On Privacy and Personalization in Cross-Silo Federated Learning

Ziyu Liu **Shengyuan Hu** **Zhiwei Steven Wu** **Virginia Smith**
Carnegie Mellon University
{kzliu, shengyuanhu, zstevenwu, smithv}@cmu.edu

Abstract

While the application of differential privacy (DP) has been well-studied in cross-device federated learning (FL), there is a lack of work considering DP and its implications 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* DP are less suitable as real-world privacy regulations typically concern the in-silo data subjects rather than the silos themselves. In this work, we instead consider an alternative notion of *silo-specific sample-level* DP, where silos set their own privacy targets for their local examples. Under this setting, we reconsider the roles of personalization in federated learning. In particular, we show that mean-regularized multi-task learning (MR-MTL), a simple personalization framework, is a strong baseline for cross-silo FL: under stronger privacy requirements, silos are incentivized to federate more with each other to mitigate DP noise, resulting in consistent improvements relative to standard baseline methods. We provide an empirical study of competing methods as well as a theoretical characterization of MR-MTL for mean estimation, 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 of future work in this area.

1 Introduction

Recent advances in machine learning often rely on large, centralized datasets [84, 23, 65], 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) [71, 50] 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 [101, 102]. To obtain formal privacy guarantees, there has thus been extensive work applying *differential privacy* (DP) [27, 28] to various parts of the FL pipeline (e.g. [32, 73, 41, 48, 4, 63, 44, 83, 34]).

Existing approaches for differentially private FL 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"), and DP is achieved by clipping and noising the client model updates. While client-level DP is considered a strong privacy notion as all data of a single client is protected, 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 [96], compromising the privacy of the federating clients.

In this work, we instead consider a more natural model of *silo-specific sample-level privacy* (Fig. 1, with variants appearing in [41, 66, 109, 51]): the k-th silo may set its own $(\varepsilon_k, \delta_k)$ sample-level DP target for any learning algorithm with respect to its local dataset. With this formulation in mind, we then reconsider the impact of privacy, heterogeneity, and personalization in cross-silo FL. In particular,

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

we explore existing baselines for FL (mostly developed in cross-device settings) across private cross-silo benchmarks, and 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 practice. In addition to establishing baselines for cross-silo FL, we also identify interesting future directions in this area (§7 and Appendix G). We summarize our contributions below:[1]

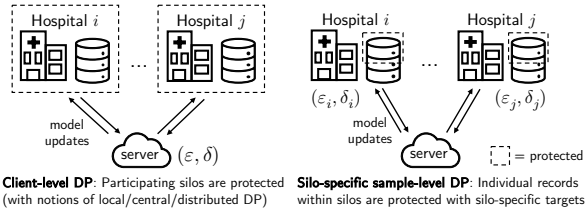

Figure 1: **Client-level** DP vs **Silo-specific sample-level** DP.

- We consider the notion of *silo-specific sample-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 sample-level DP. Core to its effectiveness is its ability to (roughly) interpolate on the model personalization spectrum between local training and FedAvg with minimal 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 examine the complications of deploying an optimal MR-MTL instance that stem 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 sample-level DP.

2 Preliminaries

Federated Learning (FL) [71, 58, 50] 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 [71], 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 ability to mitigate systemic privacy risks, FL by itself does not provide formal privacy guarantees for participants' data [50, 102, 101], 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 [27, 28])**.** A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$, where $\mathcal{X}^n$ is the set of datasets with n samples 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' differing in only one sample (by 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 [28], which involves bounding the contribution (ℓ^2-norm) of each sample in the dataset 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 [92, 11, 1] for ensuring sample-level DP for model training, and we use Rényi DP [77] and zCDP [15] for tight privacy composition. In certain FL algorithms, clients also perform additional

[1] Code is available at `https://github.com/kenziyuliu/private-cross-silo-fl`.

