by cross-validation as given not only for our methods but also for the baseline methods (DP-AGGR and DP-MTRL). We plan to explore an effective cross-validation method using the minimum privacy budget with the optimum utility in future work.

For all the experiments, the δ values in the MP-MTL algorithms were set to $1/m\log(m)$, where m is the number of tasks as suggested by Abadi et al. [1], and the δ values in the baseline MP-MTL methods, i.e., DP-MTRL and DP-AGGR, were set in accordance with Proposition 3.

All experiments were replicated 100 times under each model setting.

5.3 Evaluation Metrics

We adopt the evaluation metrics commonly encountered in MTL approaches. For least-square regression, we use nMSE [18, 29], which is defined as the mean squared error (MSE) divided by the variance of the target vector. For binary classification, we use the average AUC [19], which is defined as the mean value of the area under the ROC curve for each task.

5.4 Simulation

We created a synthetic dataset as follows. The number of tasks was $m = 320$, the number of training samples for each task was $n_i = 30$, and the feature dimensionality of the training samples was $d = 30$. The entries of the training data $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ (for the i-th task) were randomly generated from the normal distribution $\mathcal{N}(0, 1)$ before being normalized such that the ℓ_2 norm of each sample was one.

To obtain a low-rank pattern, we first generated a covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$ as shown in Fig. 3 (a). Then, the model parameter matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ (see Fig. 3 (d)) was generated from a matrix variate normal (MVN) distribution [32], i.e., $\mathbf{W} \sim MVN(\mathbf{0}, \mathbf{I}, \mathbf{\Sigma})$. Whereas to obtain a group-sparse pattern, we generated the model parameter matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ such that the first 4 rows were nonzero. The values of the nonzero entries were generated from a uniform distribution in the range $[-50,-1] \cup [1,50]$.

Without loss of generality, we consider only the simulation of least-square regression. The results for logistic regression are similar. The response (target) vector for each task was $\mathbf{y}_i = \mathbf{X}_i\mathbf{w}_i + \varepsilon_i \in \mathbb{R}^{n_i} (i \in [m])$, where each entry in the vector ε_i was randomly generated from $\mathcal{N}(0, 1)$.

The test set was generated in the same manner; the number of test samples was $9n_i$.

5.4.1 Privacy Budget Allocation—The privacy-budget allocation strategies in Section 4.5 were evaluated based on the synthetic data associated with the low-rank model matrix. The results shown in Fig. 4 are from a 5-fold cross-validation on the training data. The prediction performances increase when acceleration is used, and achieve local optima at small positive values of the horizontal axes, which is consistent with our utility analyses. A local optimum exists in the negative horizontal axis in Fig. 4 (b) when acceleration is used—perhaps because m is not sufficiently large as assumed in Assumption 1.

5.4.2 Noise-to-Signal Ratio—Based on the setting in Section 5.4.1, the noise-to-signal ratios under the best privacy-budget allocation strategy (using acceleration and

considering basic convexity) are shown in Fig. 5, in which we executed Algorithm 2 on the synthetic data set with the low-rank pattern. In contrast, for DP-MTRL, $\log_{10}(\|\mathbf{E}\|_F/\|\widetilde{\Sigma}^{(t)}\|_F) = 0.2670 \pm 0.0075$ under the best iteration number $T = 1$. The output model matrices of DP-MTRL and our method are shown in Fig. 3 (e) and (f), and their respective covariance matrices are shown in Fig. 3 (b) and (c), respectively. These plots show that our method successfully learned the task relationships (every 75 tasks are similar) and the model parameters (which are similar to those in Fig. 3 d), while DP-MTRL did not. The results suggest that the high levels of noise added in our method had little influence on the output model matrix and the pattern in its covariance matrix, because our method adds noise only to the knowledge-sharing process and our method degrades to an STL method under high noise levels (as shown in Proposition 1). In contrast, in DP-MTRL, the output model matrix and the pattern in the covariance matrix are significantly affected or even destroyed because the noise was added directly to the model matrix, resulting in negative side effects. This result may also have occurred because DP-MTRL is *a local private learning algorithm*, which needs a much larger m to achieve acceptable utility (see the discussion in Section 2.2).

5.4.3 Privacy-Accuracy Tradeoff—In Fig. 6, the performances of both of our MP-MTL algorithms (i.e., Algorithms 2 and 3) fall between those of the non-private STL and non-private MTL methods, suggesting that our methods are useful as MTL methods but may be affected by the introduced noise. In Fig. 6 (a), Algorithm 3 underperforms compared with Algorithm 2, because the true model matrix is not group-sparse. DP-MTRL outperforms the STL method and our Algorithm 3 when ϵ is large because it suits the true model matrix, in which the relatedness among tasks is modeled by a graph. In Fig. 6 (b), the true model matrix is group-sparse and is not suitable for DP-MTRL; hence, DP-MTRL underperforms compared with the STL method even when ϵ is large. Algorithm 2 rivals Algorithm 3 because the true model matrix is also low-rank. In both panels of Fig. 6, Algorithm 2 rivals the non-private MTL when $\epsilon = 10$.

Fig. 7 shows the detailed performances for DP-MTRL and DP-AGGR corresponding to those in Fig. 6. Fig. 7 (c) is used to show that the accuracy of DP-AGGR grows with ϵ under the same setting as in Fig. 7 (b). As discussed previously, DP-AGGR performs only model averaging, which is not suitable for the true model matrices in both settings of Fig. 7 (a) and (b); hence, the accuracies of DP-AGGR are much worse than those of the respective STL methods.

5.4.4 Varying the Number of Tasks—Based on the setting in Section 5.4.1, the average performances of the first 20 of the 320 total tasks are shown in Fig. 8 under different numbers of training tasks. The accuracy increases with the number of tasks involved, which is consistent with our utility analyses. Specifically, when we consider all the epsilon values, the nMSE values are significantly smaller when the number of tasks m is larger: Mann–Whitney–Wilcoxon (MWW) tests [35] showed a p-value = 0.0068 for nMSE values of $m = 40$ smaller than those of $m = 20$, and a p-value = $1.57e-6$ for nMSE values of $m = 80$ smaller than those of $m = 40$. Second, for each epsilon value, when the $\log_{10} \epsilon$ is in the range of [−1.67,3], the nMSE values are significantly smaller when the number of tasks m is

larger: MWW tests showed that p-values ≤ 0.0013 for nMSE values of $m = 40$ smaller than those of $m = 20$, and p-values ≤ $1.55e-11$ for nMSE values of $m = 80$ smaller than those of $m = 40$. It is worth mentioning that the p-values are still small when $\log_{10} \epsilon$ is above 1 because the standard deviations are small.

