On Privacy and Personalization in Cross-Silo Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Although differential privacy (DP) is a well-studied topic in cross-device federated learning (FL), there is a lack of work considering DP for cross-silo FL, a setting characterized by a limited number of clients each containing many data subjects. In cross-silo FL, usual notions of *client-level* privacy are less suitable as real-world privacy regulations typically concern in-silo data subjects rather than the silos themselves. In this work, we instead consider the more realistic notion of *silo-specific item-level* privacy, where silos set their own privacy targets for local examples. Under this setting we reconsider the roles of privacy and personalization in federated learning. In particular, we show that mean-regularized multi-task learning (MR-MTL), a simple personalization framework, is a surprisingly strong baseline for cross-silo FL: under stronger privacy, silos are further incentivized to "federate" with each other to mitigate DP noise, resulting in consistent improvements relative to standard cross-device baselines. We provide a thorough empirical study of competing methods as well as a theoretical characterization of MR-MTL for a mean estimation problem, highlighting the interplay between privacy and cross-silo data heterogeneity. Our work serves to establish baselines for private cross-silo FL as well as identify key directions for future work in this area.

1 Introduction

Recent advances in machine learning often rely on large, centralized datasets [66, 18, 50], but curating such data may not always be viable, particularly when the data contains private information and must remain siloed across clients (e.g. mobile devices or hospitals). Recently, federated learning (FL) [55, 40] has emerged as a paradigm for learning from such distributed data, but it has been shown that its data minimization principle alone may not provide adequate privacy protection for participants [77, 78]. To obtain formal privacy guarantees, there has thus been extensive work applying *differential privacy* (DP) [22, 23] to various parts of the FL pipeline (e.g. [27, 57, 34, 38, 3, 48, 35, 65, 29]).

These approaches are typically designed for *client-level* DP in that they protect the federated clients, such as mobile devices ("user-level"), tasks in multi-task learning ("task-level"), or data silos like institutions ("silo-level"). DP is then achieved by appropriately clipping and noising model updates from the clients. While client-level DP is considered a strong privacy notion as all data of a single client is protected [11], it may not be suitable for *cross-silo* FL, where there are fewer clients, but each hold many data subjects that require protection. For example, when hospitals/banks/schools wish to federate patient/customer/student records, it is the people owning those records rather than the participating silos that should be protected; in fact, laws and regulations may mandate such participation in FL be disclosed publicly [73], nullifying any client-level DP guarantees.

In this work we instead consider a more natural model of *silo-specific item-level privacy* (Fig. 1): the k-th silo may set its own $(\varepsilon_k, \delta_k)$ item-level DP target for any learning process with respect to its local dataset. With this formulation in mind, we then reconsider the impact of privacy, heterogeneity,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

and personalization in cross-silo FL. In particular, we explore existing baselines for FL (mostly developed in cross-device settings) across private cross-silo benchmarks. Surprisingly, we find that the simple baseline of mean-regularized MTL (MR-MTL) has many advantages for this setting relative to other more common (and possibly more complex) methods. We then further analyze the performance of MR-MTL under varying levels of heterogeneity and privacy, both in theory and in practice. In addition to establishing baselines for cross-silo FL, our work identifies several interesting open problems in this area. We summarize our contributions below:

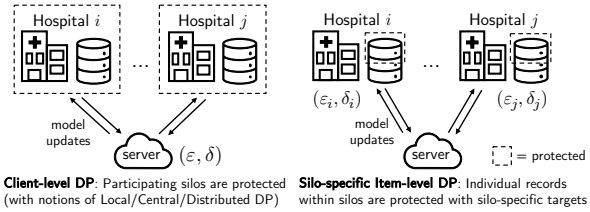

Figure 1: **Client-level** DP vs **Silo-specific Item-level** DP.

- We consider the notion of *silo-specific item-level differential privacy* (DP) as a more realistic privacy model for *cross-silo federated learning* (FL). We analyze its implications on existing FL algorithms and, in particular, how it interfaces with data heterogeneity across silos.
- We empirically show that mean-regularized multi-task learning (MR-MTL), a simple form of model personalization, is a remarkably strong baseline under silo-specific item-level DP. Core to its effectiveness is its ability to (roughly) interpolate on the model personalization spectrum between local training and FedAvg with little to no additional privacy overhead.
- We provide a theoretical analysis of MR-MTL under mean estimation and characterize how MR-MTL navigates the tension between privacy and cross-silo data heterogeneity.
- Finally, we also examine the complications of deploying an optimal MR-MTL instance stemming from the privacy cost of *hyperparameter tuning*. Our reasoning also applies to other personalization methods whose advantage over local training and/or FedAvg hinges on selecting the best hyperparameter(s). This raises important questions around the practicality of leveraging personalization to balance the emerging tradeoffs under silo-specific item-level DP.

2 Preliminaries

Federated Learning (FL) [55, 45, 40] is a distributed learning paradigm with an emphasis on data protection: in every training round, each client (participant) downloads the current global model from a central server, trains it with the local dataset, and uploads the model changes (instead of the data) back to the server, which then aggregates the changes into a new global model. A basic instantiation of FL is FedAvg [55], where clients are stateless and the server performs a simple (weighted) average. *Cross-device* FL refers to settings with many clients each with limited data, bandwidth, availability, etc. (e.g. mobile devices). In contrast, *cross-silo* FL typically involves less clients (e.g. banks, schools, hospitals) but each with more resources. Two distinguishing characteristics of cross-silo FL relevant to our work are that (1) silos may have sufficient data to fit a reasonable local model *without* FL, and (2) each data point in a silo tends to map to a data subject (a person) requiring privacy protection.

Differential Privacy (DP). Despite its promises to mitigate systemic privacy risks, FL by itself does not provide formal privacy guarantees for participants' data [40, 78, 77], and differential privacy is often used in conjunction with FL to ensure that an algorithm does not leak the privacy of its inputs.

Definition 2.1 (Differential Privacy [23])**.** A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$, where $\mathcal{X}^n$ is a dataset with n items and $\mathcal{Y}$ is the set of outputs, is (ε, δ)-DP if for any subset $S \subseteq \mathcal{Y}$ and any neighboring $x, x' \in \mathcal{X}^n$ differing in only one item (by addition, removal, or replacement), we have

$$\Pr[M(x) \in S] \leq \exp(\varepsilon) \cdot \Pr[M(x') \in S] + \delta. \tag{1}$$

To apply DP to a dataset query, one commonly used method is the Gaussian mechanism [23], which involves bounding the contribution (ℓ^2-norm) of each item followed by adding Gaussian noise proportional to that bound onto the aggregate. To apply DP in FL, one needs to define the "dataset" to protect; typically, as in *client-level* DP, this is the set of FL participants and thus the model updates from each participant in every round should be bounded and noised. In learning settings, we need to repeatedly query a dataset and the privacy guarantee composes. We use DP-SGD [1] for ensuring item-level DP for model training, and we use a relaxation of DP known as Rényi DP [61] for tight privacy composition. In certain FL algorithms, clients also perform additional work such as cluster selection [53, 28] that incurs privacy overhead with respect to its local dataset that must be accounted for independently from DP-SGD. See Appendix A for additional background.