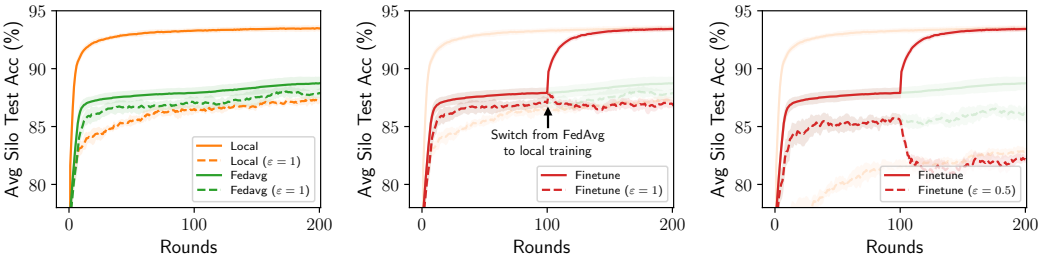

Figure 2: **Two notable phenomena under *silo-specific sample-level DP***: (1) FedAvg can serve to cancel out per-silo DP noise and thus outperform local training even when the latter works better without privacy (left); (2) Local finetuning [104, 19] (FedAvg followed by local training) may not improve utility as expected, as the effect of noise reduction is removed when finetuning begins (mid & right). Results report mean test acc $\pm$ std on the Vehicle dataset over 5 seeds. For simplicity, all silos budgets for the same labeled ε with $\delta = 10^{-7}$. Transparent curves refer to local/FedAvg runs with the same ε labeled for finetuning (compare left & mid).

work such as cluster selection [69, 33] 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.[2] Past work has examined the roles of local adaptation [99, 104, 19], multi-task learning [91, 89], clustering [33, 22, 69, 89], public data [108, 69], meta learning [46, 56, 31], or other forms of model mixtures [61, 57, 69, 38, 24, 3]. Notably, many methods leverage extra computation to some extent (e.g. extra iterations [61, 57, 19] or cluster selection [33, 69]), which will result in privacy overhead under silo-specific sample-level DP as discussed in the following section. Of particular interest is the family of mean-regularized multi-task learning (MR-MTL) methods [30, 94, 38, 37] (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 participating clients, i.e. client-level DP. For cross-silo FL, however, several factors render client-level DP less appropriate. First, cross-silo FL often involves a small number of clients and it can be utility-wise more costly to attain the same privacy targets. For example, privacy amplification via sampling [1, 78] may not apply on the client level since all silos typically participate in every round. Second, many existing methods focus on enforcing client-level DP in a non-local model and thus defines a shared privacy target for all participants, but in real-world cross-silo settings, participants under different jurisdictions (e.g. states) may have varying privacy requirements and thus opt for different privacy-utility tradeoffs. Third, while silo-level protection implies sample-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 sample-level* DP as an alternative privacy model (Fig. 1):

Definition 3.1 (Silo-specific sample-level DP)**.** A cross-silo FL algorithm with K clients (silos) satisfy $\{(\varepsilon_k, \delta_k)\}_{k\in[K]}$-"*silo-specific sample-level DP*" if the local (personalized) model M_k of every silo $k \in [K]$ satisfies $(\varepsilon_k, \delta_k)$-DP w.r.t. the silo's local dataset of training examples.

Characteristics of silo-specific sample-level privacy. Importantly, silo-specific sample-level DP is defined over the *disjoint* datasets of the individual silos, rather than the combined dataset of all silos.[3] To instantiate this setup in FL, each silo can simply run DP-SGD [92, 11, 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 privacy notion has several important implications on the dynamics of FL:

1. **Silos incur privacy costs with queries to their data, but *not* with participation in FL.** This follows from DP's robustness to post-processing: the silos' model updates in each round already satisfy their own sample-level DP targets, and participation by itself does not involve extra dataset

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

[3] A record in such a combined dataset is at most $(\max_i \varepsilon_i, \max_i \delta_i)$-DP [76, 103, 63]. 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 each entity has at most one record across the combined dataset (e.g. students attending exactly one school). See Appendix B for discussions.

queries (e.g. DP-SGD steps). In contrast, local training without communication can be kept noise-free under client-level DP, but participation in FL requires privatization. Two immediate consequences of the above are that (1) local training and FedAvg now have *identical* privacy costs, and (2) *local finetuning* for model personalization may no longer work as expected (Fig. 2).

2. **Less reliance on a trusted server.** As a corollary of the above, all model updates of silo k satisfy (at least) $(\varepsilon_k, \delta_k)$-DP against external adversaries, including all other silos and the orchestrating server [28, 109]. In contrast, client-level DP under a non-local model necessitates some trust on the server, even for distributed DP methods (e.g. [29, 21, 48, 4, 20, 18]).
3. **Tradeoff emerges between costs from privacy and heterogeneity.** As privacy-perserving noises are added independently on each silo, they are reflected in silos' model updates and can thus be mitigated when the model updates are aggregated (e.g. via FedAvg), leading to a smaller utility drop due to DP for the shared model. On the other hand, federation also means that the shared model may suffer from *client heterogeneity* (non-iid data across silos). This intuition is observed in Fig. 2: while local training may outperform FedAvg without privacy (as a result of heterogeneity), the opposite can be true when privacy is added (as a result of noise variance reduction).