5.5 Application

5.5.1 Data Description—We also evaluate the considered methods on the following two real datasets.

School Data.: The School dataset[1] is a popular dataset for MTL [29] that consists of the exam scores of 15,362 students from 139 secondary schools. Each student is described by 27 attributes, including both school-specific information and student-specific information such as gender and ethnic group. The problem of predicting exam scores for the students can be formulated as an MTL problem: the number of tasks is $m = 139$, the data dimensionality is $d = 27$, and the number of data samples is $\sum_i n_i = 15,362$.

LSOA II Data.: These data are from the Second Longitudinal Study of Aging (LSOA II) [2]. LSOA II was a collaborative study conducted by the National Center for Health Statistics (NCHS) and the National Institute of Aging from 1994 to 2000. A national representative sample of 9,447 subjects of 70 years of age and older were selected and interviewed. Three separate interviews were conducted with each subject, one each during the periods of 1994–1996, 1997–1998, and 1999–2000, referred to as WAVE 1, WAVE 2, and WAVE 3, respectively. Each wave of interviews included multiple modules covering a wide range of assessments. We used data from WAVE 2 and WAVE 3, which include a total of 4,299 sample subjects and 44 targets (each subject corresponded to 44 targets). We extracted 188 features from the WAVE 2 interviews. The targets include $m = 41$ binary outcomes used in this study. These outcomes fall into several categories: 7 measures of fundamental daily activity, 13 of extended daily activity, 5 of social involvement, 8 of medical condition, 4 of cognitive ability, and 4 of sensation condition.

The features include demographic, family structure, daily personal care, medical history, social activity, health opinions, behavior, nutrition, health insurance and income and asset attributes, the majority of which are binary values.

Both the targets and the features have missing values due to non-responsed and questionnaire filtering. The average missing value rates of the targets and features are 13.7% and 20.2%, respectively. To address the missing values among the features, we adopted the following preprocessing procedure. For the continuous features, missing values were imputed with the sample mean. For binary features, it is better to treat the missing values as a third category because the absence of a value may also carry important information. Therefore, two dummy variables were created for each binary feature with missing values (no third variable is necessary in such a case) resulting in a total of $d = 295$ features. To

1. http://www.cs.ucl.ac.uk/staff/a.argyriou/code/
2. https://www.cdc.gov/nchs/lsoa/lsoa2.htm.

address the missing values among the targets, we included the samples associated with the observed targets for each task, resulting in $\max_{i\in[m]} n_i = 3{,}473$.

For both real-world datasets, we randomly selected 30% of the samples from each task to form the training set and used the remaining samples as the test set. For all the tasks, each data point was normalized to have a unit length.

5.5.2 Privacy-Accuracy Tradeoff—From Fig. 9, we can observe results similar to those seen in Fig. 6. In addition, our MP-MTL algorithms outperform the baseline MP-MTL methods, DP-MTRL and DP-AGGR, especially when ϵ is small. DP-AGGR underperforms compared with the STL method because its model averaging approach assumes that the tasks are homogeneous. In Fig. 9 (b), the aAUC values of DP-MTRL and our Algorithms 2 and 3 increase slowly because the feature dimension is large and the number of tasks is insufficient, which is consistent with our utility analyses. Fig. 10 shows the detailed performances of DP-AGGR. In Fig. 10 (b), because the dimension is large and the number of tasks is insufficient, the accuracy of DP-AGGR barely grows with ϵ.

Because the MTL behavior may change when the training-data percentage (the size of the training data divided by the size of the entire dataset) changes, we evaluated the methods on both real-world datasets at different training-data percentages and achieved similar results; see the supplementary material (Appendix E) for more details.

6 Conclusions

In this paper, we discussed the potential security risks of multi-task learning approaches and presented a rigorous mathematical formulation of the model-protected multi-task learning (MP-MTL) problem. We proposed an algorithmic framework for implementing MP-MTL along with two concrete framework instantiations that learn the low-rank and group-sparse patterns in the model matrix. We demonstrated that our algorithms are guaranteed not to underperform compared with single-task learning methods under high noise levels. Privacy guarantees were provided. The utility analyses suggested that both the strong-convexity condition and the acceleration strategy improve the utility bounds and that the acceleration strategy also improves the runtime. A utility analysis for privacy-budget allocation yielded a recommendation for privacy budgets that are non-decreasing over the iterations. The experiments demonstrated that our algorithms significantly outperform baseline methods constructed by existing privacy-preserving MTL methods on the proposed model-protection problem. Some interesting future research directions include extending our approach for nonlinear or deep models and developing concrete MP-MTL algorithms for other MTL approaches and other optimization schemes.

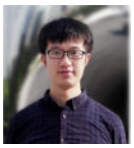
Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This research was supported in part by the National Science Foundation (NSF) under grants IIS-1565596, III-1615597 and IIS-1650723, in part by the Office of Naval Research (ONR) under grant number N00014-14-1-0631, and in part by the National Institutes of Health (NIH) under grants R00LM011392 and R21LM012060. In addition, we would like to thank Yuxiang Wang from the School of Computer Science at Carnegie Mellon University for his valuable comments on how to improve the properties of iterative MP-MTL algorithms.

Biography



Jian Liang received his B.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2012 and his Ph.D. degree from Tsinghua University, Beijing, China, in 2018. He is currently a senior researcher in the Wireless Security Products Department of the Cloud and Smart Industries Group at Tencent, Beijing. His research interests include authentication, transfer learning, semi-supervised learning, and imperfectly supervised learning. He won the Best Short Paper Award at the 2016 IEEE International Conference on Healthcare Informatics (ICHI).

Ziqi Liu Ziqi Liu is currently a Ph.D. student at Xi'an Jiaotong University under advisor Professor Qinghua Zheng. His research interests lie in scalable machine learning and nonparametric modeling. The major awards he has received include the WSDM'16 Best Research Paper Award.

Jiayu Zhou Jiayu Zhou is currently an assistant professor in the Department of Computer Science and Engineering at Michigan State University. He received his Ph.D. degree in computer science from Arizona State University in 2014. He has broad research interests in the fields of large-scale machine learning and data mining as well as biomedical informatics. He has served as a technical program committee member for premier conferences such as NIPS, ICML, and SIGKDD. His papers have received the Best Student Paper Award at the 2014 IEEE International Conference on Data Mining (ICDM), the Best Student Paper Award at the 2016 International Symposium on Biomedical Imaging (ISBI) and the Best Paper Award at IEEE Big Data 2016.

Xiaoqian Jiang is an assistant professor in the Department of Biomedical Informatics at the University of California, San Diego. He received his Ph.D. in computer science from Carnegie Mellon University. He is an associate editor of BMC Medical Informatics and Decision Making and serves as a member of the editorial board of the Journal of the American Medical Informatics Association. He works primarily in the fields of health data privacy and predictive models in biomedicine. Dr. Jiang was a recipient of the NIH K99/R00 award and won the Distinguished Paper Award from the American Medical Informatics Association Clinical Research Informatics (CRI) Summit in 2012 and 2013.