Personalized FL. Model personalization is a key technique for improving utility under data heterogeneity across silos.[1] Past work has examined the roles of local adaptation [75, 80, 15], multi-task learning [70, 69], clustering [28, 17, 53, 69], public data [83, 53], meta learning [37, 43, 26], or other forms of model mixtures [46, 44, 53, 33, 19, 2]. Notably, many methods leverage extra computation to some extent (e.g. extra iterations [46, 44, 15] or cluster selection [28, 53]), resulting in privacy overhead under silo-specific item-level DP. Of particular interest is the family of mean-regularized multi-task learning (MR-MTL) methods [25, 71, 33, 32] (see Algorithm A1 for a typical instantiation). We find that MR-MTL, while extremely simple, is a strong baseline for private cross-silo FL.

3 Revisiting the Privacy Model for Cross-Silo Federated Learning

To date, the prevalent privacy model for federated learning has been to protect the participants of the FL protocol (client-level DP). For cross-silo FL, however, several factors render client-level DP less appropriate: (1) Cross-silo FL often involves a small number of clients, meaning that it can be utility-wise more costly to attain the same privacy compared to cross-device FL. For example, privacy amplification via sampling [1, 62] may not apply on a client level, as silos typically participate in every round. (2) Client-level DP canonically defines a shared privacy guarantee for all participants (as is often the premise for many existing methods), but in real-world cross-silo settings, participants under different jurisdictions (e.g. states) may have varying privacy requirements and may opt for different privacy-utility tradeoffs. (3) While silo-level protection implies item-level protection, it may be too stringent in practice as silos often have large local datasets. These unique properties for private cross-silo learning motivate us to consider *silo-specific item-level* DP as an alternative model.

Characteristics of Silo-specific Item-level Privacy. Under silo-specific item-level DP (Fig. 1), each silo k sets its own item-level DP budget $(\varepsilon_k, \delta_k)$ w.r.t. its own local dataset. Critically, DP is now defined over disjoint datasets of the individual silos, rather than the combined dataset of all silos.[2] To instantiate this setup in FL, the k-th silo can simply run DP-SGD [1] with a noise scale calibrated to gradually spend its privacy budget over T training rounds, and return the noisy model update at each round. This change in privacy notion has several important implications on the dynamics of FL:

1. **Less reliance on a trusted server.** By the post-processing property of DP, all model iterates of silo k satisfy (at least) $(\varepsilon_k, \delta_k)$-DP against external adversaries, including the server that orchestrates the FL protocol as well as other silos [23, 84]. In contrast, client-level DP necessitates some trust on the central server for non-local DP guarantees, even for distributed DP methods [24, 16, 38].
2. **Silos incur privacy costs with queries to their data, but *not* with participation in FL**, as a corollary to the above. In contrast, local training is 'free' under client-level DP, but participation in FL can compromise privacy. Two immediate consequences are (1) local training and FedAvg now have *identical* privacy costs, and (2) *local finetuning* may no longer work as expected (Fig. 2).
3. **Tradeoff emerges between costs from privacy and heterogeneity.** As silos add their own independent noises to satisfy their own DP requirements, these noises are reflected in their model updates and can thus be mitigated when the model updates are *averaged* (e.g. via FedAvg), leading to a smaller utility drop for the shared model. On the other hand, federation also means that the shared model may suffer from *data heterogeneity* (non-iid data across silos). This intuition is observed in Fig. 2: while local training may outperform FedAvg without privacy (result of heterogeneity), the opposite can be true when privacy is added (result of noise variance reduction).

The second and last implications are particularly interesting because they suggest that *model personalization* can play a key and distinct role in our privacy setting. Specifically, local training (no FL participation) and FedAvg (full FL participation) are two ends of a *personalization spectrum* with identical privacy costs; if local training minimizes the effect of data heterogeneity but enjoys no DP noise variance reduction, and contrarily for FedAvg, it is natural to ask whether personalizing models unique to each silo can improve utility, and, if so, what methods would work best.

Related Settings. Most existing literature on differentially private FL has primarily focused on client-level DP and cross-device FL (e.g. [27, 58, 35, 29, 39, 38, 4]). The application of DP in cross-

[1] Note that "model personalization" refers to customizing for each *client* in FL rather than a specific person.

[2] A record in such a combined dataset is at most $(\max_i \varepsilon_i, \max_i \delta_i)$-DP [60, 79, 48]. Moreover, if multiple records (either within a silo or across silos) map to the same person, then it is more intricate to protect the person rather than their records. Here we focus on the case where records bijectively map to entities (e.g. students attending exactly one school). See Appendix B for discussions.

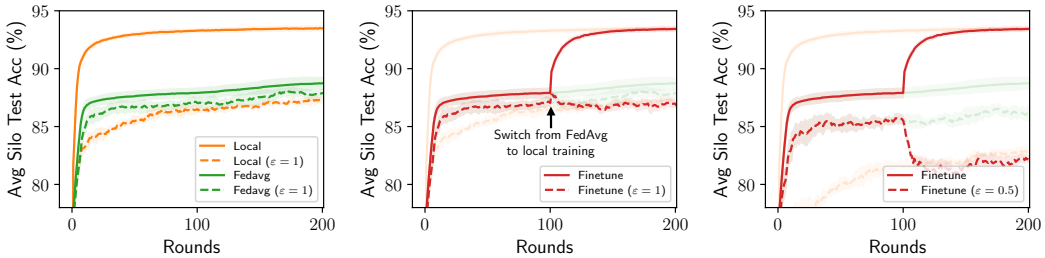

Figure 2: **We observe two notable phenomena under *silo-specific item-level DP***: (1) FedAvg can serve to cancel out per-silo DP noise and thus outperform local training even when the latter works better without privacy (*left*); (2) Local finetuning [80, 15] (FedAvg followed by local training) may not improve utility as expected, as the effect of noise reduction is removed when finetuning begins (*mid* & *right*). For all plots (mean test acc $\pm$ std on the Vehicle dataset w/ 200 rounds and 5 seeds), all silos have the same labelled ε target with $\delta = 10^{-7}$.