The first and last in the above are of particular interest 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) can be viewed as two ends of a *personalization spectrum* with identical privacy costs; if local training minimizes the effect of data heterogeneity but enjoys no DP noise reduction, and contrarily for FedAvg, it is then natural to ask whether there exist personalization methods that lie in between and achieve better utility, and, if so, what methods would work best.

Related privacy settings. Past work on differentially private FL has concentrated on client-level DP and cross-device FL (e.g. [32, 74, 44, 34, 49, 48, 6]), and the application of DP in cross-silo FL, particularly where each silo defines its own DP targets for records of its own dataset, is relatively underexplored. Privacy notions closest to ours first appeared in [95, 56, 41, 109, 66, 51, 63]. In [95], each client adds its own one-shot noise onto its outgoing update, but in learning scenarios this provides client-level protection. The works of [56, 66, 109, 41, 63] study analogous privacy notions, though they respectively focus on boosting utility [56], analyzing statistical rates [66], adapting FL to f-DP [109, 25], applying security primitives [41], and learning a better global model; the aspects of heterogeneity, DP noise reduction, the personalization spectrum, and their interplay (e.g. Figs. 2 and 5) were unexplored. The work of [51] also considers a similar privacy notion, but the authors study a disparate trust assumption where DP noise is *not* added to local training/finetuning such that the final personalized models lack privacy guarantees. We note that the trust model most suitable for private cross-silo FL may be application-specific; in this work, we focus on the setting where the outputs of the FL procedure (the personalized models) must remain differentially private.

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 four cross-silo datasets that span regression/classification and convex/non-convex tasks: Vehicle [26], School [35], Google Glass (GLEAM) [82], and CIFAR-10 [53]. The first three datasets have real-world cross-silo characteristics: Vehicle contains measurements of road segments for classifying the type of passing vehicles, School contains student attributes for predicting exam scores, and GLEAM contains motion tracking data to classify wearers' activities. CIFAR-10 has heterogeneous client splits following [94, 90]. See Appendix C.1 for more details and datasets.

Benchmark methods. We consider several representative methods in the personalized FL literature beyond local training and FedAvg [71]: local finetuning [99, 104, 19] (a simple but strong baseline for model personalization), Ditto [57] (state-of-the-art personalization method), Mocha [91] (personalization with task relationship learning), IFCA/HypCluster [33, 69] (state-of-the-art hard clustering method for client models), and the mean-regularized multi-task learning (MR-MTL) methods [30, 94, 38, 37] (which we analyze in §5). For fair comparison under silo-specific sample-level DP, we align all benchmark methods on the total privacy budget by first restricting the total number of iterations over the local datasets and then account for any privacy overheads (in the form of necessary extra steps [57] or cluster selection for IFCA/HypCluster [33, 69]). Importantly, many other personalization methods can either be reduced to one of the above under convex settings (e.g. [46, 61]) or are unsuitable due to large privacy overheads (e.g. large factor of extra steps for [31]).

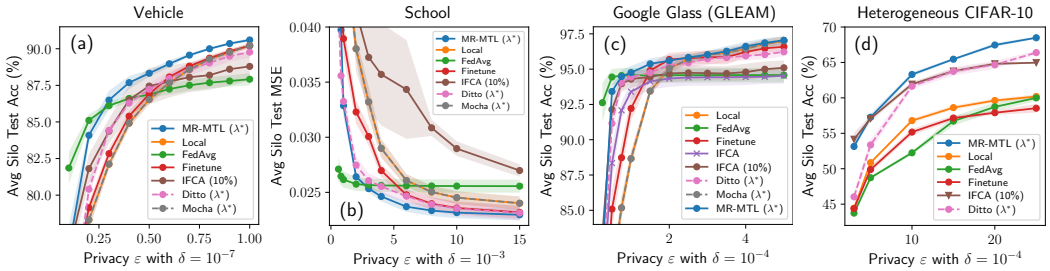

Figure 3: **Privacy-utility tradeoffs** (privacy budgets ε vs. test metrics, mean $\pm$ std w/ 5 seeds) for various personalization methods on **Vehicle**, **School**, **GLEAM**, and **Heterogeneous CIFAR-10** datasets respectively. For simplicity, every silo targets for the same (ε, δ) under silo-specific sample-level DP. λ^* denotes a tuned regularization strength where applicable. "Local" denotes local training (clients train and keep their own models). "IFCA (10%)" denotes forming clusters for only first 10% of training rounds due to privacy overhead (Fig. 4).