Changshui Zhang received his B.S. degree from Peking University, Beijing, China, in 1986 and his Ph.D. degree from Tsinghua University, Beijing, China, in 1992. He is currently a professor in the Department of Automation at Tsinghua University. He is a member of the editorial board of Pattern Recognition and an IEEE Fellow. His research interests include artificial intelligence, image processing, pattern recognition, machine learning and evolutionary computation.

Fei Wang is an assistant professor in the Division of Health Informatics of the Department of Healthcare Policy and Research at Cornell University. His major research interest lies in data analytics and its applications in health informatics. His papers have received over 3,700 citations, with an H-index of 33. He won the Best Student Paper Award at ICDM 2015, received a Best Research Paper nomination at ICDM 2010, and the Marco Romani Best Paper nomination at AMIA TBI 2014, and his papers were selected as Best Paper finalists at SDM 2011 and 2015. Dr. Wang is an active editor of the journal Data Mining and Knowledge Discovery, an associate editor of the Journal of Health Informatics Research and Smart Health, and a member of the editorial boards of Pattern Recognition and the International Journal of Big Data and Analytics in Healthcare. Dr. Wang is also the vice chair of the KDD working group at AMIA.

References

[1]. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, and Zhang L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 308–318. ACM, 2016.

[2]. Almeida MB and Martins AF. Fast and robust compressive summarization with dual decomposition and multi-task learning. In ACL (1), pages 196–206, 2013.

[3]. Amit Y, Fink M, Srebro N, and Ullman S. Uncovering shared structures in multiclass classification. In Proceedings of the 24th ICML, pages 17–24, 2007.

[4]. Ando RK and Zhang T. A framework for learning predictive structures from multiple tasks and unlabeled data. The Journal of Machine Learning Research, 6:1817–1853, 2005.

[5]. Argyriou A, Evgeniou T, and Pontil M. Multi-task feature learning. Advances in neural information processing systems, 19:41, 2007.

[6]. Argyriou A, Pontil M, Ying Y, and Micchelli CA. A spectral regularization framework for multi-task structure learning. In Advances in Neural Information Processing Systems, pages 25–32, 2007.

[7]. Argyriou A, Evgeniou T, and Pontil M. Convex multi-task feature learning. Machine Learning, 73(3):243–272, 2008.

[8]. Barak B, Chaudhuri K, Dwork C, Kale S, McSherry F, and Talwar K. Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release. In Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), pages 273–282, 2007.

[9]. Baytas IM, Yan M, Jain AK, and Zhou J. Asynchronous multi-task learning. In Data Mining (ICDM), 2016 IEEE 16th International Conference on, pages 11–20. IEEE, 2016.

[10]. Blocki J, Blum A, Datta A, and Sheffet O. The Johnson-lindenstrauss transform itself preserves differential privacy. In Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on, pages 410–419. IEEE, 2012.

[11]. Bonilla EV, Chai KMA, and Williams CK. Multi-task gaussian process prediction. In NIPs, volume 20, pages 153–160, 2007.

[12]. Caruana R. Multitask learning. Machine learning, 28(1): 41–75, 1997.

[13]. Chan T-HH, Shi E, and Song D. Private and continual release of statistics. ACM Transactions on Information and System Security (TISSEC), 14(3):26, 2011.

[14]. Chaudhuri K and Monteleoni C. Privacy-preserving logistic regression. In Advances in Neural Information Processing Systems, pages 289–296, 2008.

[15]. Chaudhuri K, Monteleoni C, and Sarwate A. Differentially Private Empirical Risk Minimization. Journal of Machine Learning Research (JMLR), 12:1069–1109, jul 2011. [PubMed: 21892342]

[16]. Chaudhuri K, Monteleoni C, and Sarwate AD. Differentially private empirical risk minimization. The Journal of Machine Learning Research, 12:1069–1109, 2011. [PubMed: 21892342]

[17]. Chen J, Tang L, Liu J, and Ye J. A convex formulation for learning shared structures from multiple tasks. In Proceedings of the 26th ICML, pages 137–144, 2009.

[18]. Chen J, Zhou J, and Ye J. Integrating low-rank and group-sparse structures for robust multi-task learning. In Proceedings of the 17th ACM SIGKDD Conference, pages 42–50. ACM, 2011.

[19]. Chen J, Liu J, and Ye J. Learning incoherent sparse and low-rank patterns from multiple tasks. ACM Transactions on Knowledge Discovery from Data (TKDD), 5(4):22, 2012. [PubMed: 24077658]

[20]. Dwork C, McSherry F, Nissim K, and Smith A. Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography Conference, pages 265–284. Springer, 2006.

[21]. Dwork C, Naor M, Pitassi T, and Rothblum GN. Differential privacy under continual observation. In Proceedings of the forty-second ACM symposium on Theory of computing, pages 715–724. ACM, 2010.

[22]. Dwork C, Roth A, et al. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4):211–407, 2014.

[23]. Dwork C, Talwar K, Thakurta A, and Zhang L. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In Proceedings of the 46th Annual ACM Symposium on Theory of Computing, pages 11–20. ACM, 2014.

[24]. Evgeniou A and Pontil M. Multi-task feature learning. In Advances in neural information processing systems, volume 19, page 41. The MIT Press, 2007.

[25]. Fei H and Huan J. Structured feature selection and task relationship inference for multi-task learning. Knowledge and information systems, 35(2):345–364, 2013.

[26]. Finlayson SG, Bowers JD, Ito J, Zittrain JL, Beam AL, and Kohane IS. Adversarial attacks on medical machine learning. Science, 363(6433):1287–1289, 2019. [PubMed: 30898923]

[27]. Ganta SR, Kasiviswanathan SP, and Smith A. Composition attacks and auxiliary information in data privacy. In Proceedings of the 14th ACM SIGKDD Conference, pages 265–273. ACM, 2008.

[28]. Goldreich O. Secure multi-party computation. Manuscript. Preliminary version, pages 86–97, 1998.

[29]. Gong P, Ye J, and Zhang C. Robust multi-task feature learning. In Proceedings of the 18th ACM SIGKDD Conference, pages 895–903. ACM, 2012.

[30]. Gu Q and Zhou J. Learning the shared subspace for multi-task clustering and transductive transfer classification. In Proceedings of ICDM, pages 159–168. IEEE, 2009.

[31]. Gu Q, Wang Z, and Liu H. Low-rank and sparse structure pursuit via alternating minimization. In AISTATS, volume 51, pages 600–609, 2016.

[32]. Gupta AK and Nagar DK. Matrix variate distributions, volume 104. CRC Press, 1999.

[33]. Gupta SK, Rana S, and Venkatesh S. Differentially private multi-task learning. In Pacific-Asia Workshop on Intelligence and Security Informatics, pages 101–113. Springer, 2016.