silo FL, and particularly the setting where each silo defines its *own DP targets* to protect the items of its *own local dataset*, is relatively underexplored. Privacy notions closest to ours first appeared in [72, 43, 84, 48]. The work of [72] studies the setting where each client adds its own one-shot noise onto its update to the server, but in learning scenarios the privacy protection reduces to client-level DP. The work of [43] proposes the notion of "task-global DP" for meta-learning where each task's local dataset is protected with item-level DP. While related to our setting, this relaxation was proposed to boost utility, and the implications of data heterogeneity, DP noise reduction from federation, and the personalization spectrum are unexplored. The work of [84] studies an analogous setting but with a focus on f-DP [20], and similarly does not generally analyze the role of personalization or the emerging empirical phenomena (e.g. Figs. 2 and 5). The concurrent work of [48] also considers silo-specific item-level DP, but it only focuses on learning a better global model. Finally, [34] is relevant for cross-silo FL but instead focuses on applying client-level DP with security primitives.

4 Baselines for Private Cross-Silo Federated Learning

With the characteristics from §3 in mind, we now explore various methods on cross-silo benchmarks. We defer additional details as well as results on more settings and datasets to the appendix.

Datasets. We consider two real-world datasets with cross-silo characteristics: Vehicle [21] and School [30]. Vehicle is a binary classification dataset with $K = 23$ silos, each with measurements for a road segment as training examples for classifying the type of a passing vehicle. Each silo k has $872 \leq n_k \leq 1933$ examples. School is a regression dataset for predicting the exam scores of 15,362 anonymized students across $K = 139$ secondary schools; each school k has $22 \leq n_k \leq 251$ examples capturing attributes such as the student's past score and the availability of free meals.

Benchmark Methods. We consider several representative methods in the personalized FL literature beyond local training and FedAvg [55]: Local finetuning [75, 80, 15] (a simple but strong baseline for model personalization), Ditto [44] (state-of-the-art personalization method), Mocha [70] (personalization with task relationship learning), IFCA [28, 53] (state-of-the-art hard clustering method for client models), and the mean-regularized multi-task learning (MR-MTL) methods [25, 71, 33, 32] (which we analyze in §5). Importantly, for fair comparison under silo-specific item-level DP, we align all benchmark methods on the total privacy budget by first restricting the total number of iterations and then account for any necessary privacy overheads (in the form of necessary extra steps [44] or cluster selection for [28, 53]). Many other methods can either be reduced to one of the above under simplified settings (e.g. [37, 46]) or are unsuitable due to large privacy overheads (e.g. [26, 19]).

Training Setup. For all methods, we use minibatch DP-SGD [1] in each silo to satisfy silo-specific item-level privacy. While certain methods may have more efficient solvers (e.g., dual form for [70]), we need compatibility with DP-SGD as well as privacy amplification via sampling on the example level for tight accounting. For all experiments, silos train for 1 local epoch in every round (except for [44] requiring ≥ 2 epochs). Hyperparameter tuning is kept minimal and is done via grid search for all methods. Importantly, we do not account for the *privacy cost* of tuning when comparing baselines to focus on their inherent utility under a given privacy budget; we revisit this issue in §7. For simplicity, we set the same privacy target for all silos, i.e. $(\varepsilon, \delta) = (\varepsilon_k, \delta_k)$ for all $k \in [K]$; this is without loss of generality since the effect of having varying DP noise scales from varying $(\varepsilon_k, \delta_k)$ is already attained by different local dataset sizes (under fixed batch size and # training rounds).

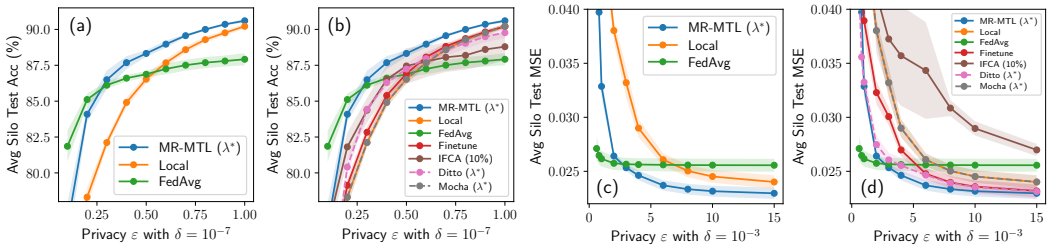

Figure 3: **Test metric** (acc and MSE, mean $\pm$ std w/ 5 seeds) vs **privacy budgets** ε for various personalization methods on the **Vehicle** dataset (*a, b*) and the **School** dataset (*c, d*) w/ 400 and 200 rounds respectively. Methods: Ditto [44], Mocha [70], IFCA [28, 53] (3 clusters), and local finetuning [80, 15]. Every silo targets for the same (ε, δ) under **silo-specific item-level DP**. λ^* denotes a good regularization strength in a grid where applicable. IFCA (10%) denotes forming clusters for only first 10% of training rounds due to privacy overhead (Fig. 4).

Results. Fig. 3 shows the privacy-utility tradeoff curves on Vehicle and School. We observe that MR-MTL consistently outperforms a suite of baseline methods (Fig. 3 (b, d)). Moreover, MR-MTL performs at least as good as local training and FedAvg (endpoints of the personalization spectrum), except at extremely small ε values (possibly different for each dataset). In particular, *there exists a range of ε values where MR-MTL can give significantly better utility over local training and FedAvg under the same privacy budgets* (Fig. 3 (a, c)); this is our key regime of interest.

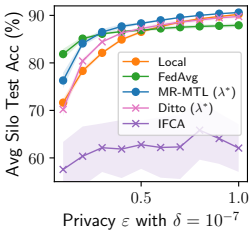

Figure 4: Effects of privacy overhead on Vehicle under silo-specific item-level DP.

Effects of Silo-specific Item-level Privacy. In Fig. 2 we saw that local finetuning may not improve utility as expected under silo-specific item-level DP, motivating us to reconsider the roles of federation and personalization (§5). In Fig. 4, we examine the effects of privacy overhead due to additional dataset queries: (1) If IFCA [28, 53] performs cluster selection at every round (default behavior), then the extra privacy cost can be prohibitive; (2) Despite its similarity to MR-MTL, Ditto [44]'s privacy overhead makes it less competitive (see also Fig. 5).

5 On the Effectiveness of Mean-Regularized MTL for Private Cross-Silo FL

Motivated by our observations in §4, we now examine the desirable properties of a good algorithm under silo-specific item-level privacy and pinpoint why MR-MTL can be a compelling candidate.