Training setup. For all methods, we use minibatch DP-SGD in each silo to satisfy silo-specific sample-level privacy; while certain methods may have more efficient solvers (e.g. dual form for [91]), we want 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 [57] which runs ≥ 2 epochs). Hyperparameter tuning is done via grid search for all methods. Importantly, when comparing the benchmark methods, we do not account for the privacy cost of hyperparameter tuning in order to focus on their inherent privacy-utility tradeoff; we revisit this issue in §7. For simplicity, we use the same privacy budget for all silos, i.e. $(\varepsilon_i, \delta_i) = (\varepsilon_j, \delta_j)$ for all $i, j \in [K]$; note that this is not a restrictive assumption since the effect of having varying DP noise scales from different budgets can be attained by varying local dataset sizes.

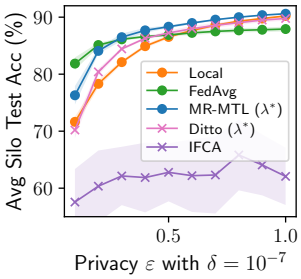

Figure 4: The privacy overhead of cluster selection (IFCA) and extra iterations (Ditto) compared to local, FedAvg, and MR-MTL under silo-specific sample-level DP.

Results. Fig. 3 shows the privacy-utility tradeoffs across four datasets. We observe that MR-MTL consistently outperforms a suite of baseline methods, and that it performs at least as good as local training and FedAvg (endpoints of the personalization spectrum), except at high-privacy regimes (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* ($\varepsilon \approx 0.5, 6, 1.5$ for Fig. 3 (a, b, c) respectively); this is our key regime of interest.

Effects of silo-specific sample-level privacy. In Fig. 2 we saw that local finetuning may not improve utility as expected, motivating us to reconsider the roles of federation and personalization (§5 below). In Fig. 4, we consider the implication of silo-specific sample-level DP from the effects of privacy overhead due to additional dataset queries: (1) If IFCA [33, 69] performs cluster selection at every round (default behavior), then the extra privacy cost can be prohibitive; (2) Despite its similarity to MR-MTL, Ditto [57]'s privacy overhead makes it less competitive (see also Fig. 5).

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

Following the observations in §4, we now examine the desirable properties of a good algorithm under silo-specific sample-level privacy and understand why MR-MTL may be an attractive candidate.

Federation as noise reduction. A key message from §3 and Fig. 2 is that the utility cost from DP can be significantly smaller for FedAvg compared to local training even when the latter spends an identical privacy budget. This implies that FedAvg may have inherent benefits for DP noise reduction, despite the noise are added in the *gradient* space rather than the parameter space (as in client-level DP). Consider a simple setting of DP gradient descent: the update rule $w_k^{(t+1)} = w_k^{(t)} - \frac{\eta}{n_k}\left(z^{(t)} + \sum_{i=1}^{n_k} g_{k,i}^{(t)}\right)$ for silo k recursively expands to $w_k^{(T)} = w_k^{(0)} - \frac{\eta}{n_k}\sum_{t=0}^{T-1} z^{(t)} - \frac{\eta}{n_k}\sum_{t=0}^{T-1}\sum_{i=1}^{n_k} g_{k,i}^{(t)}$ over T steps, where $g_{k,i}^{(t)}$ is the clipped gradient of the i-th local example at step t (with norm bound c) out of a total of n_k examples, and $z^{(t)} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is the Gaussian noise added to the gradient sum at step t that targets for an overall privacy 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

$$Z^{(T)} \triangleq -\frac{\eta}{n_k} \sum_t z^{(t)} \sim \mathcal{N}\left(0,\ O\left(\frac{\eta^2 c^2 T^2 \ln(1/\delta_k)}{n_k^2 \varepsilon_k^2}\right) \cdot \mathbf{I}\right) \tag{2}$$

indeed implies that each silo's model update has an independent Gaussian random walk component [88, 7, 100] whose variance can be reduced by averaging with other silos' updates, as in FedAvg.[4] A similar reasoning applies to SGD cases since the additive DP noises are i.i.d. across the minibatches.