[34]. Han L and Zhang Y. Multi-stage multi-task learning with reduced rank. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[35]. Hollander M, Wolfe DA, and Chicken E. Nonparametric statistical methods, volume 751. John Wiley & Sons, 2013.

[36]. Hsu J, Roth A, and Ullman J. Differential privacy for the analyst via private equilibrium computation. In Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pages 341–350. ACM, 2013.

[37]. Jaggi M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In ICML (1), pages 427–435, 2013.

[38]. Jalali A, Sanghavi S, Ruan C, and Ravikumar PK. A dirty model for multi-task learning. In Advances in Neural Information Processing Systems, pages 964–972, 2010.

[39]. Ji S and Ye J. An accelerated gradient method for trace norm minimization. In Proceedings of the 26th annual international conference on machine learning, pages 457–464. ACM, 2009.

[40]. Ji Z, Jiang X, Li H, Xiong L, and Ohno-Machado L. Select and Label (S & L): a Task-Driven Privacy-Preserving Data Synthesization Framework. In Translational Bioinformatics Conference (TBC), Qingdao, China, 2014.

[41]. Jiang W, Xie C, and Zhang Z. Wishart mechanism for differentially private principal components analysis. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[42]. Kairouz P, Oh S, and Viswanath P. The composition theorem for differential privacy. IEEE Transactions on Information Theory, 2017.

[43]. Kasiviswanathan SP, Lee HK, Nissim K, Raskhodnikova S, and Smith A. What can we learn privately? SIAM Journal on Computing, 40(3):793–826, 2011.

[44]. Kearns M, Pai M, Roth A, and Ullman J. Mechanism design in large games: Incentives and privacy. In Proceedings of the 5th conference on Innovations in theoretical computer science, pages 403–410. ACM, 2014.

[45]. Kearns M, Pai MM, Rogers R, Roth A, and Ullman J. Robust mediators in large games. arXiv preprint arXiv:1512.02698, 2015.

[46]. Kurakin A, Goodfellow I, and Bengio S. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[47]. Li H, Xiong L, and Jiang X. Differentially Private Synthesization of Multi-Dimensional Data using Copula Functions. In 17th International Conference on Extending Database Technology (EDBT 2014), Athens, Greece, 2014.

[48]. Li J, Tian Y, Huang T, and Gao W. Probabilistic multi-task learning for visual saliency estimation in video. International Journal of Computer Vision, 90(2):150–165, 2010.

[49]. Liu J, Ji S, and Ye J. Multi-task feature learning via efficient l 2, 1-norm minimization. In Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pages 339–348. AUAI Press, 2009.

[50]. Mathew G and Obradovic Z. Distributed privacy preserving decision support system for predicting hospitalization risk in hospitals with insufficient data. In Machine Learning and Applications (ICMLA), 2012 11th International Conference on, volume 2, pages 178–183. IEEE, 2012.

[51]. Moosavi-Dezfooli S-M, Fawzi A, and Frossard P. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2574–2582, 2016.

[52]. Mopuri KR, Garg U, and Babu RV. Fast feature fool: A data independent approach to universal adversarial perturbations. arXiv preprint arXiv:1707.05572, 2017.

[53]. Ng K, Sun J, Hu J, and Wang F. Personalized predictive modeling and risk factor identification using patient similarity. AMIA Summits on Translational Science Proceedings, 2015:132, 2015.

[54]. Nie F, Huang H, Cai X, and Ding CH. Efficient and robust feature selection via joint 2, 1-norms minimization. In Advances in neural information processing systems, pages 1813–1821, 2010.

[55]. Nissim K, Raskhodnikova S, and Smith A. Smooth sensitivity and sampling in private data analysis. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pages 75–84. ACM, 2007.

[56]. Papernot N, McDaniel P, and Goodfellow I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277, 2016.

[57]. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, and Swami A. Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint arXiv:1602.02697, 2016.

[58]. Pathak M, Rane S, and Raj B. Multiparty differential privacy via aggregation of locally trained classifiers. In Advances in Neural Information Processing Systems, pages 1876–1884, 2010.

[59]. Pong TK, Tseng P, Ji S, and Ye J. Trace norm regularization: reformulations, algorithms, and multi-task learning. SIAM Journal on Optimization, 20(6):3465–3489, 2010.

[60]. Schmidt M, Roux NL, and Bach FR. Convergence rates of inexact proximal-gradient methods for convex optimization. In Advances in neural information processing systems, pages 1458–1466, 2011.

[61]. Shokri R and Shmatikov V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pages 1310–1321. ACM, 2015.

[62]. Song S, Chaudhuri K, and Sarwate AD. Stochastic gradient descent with differentially private updates. In 2013 IEEE Global Conference on Signal and Information Processing, pages 245–248. IEEE, dec 2013. ISBN 978–1-4799–0248-4. doi: 10.1109/GlobalSIP.2013.6736861.

[63]. Su C, Yang F, Zhang S, Tian Q, Davis LS, and Gao W. Multi-task learning with low rank attribute embedding for multi-camera person re-identification. IEEE transactions on pattern analysis and machine intelligence, 40(5):1167–1181, 2018. [PubMed: 28287958]

[64]. Sun Z, Wang F, and Hu J. Linkage: An approach for comprehensive risk prediction for care management. In Proceedings of the 21th ACM SIGKDD Conference, pages 1145–1154. ACM, 2015.

[65]. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, and Fergus R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[66]. Turlach BA, Venables WN, and Wright SJ. Simultaneous variable selection. Technometrics, 47(3):349–363, 2005.

[67]. Vadhan S. The complexity of differential privacy. Work. Pap., Cent. Res. Comput. Soc., Harvard Univ. http://privacytools.seas.harvard.edu/publications/complexity-differential-privacy, 2016.

[68]. Wang X, Wang F, Hu J, and Sorrentino R. Exploring joint disease risk prediction. In AMIA Annual Symposium Proceedings, volume 2014, page 1180. American Medical Informatics Association, 2014.

[69]. Wang Y-X, Fienberg S, and Smola A. Privacy for Free: Posterior Sampling and Stochastic Gradient Monte Carlo. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 2493–2502, 2015.

[70]. Wright SJ, Nowak RD, and Figueiredo MA. Sparse reconstruction by separable approximation. IEEE Transactions on Signal Processing, 57(7):2479–2493, 2009.

[71]. Xu M and Lafferty JD. Conditional sparse coding and grouped multivariate regression. In Proceedings of the 29th ICML, pages 1479–1486, 2012.

[72]. Zantedeschi V, Nicolae M-I, and Rawat A. Efficient defenses against adversarial attacks. arXiv preprint arXiv:1707.06728, 2017.

[73]. Zhang J, Zhang Z, Xiao X, Yang Y, and Winslett M. Functional mechanism. Proceedings of the VLDB Endowment, 5(11):1364–1375, jul 2012. ISSN 21508097. doi: 10.14778/2350229.2350253.