Federation as Noise Reduction. A key message from Fig. 2 is that in a setting where local training outperforms FedAvg without privacy, the converse can be true when privacy is added. In particular, the *utility cost* from DP is significantly smaller under FedAvg, even though local training spends an identical privacy budget. This points to the hypothesis that FedAvg has inherent benefits for DP noise reduction, despite the noises being added in the *gradient* space rather than the *parameter* space (as in client-level DP). Indeed, in the simplest setting of DP full-batch gradient descent, the update rule $w_k^{(t+1)} = w_k^{(t)} - \eta \left(z^{(t)} + {}^1\!/_{n_k} \sum_{i=1}^{n_k} g_{k,i}\right)$ for silo k recursively expands to $w_k^{(T)} = w_k^{(0)} - \eta \sum_{t=0}^{T} z^{(t)} - {}^\eta\!/_{n_k} \sum_{t=0}^{T} \sum_{i=1}^{n_k} g_{k,i}$ over T steps, where $g_{k,i}$ is the i-th clipped gradient (with norm bound c) of a total of n_k examples, and $z^{(t)} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ is the i.i.d. Gaussian noise for a total budget of $(\varepsilon_k, \delta_k)$ over T steps, with $\sigma^2 = O\left(c^2 T \ln(1/\delta_k)/\varepsilon_k^2\right)$ [1]. The cumulative noise term $-\eta \sum_t z^{(t)} \sim \mathcal{N}\left(0,\ O\left(\eta^2 c^2 T^2 \ln(1/\delta_k)/\varepsilon_k^2\right)\right)$ implies that each silo's model update has an independent Gaussian random walk component [68, 5, 76] whose variance can be reduced by averaging with other silos' updates, as in FedAvg.[3]

The Role of Model Personalization. The downside of FedAvg is that it can underperform simple local training due to *data heterogeneity*, and under cross-silo FL it does so more easily since clients often have enough data to fit a reasonable local model. This suggests a subtle shift in the role of *model personalization*: previous work primarily considers personalization for better utility, robustness, or fairness [75, 80, 44] under heterogeneity, but in our setting it can also serve to trade-off utility costs

[3] The work of [39] examines the benefits of adding correlated (instead of independent) noises z_t across time steps. While this is a potential orthogonal extension to our use of local DP-SGD *within* each silo, it may not be applicable to our main focus of reducing noise variance *across* silos, since for each silo k to satisfy its own $(\varepsilon_k, \delta_k)$ requirement, it must add noise independent to other silos.

from *privacy*, since full personalization (local training) and zero personalization (FedAvg) now form two ends of a spectrum with identical privacy guarantees that minimize the effect of heterogeneity and privacy noise respectively. It is then natural to ask whether a midpoint on this personalization spectrum may yield better utility. In particular, our empirical observations motivate the following key properties for a candidate personalization algorithm:

1. **Noise reduction**: The effect of noise reduction is present throughout training so that the utility costs from DP do not "rebound". Local finetuning is one such counter-example (Fig. 2).
2. **Minimal privacy overhead**: There are little to no additional training iterations or queries on the local datasets to prevent extra noise for DP-SGD under a fixed privacy budget. In effect, such privacy overhead can shift the utility tradeoff curve downwards (Figs. 4 and 5).
3. **Smooth interpolation**: The interpolation should be fine-grained (if not continuous), such that if an optimal tradeoff exists it is ideally attainable in most privacy settings.

These properties are rather restrictive. For example, model mixture [46, 10, 19] and local adaptation [80, 15] methods can incur linear overhead in iterations over the local datasets. Similarly, multi-task learning [44, 70, 69] methods benefit from additional training iterations. Clustering methods [28, 17, 53, 69] can also incur overhead with cluster selection [28, 53], distillation [17], or training restarts [69], and they discretize the personalization spectrum in a way that depends on external parameters (e.g., the number of clients, clusters, or top-down partitions).

The Promises of Mean-Regularization. These considerations point to *mean-regularized multi-task learning* (MR-MTL) as one of the simplest yet particularly suitable forms of personalization. MR-MTL has manifested in various forms in the literature [25, 81, 71, 33, 15] with the key idea that a personalized model w_k for each silo k should be close to the mean of all personalized models $\bar{w}$ via a regularization penalty $\lambda/2\|w_k - \bar{w}\|_2^2$. The hyperparameter λ serves as a smooth knob between local training and FedAvg, with $\lambda = 0$ recovering local training and a larger λ forces the personalized models w_k to be closer to each other ("federate more"). It is however an imperfect knob as $\lambda \to \infty$ may *not* recover FedAvg under a typical optimization setup (i.e. local SGD with mean-regularization panelty), and may thus underperform FedAvg in high privacy settings requiring a large λ (Fig. 3).

MR-MTL contains our desired properties: (1) noise reduction is achieved throughout training via a soft constraint that personalized models are close to an averaged model; (2) for fixed λ it has *zero* privacy overhead compared to local training/FedAvg as it does not involve extra queries; and (3) λ provides a smooth interpolation along the personalization spectrum. Moreover, compared to other regularization-based MTL methods, it adds only one hyperparameter λ (cf. [36, 85, 31]; this has important practical implications as will be discussed in §7), has fast convergence [82], empirically works well in the primal (thus easily extends to deep learning; cf. [8, 70, 49]), and is sufficently extensible to handle structured heterogeneity (discussed below). As we will argue through the following empirical analyses, these properties make MR-MTL a strong baseline in our privacy setting.

Navigating the Privacy-Heterogeneity Tradeoff. In Fig. 5 we study the effect of the regularization strength λ on the model utility directly. There are several interesting observations: (1) In both private and non-private settings, λ serves to interpolate between local training and FedAvg. (2) The utility at the best λ^* may outperform both endpoints. This is significant for the private setting since MR-MTL achieves an *identical* privacy guarantee as the endpoints. (3) Moreover, the *advantage* of MR-MTL over the best of the endpoints are also larger under privacy. (4) The value of λ^* also increases under privacy, indicating that the personalized silo models are closer to each other (i.e. silos are encouraged to "federate more") for noise reduction. We will characterize these behaviors in §6. Ditto [44], a state-of-the-art personalization method that is algorithmically similar to MR-MTL, is used to illustrate the effect of privacy overhead.