Model personalization for privacy-heterogeneity cost tradeoff. A major downside of FedAvg is that it may underperform simple local training due to data heterogeneity (e.g. [104] and Fig. 2), particularly given that clients in cross-silo FL often have sufficient data to fit reasonable local models. This suggests an emerging role for model personalization on top of its benefits in terms of utility [99], robustness [104], or fairness [57] under heterogeneity: our privacy model allows local training and FedAvg to be viewed as two endpoints of a *personalization spectrum* that respectively mitigate the utility costs of heterogeneity and privacy noise with identical privacy budgets (recall §3); this means that personalization methods could be viewed as interpolating between these endpoints and that various personalization methods essentially do so in different ways. However, our empirical observations motivate the following key properties of a good personalization algorithm:

1. **Noise reduction**: The effect of noise reduction is present throughout training so that the utility costs from DP can be consistently mitigated. Local finetuning is a counter-example (Fig. 2).
2. **Minimal privacy overhead**: There are little to no additional local dataset queries to prevent extra noise for DP-SGD under a fixed privacy budget. In effect, such privacy overhead can shift the utility tradeoff curve downwards, and Ditto [57] may be viewed as a counter-example (Fig. 4).
3. **Smooth interpolation along the personalization spectrum**: The interpolation between local training and FedAvg should be fine-grained (if not continuous) such that an optimal tradeoff should be attainable. Clustering [33, 69] may be viewed as a counter-example when there are no clear heterogeneity strucutre across clients.

These properties are rather restrictive and they render many promising algorithms less attractive. For example, model mixture [61, 13, 24] and local adaptation [104, 19] methods can incur linear overhead in dataset iterations, and so can multi-task learning [57, 91, 89] methods that benefit from additional training. Clustering methods [33, 22, 69, 89] can also incur overhead with cluster selection [33, 69], distillation [22], or training restarts [89], 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 case for 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 [30, 105, 94, 38, 19, 44] 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 $\lambda/2 \|w_k - \bar{w}\|_2^2$ (see Algorithm A1 for a typical instantiation). 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"). However, it is an imperfect knob as $\lambda \to \infty$ may *not* recover FedAvg under a typical optimization setup as the regularization term may dominate the gradient step $w_k^{(t+1)} = w_k^{(t)} - \eta\left(g_t + \lambda\left(w_k^{(t)} - \bar{w}^{(t)}\right)\right)$ where g_t is the noisy clipped gradient, and MR-MTL may thus underperform FedAvg in high-privacy regimes that necessitate a large λ to mitigate DP noise (Fig. 3).

MR-MTL has the attractive properties that: (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* additional privacy cost compared to local training/FedAvg as it does not involve extra dataset 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. [45, 110, 36]); this has important practical implications as will be discussed in §7. It also has fast convergence [106] and easily extends to deep learning with good empirical performance in the primal [94, 44] (cf. [10, 91, 64]). It is also sufficently extensible to handle structured heterogeneity (discussed below).

[4] The work of [49] examines the benefits of adding negatively 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 directly 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.

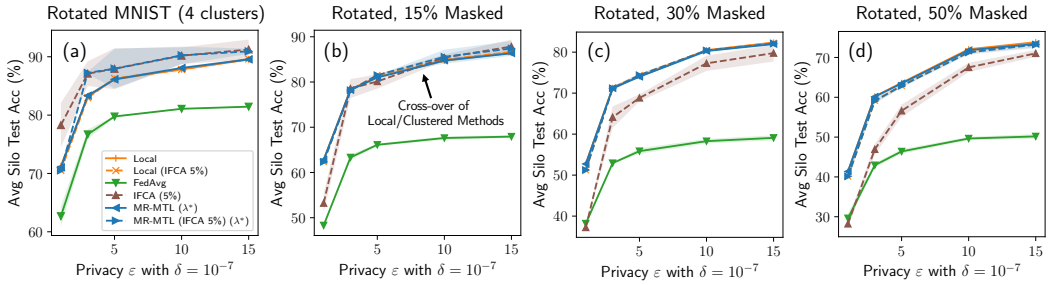

Figure 6: **Test acc $\pm$ std on Rotated & Masked MNIST**. (IFCA 5%) denotes warm-starting the method by running IFCA [33, 69] for first 5% of rounds followed by running the method within the cluster structures.

We argue through the following empirical analyses that these properties make MR-MTL a strong baseline under silo-specific sample-level DP.

Navigating the emerging privacy-heterogeneity cost tradeoff. In Fig. 5 we study the effect of the regularization strength λ on the model utility directly. There are several notable observations: (1) In both private and non-private settings, λ serves to roughly 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. We also consider Ditto [57] in Fig. 5, a state-of-the-art personalization method that resembles MR-MTL and exhibits similar behaviors, to illustrate the effect of privacy overhead from its extra local training iterations.