[74]. Zhang P, Wang F, Hu J, and Sorrentino R. Towards personalized medicine: leveraging patient similarity and drug similarity analytics. AMIA Summits on Translational Science Proceedings, 2014:132, 2014.

[75]. Zhang T, Ghanem B, Liu S, and Ahuja N. Robust visual tracking via multi-task sparse learning. In IEEE CVPR Conference, pages 2042–2049. IEEE, 2012.

[76]. Zhang Y and Yang Q. A survey on multi-task learning. arXiv preprint arXiv:1707.08114, 2017.

[77]. Zhang Y and Yeung D-Y. A convex formulation for learning task relationships in multi-task learning. In Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, pages 733–742. AUAI Press, 2010.

[78]. Zhen X, Yu M, He X, and Li S. Multi-target regression via robust low-rank learning. IEEE transactions on pattern analysis and machine intelligence, 40(2):497–504, 2018. [PubMed: 28368816]

[79]. Zhou J, Chen J, and Ye J. Clustered multi-task learning via alternating structure optimization. In Advances in neural information processing systems, pages 702–710, 2011.

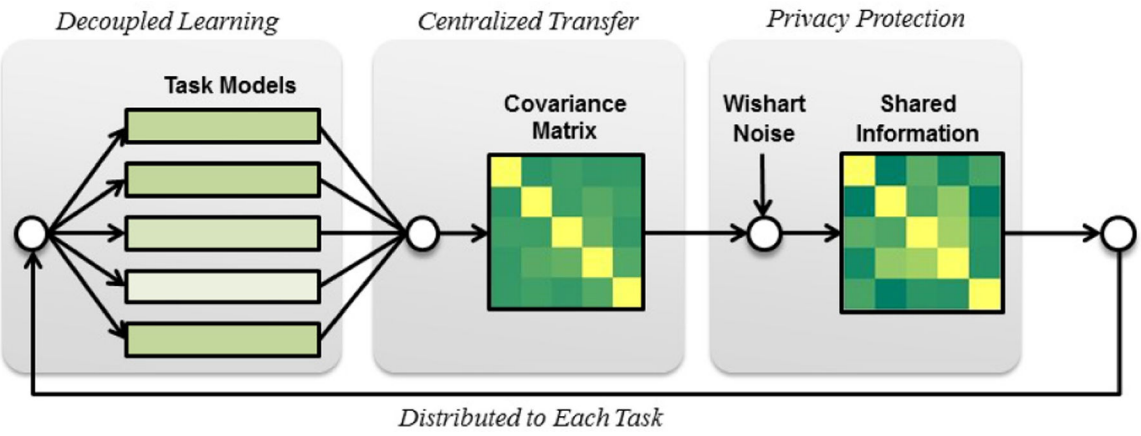


Figure 1.

Model-protected multi-task learning framework. The solution process is a recursive two-step procedure. The first step is a decoupled learning procedure in which the model parameters for each task are estimated independently using the precomputed shared information among tasks. The second step is a centralized transfer procedure in which the information shared among tasks is extracted for distribution to each task for the decoupled learning procedure in the next step. The shared information is extracted from the tasks' covariance matrix, into which Wishart noise is introduced to ensure model security.

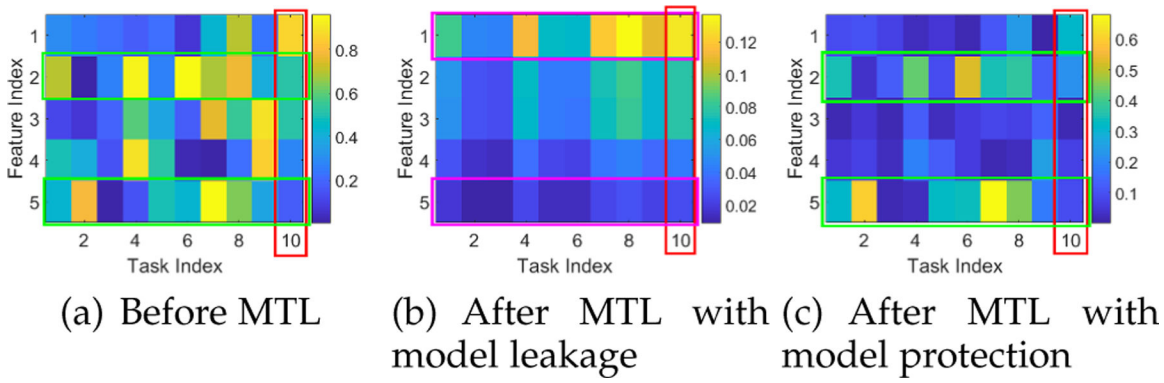


Figure 2.

Examples of model leakage and model protection showing model matrices, where columns correspond to tasks and rows correspond to features. The columns shown have been divided by their respective ℓ_2 norms.

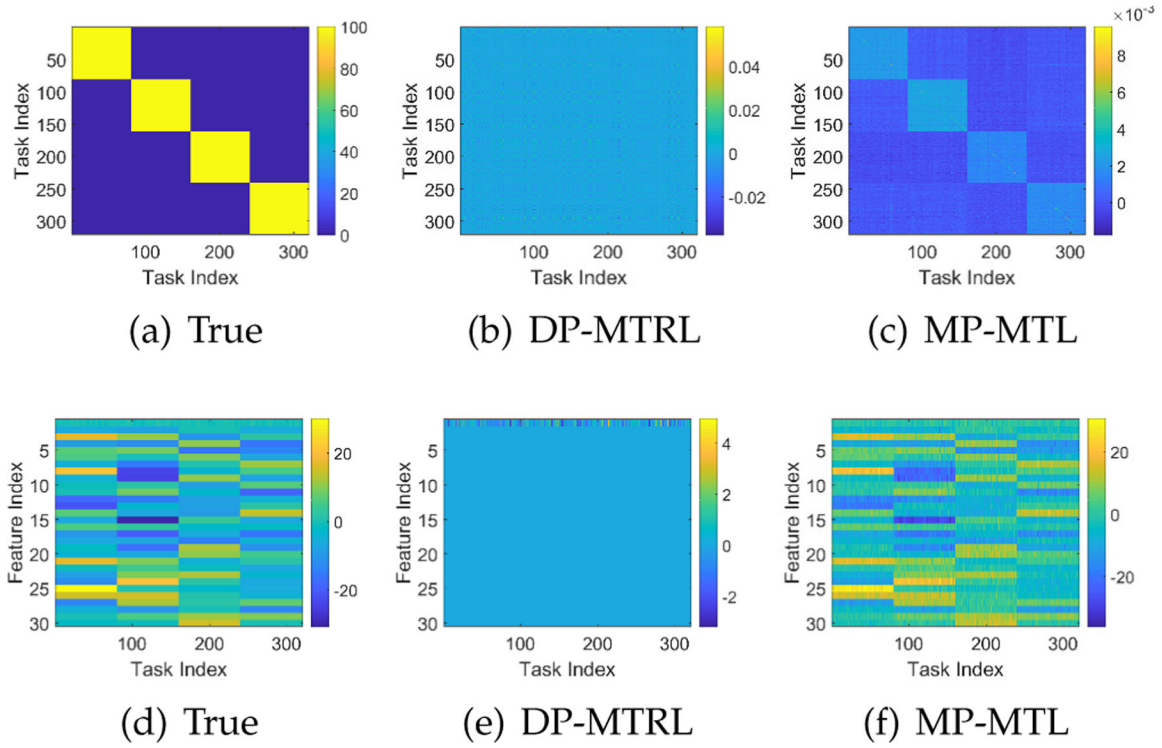


Figure 3.