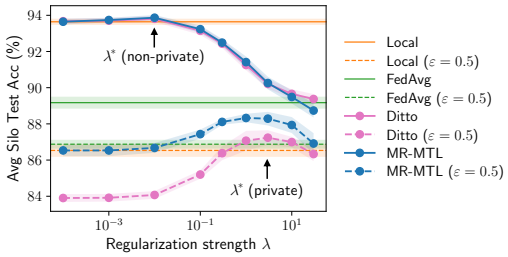

Figure 5: **Test acc $\pm$ std of MR-MTL with varying** λ (corresponding to $\varepsilon = 0.5$ in Fig. 3 (a)). Optimal points (λ^*) exist where it outperforms both ends of the spectrum under the *same* privacy. Ditto [44] gives a similar interpolation but has strictly worse privacy-utility tradeoffs due to its privacy overhead.

MR-MTL under Structured Heterogeneity. We further study (1) the extensibility of MR-MTL to handle clustering structures of silo data distributions and (2) its flexibility to handle varying heterogeneity levels by manually introducing two layers of heterogeneity to the MNIST dataset [41].

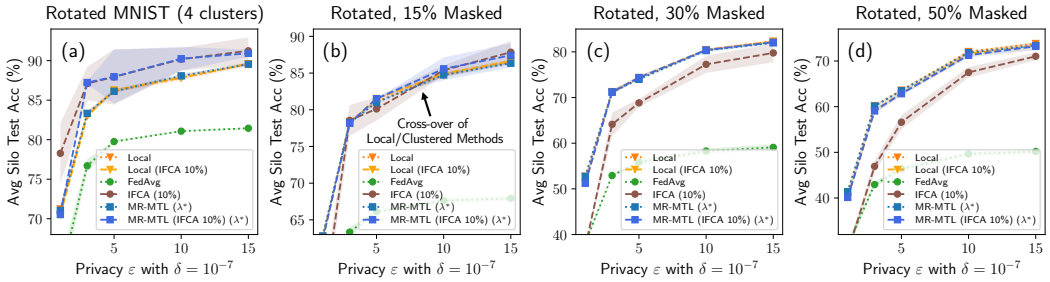

Figure 6: **Test acc $\pm$ std on Rotated + Masked MNIST**. (IFCA 10%) denotes warm-starting the method by running IFCA [28, 53] for first 10% of rounds followed by running the method within the cluster structures.

The first layer is 4-way rotations: train/test images are evenly split into 4 groups of 10 silos, with each group applying $\{0°, 90°, 180°, 270°\}$ of rotation to their images. The second layer is *silo-specific* masking: each silo generates and applies its unique random mask of 2×2 white patches to its images, with varying masking probability. Together, the 1st layer creates 4 well-defined silo clusters, and the 2nd layer (gradually) adds *intra-cluster* heterogeneity (see examples in appendix). Importantly, our goal is *not* to contrive a utility advantage of MR-MTL (in fact, the added heterogeneity is disadvantageous to mean-regularization), but to examine its extensibility and flexibility to match the best private baselines *by construction*. In the 1st layer, clustered FL methods [28, 53] should be optimal if the correct clusters are formed since there is no intra-cluster heterogeneity; with increasing silo-specific heterogeneity in the 2nd layer, local training should be increasingly more attractive.

We propose a simple heuristic to "warm-start" MR-MTL with a small number of training rounds by running private clustering (with IFCA [28, 53]) followed by mean-regularized training *within each formed cluster*. This simple heuristic, with convergence properties carried over from its components [28, 25, 82], enables MR-MTL to excel at all levels of heterogeneity: in Fig. 6 (a), the warm-start allows MR-MTL to match IFCA (optimal by construction) while local training (full personalization) does not benefit from the same heuristic; in Fig. 6 (b, c, d), MR-MTL remains optimal across different levels of silo-specific heterogeneity (2nd layer) while the gains from warm-start gradually drop. We argue that extensibility and flexibility are good properties that make MR-MTL a strong baseline, as heterogeneity in practical settings is likely less adversarial than what we presented.

6 Analysis

In this section we provide a theoretical analysis of MR-MTL under mean estimation as a simplified proxy for (single-round) FL, with a Bayesian framework extending on [44]. We provide expressions for the Bayes optimal estimator MR-MTL (λ^*), and describe how MR-MTL behaves with varying λ to characterize our observations from Fig. 5. Proofs and extensions are deferred to Appendix E.

We start with a total of K silos where the k-th silo holds n training points $X_k \triangleq \{x_{k,i} \in \mathbb{R}\}_{i \in [n]}$, each normally distributed around an unknown center w_k with variance σ^2; i.e. $x_{k,i} = w_k + z_{k,i}$ with $z_{k,i} \sim \mathcal{N}(0, \sigma^2)$. To quantify heterogeneity, the silo centers are also normally distributed around some unknown fixed meta-center θ with variance τ^2; i.e. $w_k = \theta + z$ with $z \sim \mathcal{N}(0, \tau^2)$. A large τ means that w_k are distant from each other and thus their local objectives are heterogeneous, and contrarily for a small τ. Our goal is for each silo k to compute a item-level private estimate of w_k that minimizes the *generalization* error (i.e. on unseen points from the same local distribution). Each silo targets (ε, δ) item-level DP and runs the Gaussian mechanism with noise scale $\sigma_{\text{DP}} = c\sqrt{2\ln(1.25/\delta)}/\varepsilon$ and clipping bound c.[4] Under this setting, the MR-MTL objective for the k-th silo is

$$h_k(w) = \tilde{F}_k(w) + \frac{\lambda}{2}\|w - \bar{w}\|_2^2. \tag{2}$$

Here, $\tilde{F}_k(w) \triangleq \frac{1}{2}(w - \frac{1}{n}(\xi_k + \sum_{i=1}^n x_{k,i} \cdot \min(1, c/\|x_{k,i}\|_2)))^2$ is the local objective to privately estimate the mean of the local data points with privacy noise $\xi_k \sim \mathcal{N}(0, \sigma_{\text{DP}}^2)$. Since the data are subgaussian, we assume one can choose c such that w.h.p. no clipping error is introduced, so $\hat{w}_k \triangleq \operatorname{argmin} F_k(w) = \frac{1}{n}(\xi_k + \sum_i x_{k,i})$ is the best local estimator. $\bar{w} = \frac{1}{K}\sum_k \hat{w}_k$ is the average estimator across silos, which is the same as the FedAvg estimator under mean estimation. We also

[4] For simplicity we start with the same n, σ, σ_{DP} for all silos; one may also extend to silo-specific values.

consider the *external* average local estimators for silo k, defined as $\hat{w}_{\setminus k} \triangleq \frac{1}{K-1}\sum_{j\neq k}\hat{w}_j$. The following lemma gives the best MR-MTL estimator $\hat{w}_k(\lambda)$ as a function of λ.