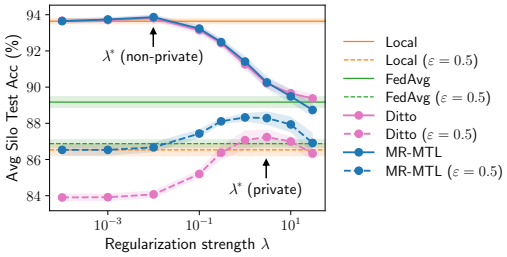

Figure 5: **Test acc $\pm$ std of MR-MTL with varying** λ (corresponding to $\varepsilon = 0.5$ in Fig. 3 (a)). Optimal points (λ^*) exist where it outperforms both ends of the spectrum under the *same* privacy. Ditto [57] gives a similar interpolation but has strictly worse privacy-utility tradeoffs due to its privacy overhead. See Appendix H.2 for extensions of this figure to other privacy settings and datasets.

MR-MTL under structured heterogeneity. We further study (1) the extensibility of MR-MTL as a strong baseline method 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 [54]. 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. 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 as a strong baseline method to match the best methods by construction. Under the 1st layer of heterogeneity, clustered FL methods [33, 69] 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. See Appendix C.1 for more details on the setup and examples of images.

We propose a simple heuristic to precondition or "warm-start" MR-MTL with a small number of training rounds by running private clustering (with IFCA [33, 69]) followed by mean-regularized training *within each formed cluster* (see Appendix D for details). We find that this simple heuristic, with convergence properties carried forward from its components [33, 30, 106], enables MR-MTL to excel at all levels of heterogeneity: in Fig. 6 (a), the preconditioning allows MR-MTL to match IFCA (optimal by construction) while local training (full personalization) does not benefit from the same preconditioning; in Fig. 6 (b, c, d), MR-MTL remains optimal across different levels of silo-specific heterogeneity (the 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 using a Bayesian framework extending on [57]. We provide expressions for the Bayes optimal estimator MR-MTL (λ^*) and describe how MR-MTL behaves with varying λ in relation to the personalization spectrum to characterize our observations from Fig. 5. Proofs and extensions are deferred to Appendix E.

Setup. We start with a total of K silos where the k-th silo holds n training samples $X_k \triangleq \{x_{k,i} \in \mathbb{R}\}_{i\in[n]}$, each normally distributed around a hidden 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 $\{w_k\}_{k\in[K]}$ 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 the silo centers 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 sample-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) sample-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.[5] Under this setting, the MR-MTL objective for the k-th silo is

$$h_k(w) = \tilde{F}_k(w) + \frac{\lambda}{2}\|w - \bar{w}\|_2^2. \tag{3}$$

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 (sub-)Gaussian, we assume one can choose c such that no clipping error is introduced w.h.p., so $\hat{w}_k \triangleq \operatorname{argmin} \tilde{F}_k(w) = \frac{1}{n}(\xi_k + \sum_i x_{k,i})$ 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 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*

$$\hat{w}_k(\lambda) = \alpha \cdot \hat{w}_k + (1 - \alpha) \cdot \hat{w}_{\setminus k}. \tag{4}$$

Note that the best λ is always 0 for *training* error (i.e. estimating the empirical mean of the local 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

$$\sigma_{\text{loc}}^2 \triangleq \frac{\sigma^2}{n} + \frac{\sigma_{\text{DP}}^2}{n^2} \tag{5}$$

denote the "local variance" of $\hat{w}_k$ around w_k due to both data sampling and privacy noise.

Behavior of MR-MTL at optimal λ^*. We first derive the following lemma using Lemma 11 of [68].

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 $\zeta_k \sim \mathcal{N}(0, \sigma_w^2)$,*

$$\sigma_w^2 \triangleq \left(\frac{1}{\sigma_{\text{loc}}^2} + \frac{K-1}{K\tau^2 + \sigma_{\text{loc}}^2}\right)^{-1} \quad \textit{and} \quad \mu_k \triangleq \sigma_w^2\left(\frac{1}{\sigma_{\text{loc}}^2} \cdot \hat{w}_k + \frac{K-1}{K\tau^2 + \sigma_{\text{loc}}^2} \cdot \hat{w}_{\setminus k}\right). \tag{6}$$

Lemma 6.2 expresses the unobserved true silo 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*

$$\lambda^* = \operatorname*{argmin}_{\lambda} \mathbb{E}\left[(w_k - \hat{w}_k(\lambda))^2 \mid \hat{w}_k, \hat{w}_{\setminus k}, \{X_k\}_{k\in[K]}\right] = \frac{1}{n\tau^2}\left(\sigma^2 + \frac{\sigma_{\text{DP}}^2}{n}\right). \tag{7}$$

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 ($\sigma_{\text{DP}}^2 \to \infty$) to encourage silos to "federate more" with others. This was empirically observed in Fig. 5. We now characterize the utility of $\hat{w}(\lambda^*)$.