Task relationships and output model matrices for the synthetic data experiments: (a), (b) and (c) are task relationship matrices, (d), (e) and (f) are the output model matrices. In (a), a high-value entry of the matrix indicates that a pair of tasks have similar model parameters. As in (d), a column shows the model parameters of one task, and every 75 columns are similar, which is consistent with (a). (b) and (c) are learned relationship matrices, in which the task-relationship patterns reflected by the relative values of entries are suppose to be similar to the pattern in (a). In addition, (e) and (f) are learned model matrices, and every 75 columns are supposed to be similar as in (d). Moreover, in (e) and (f), the relative values in each column are also supposed to be similar to those in the correspond column in (d). The results shown are the averages of 100 runs with $\epsilon = 0.1$.

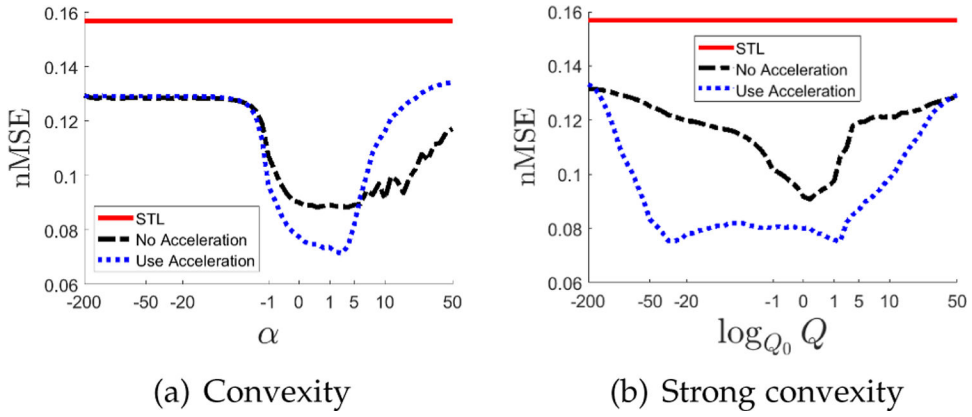


Figure 4.

Evaluations for privacy-budget allocation strategies. In (a), we set $\epsilon_t = \Theta(t^{\alpha})$, for $t \in [T]$; in (b), we set $\epsilon_t = \Theta(Q^{-t})$, for $t \in [T]$. $Q_0 = 1 - \sqrt{\mu} \approx 0.9684$. The results shown are averages of 100 runs with $\epsilon = 1$. For the non-private MTL method, the nMSE was 0.0140.

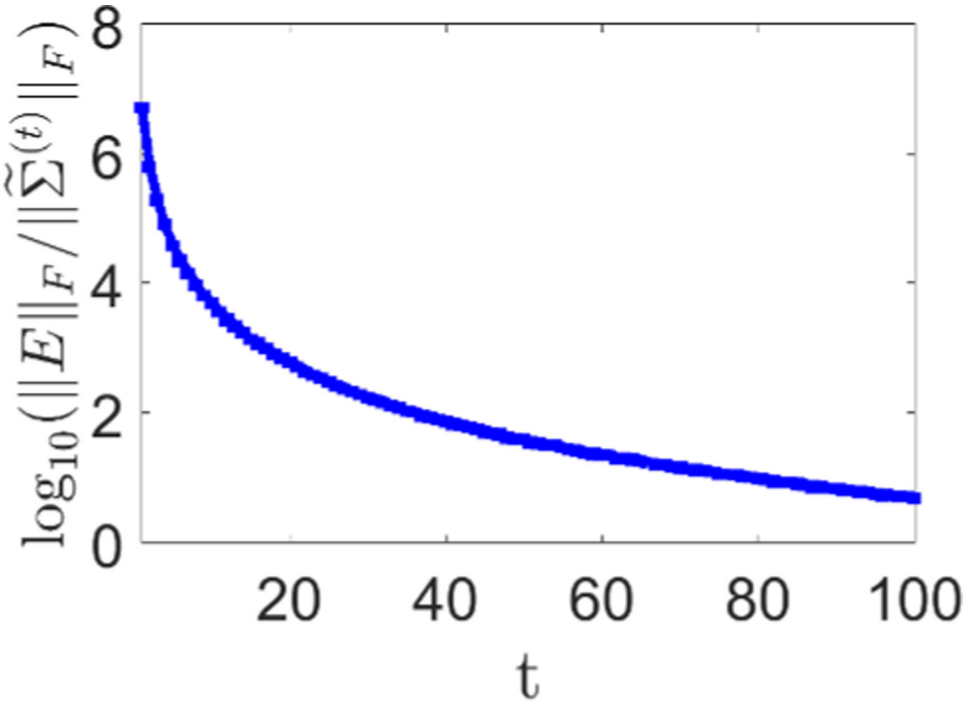
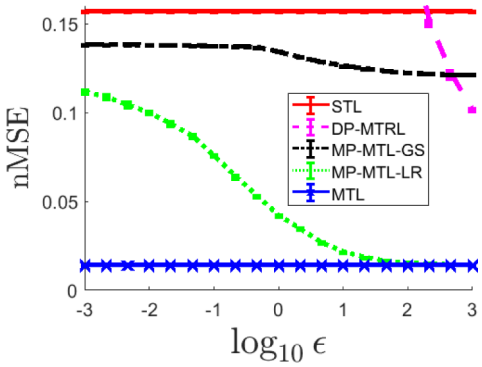
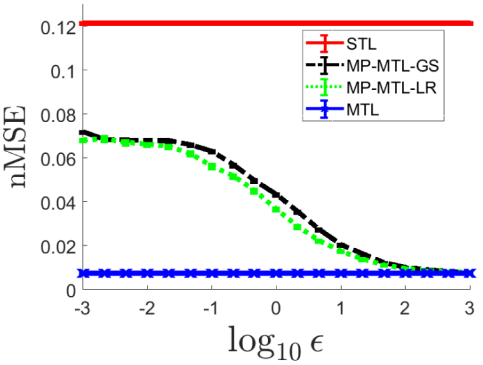


Figure 5.