Lemma 6.1. *Let $\lambda \geq 0$ and $\alpha = \frac{K+\lambda}{(1+\lambda)K} \in (1/K, 1]$. The minimizer of $h_k(w)$ is given by*

$$\hat{w}_k(\lambda) = \alpha \cdot \hat{w}_k + (1-\alpha)\cdot \hat{w}_{\setminus k}. \tag{3}$$

Importantly, note that the best λ is always 0 for *training* error (i.e. estimating the mean of the available data $\{x_{k,i}\}$); our hope is that with some $\lambda > 0$, $\hat{w}_k(\lambda)$ yields a better *generalization* error.

We now present the main takeaways. At a high level, the basis of our analysis relies on expressing the true center w_k in terms of $\hat{w}_k$ and $\hat{w}_{\setminus k}$ conditioned on the local datasets $\{X_k\}_{k\in[K]}$. Let $\sigma^2_{\text{loc}} \triangleq \sigma^2/n + \sigma^2_{\text{DP}}/n^2$ denote the "local variance" of $\hat{w}_k$ around w_k due to both data sampling and privacy noise. Using a technical lemma from [52] (Lemma 11) we have the following lemma.

Lemma 6.2. *Given $\hat{w}_k$, $\hat{w}_{\setminus k}$, and $\{X_k\}_{k\in[K]}$, we can express $w_k = \mu_k + \zeta_k$, where $\zeta_k \sim \mathcal{N}(0, \sigma^2_w)$,*

$$\sigma^2_w \triangleq \left(\frac{1}{\sigma^2_{\text{loc}}} + \frac{K-1}{K\tau^2 + \sigma^2_{\text{loc}}}\right)^{-1} \quad \textit{and} \quad \mu_k \triangleq \sigma^2_w\left(\frac{1}{\sigma^2_{\text{loc}}}\cdot\hat{w}_k + \frac{K-1}{K\tau^2+\sigma^2_{\text{loc}}}\cdot\hat{w}_{\setminus k}\right). \tag{4}$$

Lemma 6.2 expresses the unobserved true client centers w_k in terms of the (private) empirical estimators $\hat{w}_k$ and $\hat{w}_{\setminus k}$. This expression requires conditioning on the datasets X_k as they form the Markov blankets of $\hat{w}_k$. Combining Lemma 6.1 and Lemma 6.2 gives the optimal λ.

Theorem 6.3 (Optimal MR-MTL estimate)**.** *The best λ^* for the generalization error is given by*

$$\lambda^* = \operatorname*{argmin}_{\lambda} \mathbb{E}\left[(w_k - \hat{w}_k(\lambda))^2 \mid \hat{w}_k, \hat{w}_{\setminus k}, \{X_k\}_{k\in[K]}\right] = \frac{1}{n\tau^2}\left(\sigma^2 + \frac{\sigma^2_{\text{DP}}}{n}\right). \tag{5}$$

Theorem 6.3 suggests that there indeed exists an optimal point $\hat{w}(\lambda^*)$ on the personalization spectrum. Moreover, λ^* grows smoothly with stronger privacy ($\sigma^2_{\text{DP}} \to \infty$) to encourage silos to "federate more" with others. This was empirically observed in Fig. 5. We now characterize the utility of $\hat{w}(\lambda^*)$.

Corollary 6.4 (Optimal MSE with $\hat{w}(\lambda^*)$)**.** *The error of the estimator $\hat{w}(\lambda^*)$ is given by its variance. That is, $\mathcal{E}^* \triangleq \mathbb{E}\left[(w_k - \hat{w}_k(\lambda^*))^2 \mid \hat{w}_k, \hat{w}_{\setminus k}, \{X_k\}_{k\in[K]}\right] = \sigma^2_w = \frac{\sigma^2_{\text{loc}}(\sigma^2_{\text{loc}}+K\tau^2)}{K(\sigma^2_{\text{loc}}+\tau^2)}$.*

Note also that $\hat{w}(\lambda^*)$ is the MMSE estimator of w_k. Corollary 6.4 allows us to compare the best MR-MTL solver $\hat{w}_k(\lambda^*)$ against the endpoints of the personalization spectrum (local and FedAvg).

Proposition 6.5 (Optimal error gap to local training)**.** *Let $\mathcal{E}_{\text{loc}} \triangleq \mathbb{E}\left[(w_k - \hat{w}_k)^2 \mid X_k\right] = \sigma^2_{\text{loc}}$ be the error of the local estimate. Then, compared to the optimal estimator (Corollary 6.4), the local estimator incurs $\Delta_{\text{loc}} \triangleq \mathcal{E}_{\text{loc}} - \mathcal{E}^* = \left(1 - \frac{1}{K}\right)\cdot\frac{\sigma^4_{\text{loc}}}{\sigma^2_{\text{loc}}+\tau^2}$ additional error.*

Proposition 6.6 (Optimal error gap to FedAvg)**.** *Let $\mathcal{E}_{\text{fed}} \triangleq \mathbb{E}\left[(w_k - \bar{w})^2 \mid \{X_k\}_{k\in[K]}\right]$ be the error under FedAvg. Then, compared to the optimal estimator (Corollary 6.4), the FedAvg estimator incurs $\Delta_{\text{fed}} \triangleq \mathcal{E}_{\text{fed}} - \mathcal{E}^* = \left(1 - \frac{1}{K}\right)\cdot\frac{\tau^4}{\sigma^2_{\text{loc}}+\tau^2}$ additional error.*

Together, Propositions 6.5 and 6.6 suggest that the effects of stronger privacy ($\sigma^2_{\text{DP}} \to \infty$) on how MR-MTL compares against the personalization endpoints are opposite, with the benefit of MR-MTL *increasing* against local training while *diminishing* against FedAvg. The intuition is that the utility under stronger privacy should benefit from "more federation" as a means to cancel out the per-silo DP noise (characterized below in Theorem 6.8), which makes the MR-MTL estimator $\hat{w}_k(\lambda)$ closer to the FedAvg estimator $\bar{w}$. This also suggests that MR-MTL has an optimal utility advantage over both local and FedAvg when local performs on par with FedAvg with $\sigma^2_{\text{loc}} \approx \tau^2$, and the utility "bump" under privacy in Fig. 5 can be viewed as a result of this balance. Intriguingly, since data variance σ^2 and heterogeneity τ^2 are often fixed in practice, the freedom for silos to vary σ^2_{DP} (ε) makes the utility advantage of MR-MTL more flexible compared to non-private settings.

Behavior of MR-MTL as a function of λ. The above captures how MR-MTL behaves at its optimum, but in Fig. 5 we also observed that MR-MTL has the desirable property that the utility cost from DP *shrinks smoothly* with larger λ (§5). Lemma 6.7 and Theorem 6.8 below can help us understand why.