[5] For simplicity, we start with the same $n, \sigma, \sigma_{\text{DP}}$ for all silos and extend to silo-specific values in Appendix E.

Corollary 6.4 (Optimal error with $\hat{w}(\lambda^*)$)**.** *The MSE of the optimal estimator $\hat{w}(\lambda^*)$ is given by*

$$\mathcal{E}^* \triangleq \mathbb{E}\left[(w_k - \hat{w}_k(\lambda^*))^2 \mid \hat{w}_k, \hat{w}_{\setminus k}, \{X_k\}_{k\in[K]}\right] = \sigma_w^2 = \frac{\sigma_{\text{loc}}^2(\sigma_{\text{loc}}^2 + K\tau^2)}{K(\sigma_{\text{loc}}^2 + \tau^2)}. \tag{8}$$

Note also that $\hat{w}(\lambda^*)$ is the MMSE estimator of w_k. Using Corollary 6.4, we can compare $\hat{w}_k(\lambda^*)$ against the endpoints of the personalization spectrum (local training and FedAvg) with the following propositions.

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

$$\Delta_{\text{loc}} \triangleq \mathcal{E}_{\text{loc}} - \mathcal{E}^* = \left(1 - \frac{1}{K}\right) \cdot \frac{\sigma_{\text{loc}}^4}{\sigma_{\text{loc}}^2 + \tau^2}. \tag{9}$$

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 $\hat{w}(\lambda^*)$ (Corollary 6.4), the FedAvg estimator incurs an additional error of*

$$\Delta_{\text{fed}} \triangleq \mathcal{E}_{\text{fed}} - \mathcal{E}^* = \left(1 - \frac{1}{K}\right) \cdot \frac{\tau^4}{\sigma_{\text{loc}}^2 + \tau^2}. \tag{10}$$

Together, Propositions 6.5 and 6.6 suggest that the effects of stronger privacy ($\sigma_{\text{DP}}^2, \sigma_{\text{loc}}^2 \to \infty$) on how MR-MTL compares against the personalization endpoints are mixed, with the benefit of MR-MTL *increasing* against local training and *diminishing* against FedAvg. They also suggest that MR-MTL has an optimal utility advantage over both the endpoints when $\sigma_{\text{loc}}^2 \approx \tau^2$ and local training performs on par with FedAvg, and the utility "bump" under privacy observed in Fig. 5 can be viewed as a result of this balance. It is worth noting that since the data variance σ^2 and heterogeneity τ^2 are often fixed in practice, the freedom for silos to vary their privacy targets (ε and σ_{DP}^2) 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 provides a characterization.

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, $\mathcal{E}(\lambda) = \left(1 - \frac{1}{K}\right)\frac{\sigma_{\text{loc}}^2 + \lambda^2\tau^2}{(\lambda+1)^2} + \frac{\sigma_{\text{loc}}^2}{K}$.*

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 also 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.

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

Theorem 6.8 suggests that with a larger λ, the utility cost from privacy can be smoothly mitigated by up to a factor of K, matching the empirical observation in Fig. 5.

7 Discussions

In previous sections, we empirically and theoretically studied the benefits of the best personalization hyperparameter λ^* for MR-MTL, but it remains open as to how such λ^* may be obtained. In this section, we take an honest look at the complications of deploying MR-MTL through the lens of the *privacy cost* of finding λ^*. There are in general several approaches: (1) a non-adaptive search (e.g. grid/random search [12]); (2) an adaptive search (e.g. grad student descent); or (3) an online estimation during training (e.g. [97, 8, 80]). Here, we focus on approach (1) since it is generic to all personalization methods and is a setting for which we have the best privacy accounting tools [62, 79] to our knowledge. We defer technical details and further discussions to Appendix F. Note that while we focus on MR-MTL, our reasoning in principle extends to all personalization methods whose advantage depends on having the best hyperparameter(s).