Noise-to-signal ratios over the iterations of Algorithm 2. The results shown are averages of 100 runs with $\epsilon = 0.1$.



(a) Low-rank pattern



(b) Group-sparse pattern

Figure 6.

Privacy-accuracy tradeoff on synthetic datasets. For (a), the data associated with the low-rank model matrix were used; for (b), the data associated with the group-sparse model matrix were used. MP-MTL-LR denotes Algorithm 2, MP-MTL-GS denotes Algorithm 3, and STL denotes the ℓ_2-norm-penalized STL method. In both panels, STL and MTL denote non-private methods. In (b), the nMSEs of DP-MTRL are above 0.16; in both panels, the nMSEs of DP-AGGR are above 0.78. Detailed results of DP-MTRL and DP-AGGR are presented in Fig. 7.

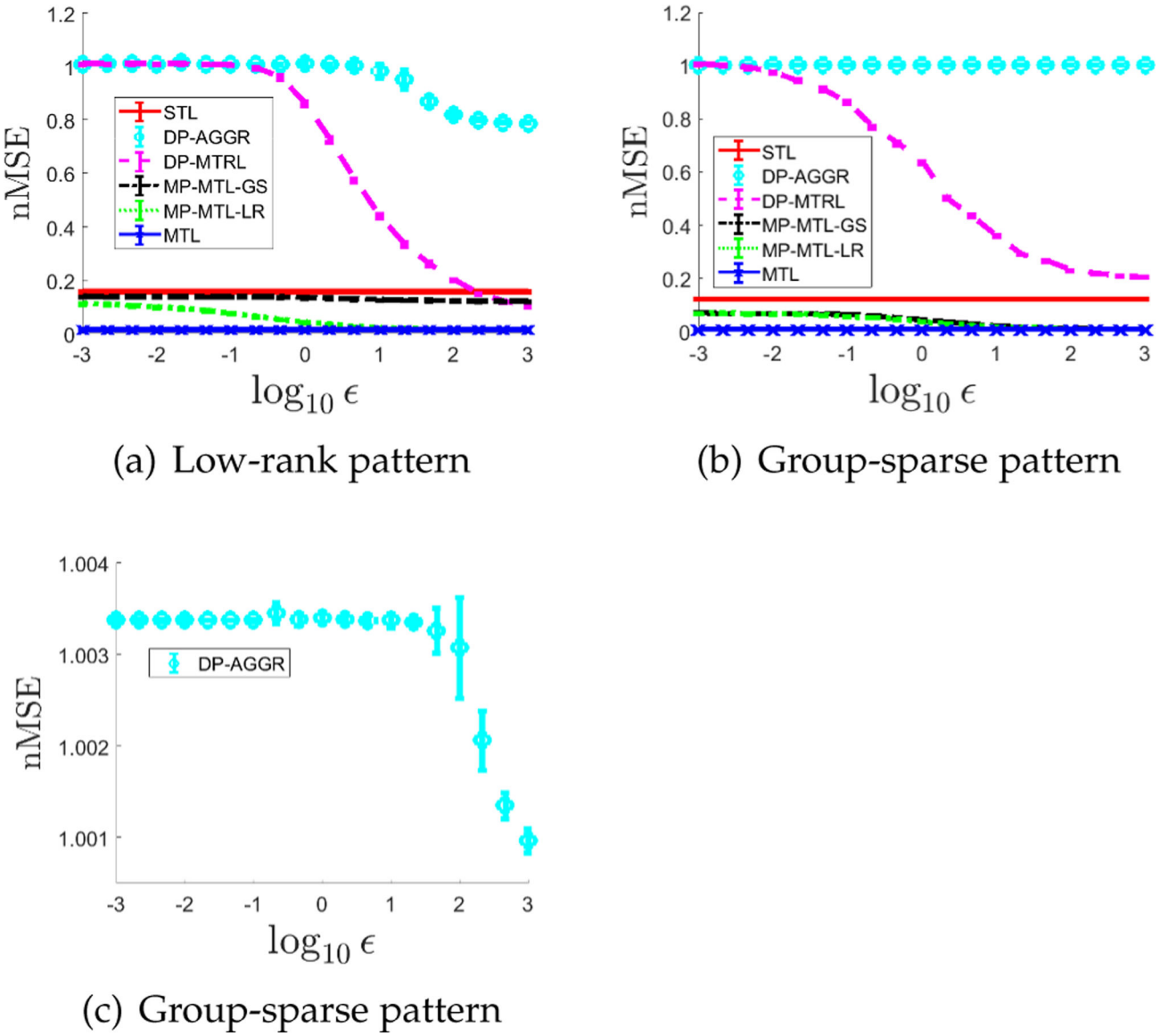


Figure 7.

Detailed privacy-accuracy tradeoff on synthetic datasets for DP-MTRL and DP-AGGR. For (a), the data associated with the low-rank model matrix were used; for (b) and (c), the data associated with the group-sparse model matrix were used. In (c), the plot shows the same performances of DP-AGGR as those in (b) but with a finer vertical axis. Other settings are the same as those used for Fig. 6.