Lemma 6.7 (Error of $\hat{w}_k(\lambda)$)**.** *Let $\mathcal{E}(\lambda) \triangleq \mathbb{E}\left[(w_k - \hat{w}_k(\lambda))^2 \mid \hat{w}_k, \hat{w}_{\setminus k}, \{X_k\}_{k\in[K]}\right]$ be the error of MR-MTL as a function of λ. Then, $\mathcal{E}(\lambda) = \left(1 - \frac{1}{K}\right)\frac{\sigma^2_{\text{loc}}+\lambda^2\tau^2}{(\lambda+1)^2} + \frac{\sigma^2_{\text{loc}}}{K}$.*

Using Lemma 6.7 we can now characterize how λ affects the utility cost from DP (recall from Figs. 2 and 5 that federation helps with noise reduction). As a side note, Lemma 6.7 suggests that MR-MTL's utility as a function of λ would have a quasi-concave shape, as was empirically observed in Fig. 5. This could potentially help make heuristic or automated search over λ easier.

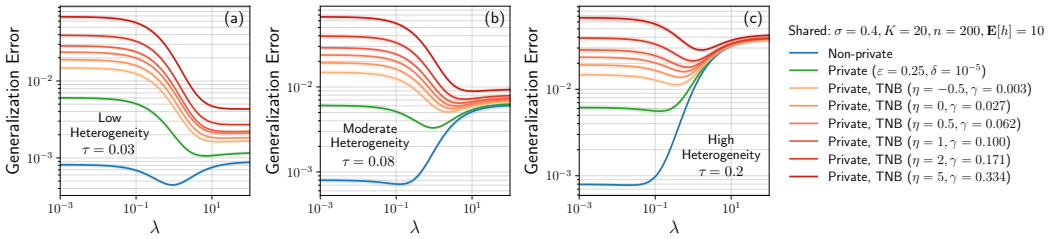

Figure 7: **Privacy costs of tuning λ on mean estimation** (setup follows §6). **Private** and **Non-Private** denote the errors of varying λ with or without silo-specific item-level DP, and the privacy cost of tuning λ is *not* included. **Private, TNB** denotes the same errors but accounts for the privacy cost of trying $\mathbf{E}[h] = 10$ values of λ, with h sampled from the truncated negative binomial distribution with parameters η, γ arriving at the same $\mathbf{E}[h]$ [63]. To interpret, observe that the lowest points of **Private, TNB** may be still higher than an endpoint of **Private**.

Theorem 6.8 (Private utility gap)**.** *Let $\hat{w}_k(\lambda)$ and $\hat{w}_k^{\mathrm{DP}}(\lambda)$ be the non-private and private estimate of w_k with $\sigma_{\mathrm{loc}}^2 \leftarrow \sigma^2/n$ and $\sigma_{\mathrm{loc}}^2 \leftarrow \sigma^2/n + \sigma_{\mathrm{DP}}^2/n^2$, respectively. Let $\mathcal{E}(\lambda)$ and $\mathcal{E}^{\mathrm{DP}}(\lambda)$ be the error of $\hat{w}_k(\lambda)$ and $\hat{w}_k^{\mathrm{DP}}(\lambda)$ respectively as in Lemma 6.7. Let $\Delta_{\mathrm{DP}}(\lambda) \triangleq \mathcal{E}^{\mathrm{DP}}(\lambda) - \mathcal{E}(\lambda)$ be the utility cost due to privacy as a function of λ. Then, $\Delta_{\mathrm{DP}}(\lambda) = \left(1 - \frac{1}{K}\right) \frac{1}{(\lambda+1)^2} \frac{\sigma_{\mathrm{DP}}^2}{n^2} + \frac{\sigma_{\mathrm{DP}}^2}{Kn^2}$.*

Theorem 6.8 suggests that with larger λ, the utility cost from silo-specific item-level privacy can be mitigated (by up to a factor of K); this was indeed empirically observed in Fig. 5.

7 Discussion

In previous sections, we studied the benefits of the best personalization hyperparameter λ^* for MR-MTL, but it remains open as to how to find such λ^*. In this section, we take an honest look at the complications of deploying MR-MTL through the lens of the *privacy cost* of obtaining λ^*. There are in general several approaches: (1) a non-adaptive search (e.g. grid/random search [9]); (2) an adaptive search (e.g. grad student descent); or (3) an online estimation during training (e.g. as in [74, 6, 64]). We focus on (1) since it is generic to all personalization methods and is a setting for which we have the best privacy accounting tools [47, 63] to our knowledge. We defer technical details and discussions for (3) to the appendix. Note that while we focus on MR-MTL, our reasoning should extend to all personalization methods whose advantage depends on having the best hyperparameter(s).

Recall that for a typical tuning procedure, a baseline algorithm M is executed h times with different hyperparameters and the best result is recorded. The work of [47, 63] shows that, with a constant h, there exists M that satisfies $(\varepsilon, \delta = 0)$-DP where the output of tuning is not $(\tilde{\varepsilon}, 0)$-DP for any $\tilde{\varepsilon} < h\varepsilon$, with similar results for Rényi DP (thus also for $\delta > 0$). This unfortunately implies that naive tuning (as done in practice) can nullify the gain of MR-MTL (λ^*) over local/FedAvg. Fortunately, by making h *random*, we can make $\tilde{\varepsilon}$ *constant* w.r.t. h [47, 63] or at most $\tilde{\varepsilon} \leq O(\log \mathbf{E}[h])$ [63]. However, using the simplified setting of mean estimation (§6), we find that even with this improved randomized protocol, there exist scenarios (Fig. 7) where the realistic cost of trying a moderate $\mathbf{E}[h] = 10$ values of λ may significantly diminish, or even *outweigh*, the utility advantage of λ^* over local training and FedAvg—that is, we might be better off by not privately tuning λ *at all*.

The above has several important implications. On the negative side, it suggests that the true efficacy of MR-MTL can be smaller in practice. Moreover, it raises the broader open question of whether the emerging privacy-heterogeneity tradeoff is best balanced by model personalization, as many existing methods including MR-MTL inherently require at least one hyperparameter to specify "how much" to personalize for general utility improvements over local training and FedAvg. Alternatively, the hyperparameter(s) may be estimated *during* training (approach (3) above), though such procedures may not be general and scalable and need to be tailored to the specific personalization method. On the positive side, it is unclear whether the choice of λ can meaningfully leak privacy in practice. MR-MTL may also be viewed favorably as a strong baseline since it only needs one hyperparameter to attain its benefits, while other methods requiring more tuning will incur even larger privacy costs.

Concluding remarks. In this work, we revisit the application of differential privacy in cross-silo FL. Our work points out several meaningful ways in which private cross-silo FL differs from the cross-device setting, particularly when analyzing tensions between privacy, utility, and heterogeneity. In doing so, we explore and establish baselines for private cross-silo FL—identifying both simple, promising methods (e.g., MR-MTL) for this setting as well as key open questions for the area at large.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Alekh Agarwal, John Langford, and Chen-Yu Wei. Federated residual learning. *arXiv preprint arXiv:2003.12880*, 2020.

[3] Naman Agarwal, Peter Kairouz, and Ziyu Liu. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Nasser Aldaghri, Hessam Mahdavifar, and Ahmad Beirami. Feo2: Federated learning with opt-out differential privacy. In *NeurIPS 2021 Workshop on New Frontiers in Federated Learning: Privacy, Fairness, Robustness, Personalization and Data Ownership*, 2021.

[5] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

[6] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34, 2021.

[7] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

[8] Aviad Barzilai and Koby Crammer. Convex multi-task learning by clustering. In *Artificial Intelligence and Statistics*, pages 65–73. PMLR, 2015.

[9] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[10] Alberto Bietti, Chen-Yu Wei, Miroslav Dudik, John Langford, and Zhiwei Steven Wu. Personalization improves privacy-accuracy tradeoffs in federated optimization. *arXiv preprint arXiv:2202.05318*, 2022.

[11] Kallista Bonawitz, Peter Kairouz, Brendan Mcmahan, and Daniel Ramage. Federated learning and privacy. *Commun. ACM*, 65(4):90–97, mar 2022.

[12] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

[13] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[14] Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020.

[15] Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 2021.

[16] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 375–403. Springer, 2019.

[17] Yae Jee Cho, Jianyu Wang, Tarun Chiruvolu, and Gauri Joshi. Personalized federated learning for heterogeneous clients with clustered knowledge transfer. *arXiv preprint arXiv:2109.08119*, 2021.

[18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[19] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[20] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2022.

[21] Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

[22] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[23] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[24] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2468–2479. SIAM, 2019.

[25] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

[26] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[27] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[28] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

[29] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2521–2529. PMLR, 2021.

[30] Harvey Goldstein. Multilevel modelling of survey data. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 40(2):235–244, 1991.

[31] Pinghua Gong, Jieping Ye, and Changshui Zhang. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903, 2012.

[32] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315, 2020.

[33] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[34] Mikko A Heikkilä, Antti Koskela, Kana Shimizu, Samuel Kaski, and Antti Honkela. Differentially private cross-silo federated learning. *arXiv preprint arXiv:2007.05553*, 2020.

[35] Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Private multi-task learning: Formulation and applications to federated learning. *arXiv preprint arXiv:2108.12978*, 2021.

[36] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep Ravikumar. A dirty model for multi-task learning. *Advances in neural information processing systems*, 23, 2010.

[37] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[38] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pages 5201–5212. PMLR, 2021.

[39] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning*, pages 5213–5225. PMLR, 2021.

[40] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[42] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. Learning with user-level privacy. *Advances in Neural Information Processing Systems*, 34, 2021.

[43] Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. Differentially private meta-learning. In *International Conference on Learning Representations*, 2020.

[44] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[45] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[46] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[47] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

[48] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. Projected federated averaging with heterogeneous differential privacy. In *International Conference on Very Large Databases*. VLDB Endowment, 2022.

[49] Sulin Liu, Sinno Jialin Pan, and Qirong Ho. Distributed multi-task relationship learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946, 2017.

[50] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[51] Linpeng Lu and Ning Ding. Multi-party private set intersection in vertical federated learning. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 707–714. IEEE, 2020.

[52] Hessam Mahdavifar, Ahmad Beirami, Behrouz Touri, and Jeff S Shamma. Global games with noisy information sharing. *IEEE Transactions on Signal and Information Processing over Networks*, 4(3):497–509, 2017.

[53] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

[54] Ryan McKenna and Daniel R Sheldon. Permute-and-flip: A new mechanism for differentially private selection. *Advances in Neural Information Processing Systems*, 33:193–203, 2020.

[55] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[56] H Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018.

[57] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[58] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[59] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[60] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[61] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

[62] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[63] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2022.

[64] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.

[65] Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

[66] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. Technical report, OpenAI, 2022.

[67] Ryan Rogers and Thomas Steinke. A better privacy analysis of the exponential mechanism. DifferentialPrivacy.org, 07 2021. https://differentialprivacy.org/exponential-mechanism-bounded-range/.

[68] Thorsteinn Rögnvaldsson. On langevin updating in multilayer perceptrons. *Neural computation*, 6(5):916–926, 1994.

[69] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[70] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.

[71] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

[72] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11, 2019.

[73] Akhil Vaid, Suraj K Jaladanki, Jie Xu, Shelly Teng, Arvind Kumar, Samuel Lee, Sulaiman Somani, Ishan Paranjpe, Jessica K De Freitas, Tingyi Wanyan, Kipp W Johnson, Mesude Bicak, Eyal Klang, Young Joon Kwon, Anthony Costa, Shan Zhao, Riccardo Miotto, Alexander W Charney, Erwin Böttinger, Zahi A Fayad, Girish N Nadkarni, Fei Wang, and Benjamin S Glicksberg. Federated learning of electronic health records improves mortality prediction in patients hospitalized with covid-19. *medRxiv*, 2020.

[74] Koen Lennart van der Veen, Ruben Seggers, Peter Bloem, and Giorgio Patrini. Three tools for practical differential privacy. In *Privacy Preserving Machine Learning (PPML) Workshop at NeurIPS 2018*, 2018.

[75] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

[76] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[77] Yuxin Wen, Jonas Geiping, Liam Fowl, Micah Goldblum, and Tom Goldstein. Fishing for user data in large-batch federated learning via gradient magnification. *arXiv preprint arXiv:2202.00580*, 2022.

[78] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[79] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 332–349. IEEE, 2019.

[80] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

[81] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. *Advances in neural information processing systems*, 28, 2015.

[82] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[83] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[84] Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie Su. Federated f-differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2251–2259. PMLR, 2021.

[85] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. *Advances in neural information processing systems*, 24, 2011.

[86] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 21:1–50, 2011.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7 as well as the appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Discussions are provided in the appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 6. We also included additional details in the appendix.
   (b) Did you include complete proofs of all theoretical results? [Yes] Complete proofs are included in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code will be attached as part of the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details are provided in the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Most experiments are repeated at least 5 times and Fig. 6 is repeated at least 3 times where computing resources permit.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details are provided in the appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Details are provided in the appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code will be attached as part of the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]