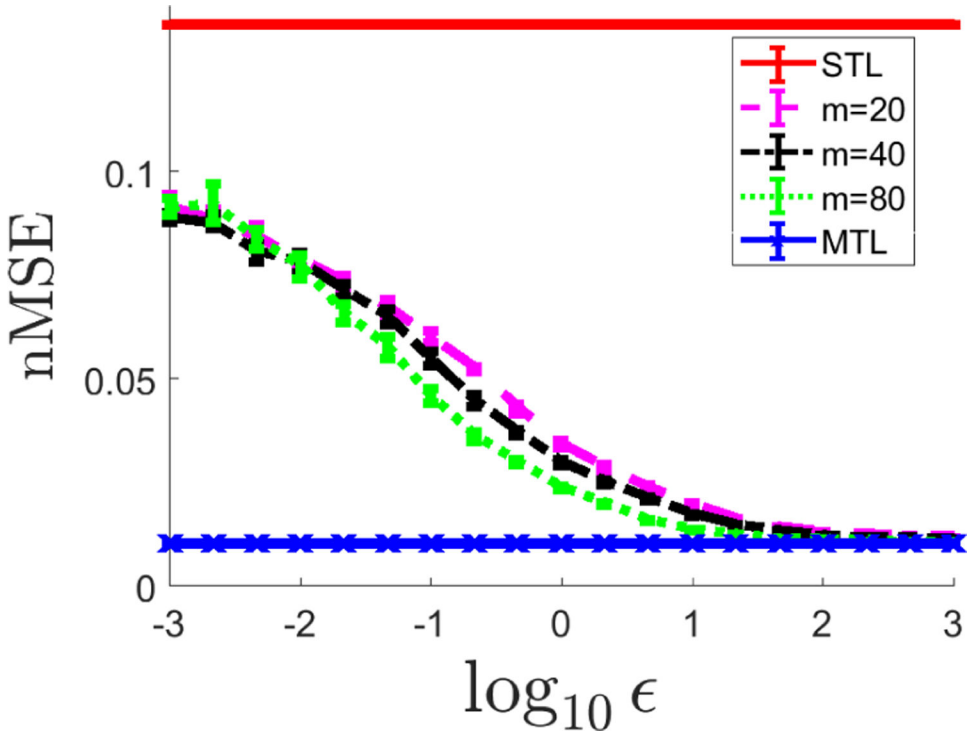
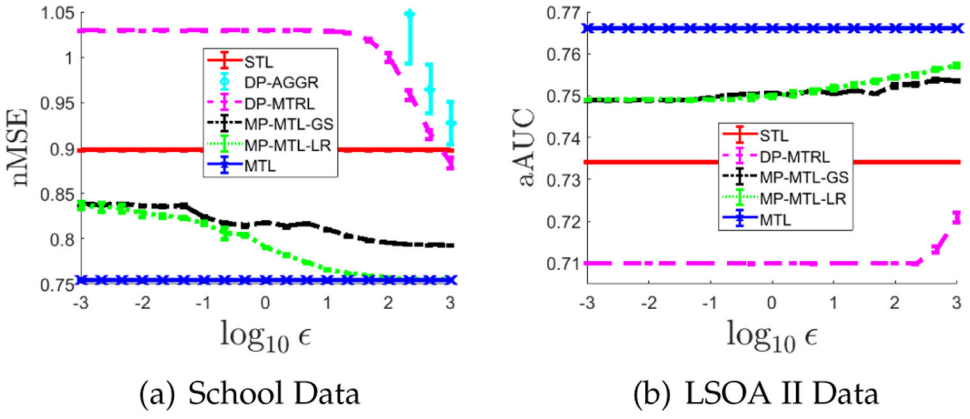


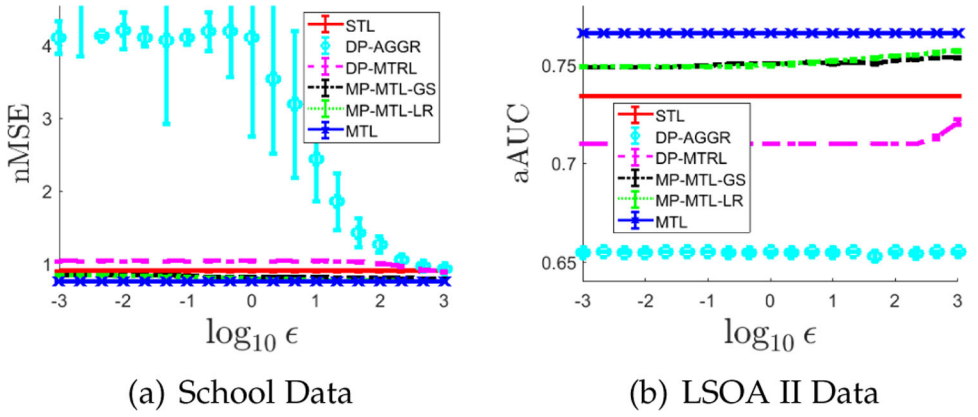
Figure 8.
Behaviors based on the number of tasks m used for training. We used 320 tasks for MTL training.



(a) School Data (b) LSOA II Data

Figure 9.

Privacy-accuracy tradeoff on real-world datasets. In both panels, MTL denotes the method with the best performance among the four non-private MTL methods proposed by Ji and Ye [39], Liu et al. [49], Zhang and Yeung [77] and DP-AGGR without perturbations; MP-MTL-LR denotes Algorithm 2, whereas MP-MTL-GS denotes Algorithm 3; STL denotes the method with the better performance between the ℓ_1- and ℓ_2-regularized methods. In (b), the aAUCs of DP-AGGR are below 0.66. The detailed performances of DP-AGGR are presented in Fig. 10.



(a) School Data (b) LSOA II Data

Figure 10.

Detailed privacy-accuracy tradeoff on real-world datasets for DP-AGGR. All the settings are the same as those in Fig. 9.

Table 1

Notations and Symbols

$[k]$	the index set $\{1, 2, \ldots, k\}$
$[-i]$	the index set with index i removed
$\|\cdot\|_*$	the trace norm of a matrix (sum of the singular values of the matrix)
$\|\cdot\|_{2,1}$	the $\ell_{2,1}$ norm of a matrix (sum of the ℓ_2 norms of the row vectors of the matrix)
$\mathrm{tr}(\cdot)$	the trace of a matrix (sum of the diagonal elements of the matrix)
$\sigma_j(\cdot)$	the j-th largest singular value of a matrix, $j \in [m]$

Table 2

Utility results.

			Low rank	Group sparse
$\delta = 0$	No Acceleration	Convex	$\mathcal{O}((\frac{d\log(d)}{\sqrt{m}\epsilon})^{\frac{1}{3}})$	$\mathcal{O}((\frac{\log(d)}{m\epsilon})^{\frac{1}{3}})$
		Strong convex	$\mathcal{O}((\frac{d\log(d)}{\sqrt{m}\epsilon})^{\frac{1}{2}})$	$\mathcal{O}((\frac{\log(d)}{m\epsilon})^{\frac{1}{2}})$
	Use Acceleration	Convex	$\mathcal{O}((\frac{d\log(d)}{\sqrt{m}\epsilon})^{\frac{2}{5}})$	$\mathcal{O}((\frac{\log(d)}{m\epsilon})^{\frac{2}{5}})$
		Strong convex	$\mathcal{O}(\frac{d\log(d)}{\sqrt{m}\epsilon})$	$\mathcal{O}(\frac{\log(d)}{m\epsilon})$
$\delta > 0$	No Acceleration	Convex	$\mathcal{O}((\frac{d\log(d)}{\sqrt{m}\epsilon})^{\frac{2}{5}})$	$\mathcal{O}((\frac{\log(d)}{m\epsilon})^{\frac{2}{5}})$
		Strong convex	$\mathcal{O}((\frac{d\log(d)}{\sqrt{m}\epsilon})^{\frac{1}{2}})$	$\mathcal{O}((\frac{\log(d)}{m\epsilon})^{\frac{1}{2}})$
	Use Acceleration	Convex	$\mathcal{O}((\frac{d\log(d)}{\sqrt{m}\epsilon})^{\frac{4}{9}})$	$\mathcal{O}((\frac{\log(d)}{m\epsilon})^{\frac{4}{9}})$
		Strong convex	$\mathcal{O}(\frac{d\log(d)}{\sqrt{m}\epsilon})$	$\mathcal{O}(\frac{\log(d)}{m\epsilon})$