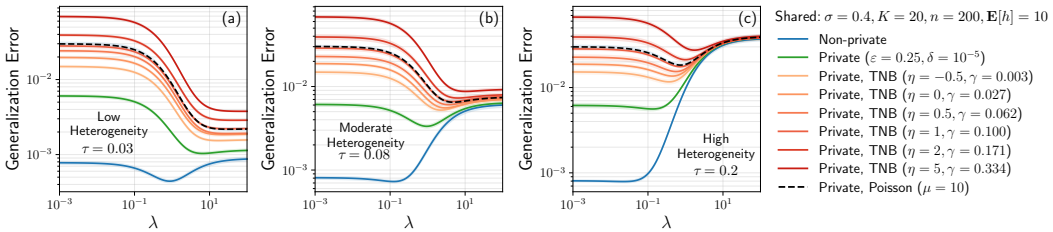

Figure 7: **Privacy costs of tuning λ on mean estimation** (setup follows §6). Labels "**Private**" and "**Non-Private**" denote the errors of varying λ with and without silo-specific sample-level DP (the privacy cost of tuning λ is *not* included). "**Private, TNB/Poisson**" [79] denotes the same errors but accounts for the privacy cost of trying on average $\mathbf{E}[h] = 10$ values of λ, with h sampled from the truncated negative binomial distribution with parameters η, γ / the Poisson distribution with parameter μ to arrive at the same $\mathbf{E}[h]$. To interpret, observe that the lowest points of "**Private, TNB/Poisson**" may be still higher than one of the endpoints of "**Private**".

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 [62, 79] 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 analogous negative results for Rényi DP (thus also for $\delta > 0$). This implies that naive tuning (as done in practice) can incur a prohibitive privacy overhead and obliterates the utility advantage of MR-MTL (λ^*) over local training/FedAvg. Instead, by making h *random*, we can make $\tilde{\varepsilon}$ *constant* w.r.t. h or at most $\tilde{\varepsilon} \leq O(\log \mathbf{E}[h])$ [62, 79]. 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 cost 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) in the first paragraph), though such procedures may not be general and/or scalable and may 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 existing methods that require more tuning will incur even larger privacy costs from hyperparameter tuning.

8 Concluding Remarks

In this work, we revisit the application of differential privacy in cross-silo FL. We examine silo-specific sample-level DP as a more appropriate privacy notion for cross-silo FL, and we point out several meaningful ways in which it differs from client-level DP commonly studied under the cross-device setting, particularly when analyzing tensions between privacy, utility, and heterogeneity. We explore and establish baselines under this privacy setting and identify desirable properties for a personalization method for balancing an emerging tradeoff between utility costs from privacy and heterogeneity. We then analyze a simple, promising method (MR-MTL) and discuss key open questions for the area at large. Some future directions include (1) extending the privacy model to cases where data subjects have multiple records across silos, (2) extending our theoretical characterization to deep learning cases or performing a large-scale empirical study, and (3) developing auto-tuning algorithms for personalization hyperparameters with minimal privacy overhead.

Acknowledgements. We thank Sebastian Caldas, Tian Li, Yash Savani, Amrith Setlur, and Peter Kairouz for helpful discussions and feedback and Thomas Steinke for guidance on implementing privacy accounting for hyperparameter tuning [79]. This work was supported in part by the NSF Grants IIS1838017 and IIS2145670, a Meta Faculty Award, an Apple Faculty Award, the Intel Private AI Center, and the CONIX Research Center. ZSW was supported in part by the NSF Award CNS2120667. Any opinions, findings, and conclusions or recommendations expressed herein are those of the author(s) and do not necessarily reflect the NSF or any other funding agency.

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] The Alzheimer's Disease Neuroimaging Initiative (ADNI). The alzheimer's disease neuroimaging initiative (adni). adni.loni.usc.edu, 05 2022. https://adni.loni.usc.edu/.

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

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

[5] Mohammad Alaggan, Sébastien Gambs, and Anne-Marie Kermarrec. Heterogeneous differential privacy. *Journal of Privacy and Confidentiality*, 7(2), 2016.

[6] 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.

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

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

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

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

[11] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE Symposium on Foundations of Computer Science*, 2014.

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

[13] Alberto Bietti, Chen-Yu Wei, Miroslav Dudik, John Langford, and Zhiwei Steven Wu. Personalization improves privacy-accuracy tradeoffs in federated optimization. In *International Conference on Machine Learning*. PMLR, 2022.

[14] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

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

[16] 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. In *NeurIPS 2019 Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

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

[18] Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine Learning*, pages 3490–3506. PMLR, 2022.

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

[20] Albert Cheu. Differential privacy in the shuffle model: A survey of separations. *arXiv preprint arXiv:2107.11839*, 2021.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. In *NIPS 2017 Workshop: Machine Learning on the Phone and other Consumer Devices*, 2017.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

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

[39] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[40] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022.

[41] Mikko A Heikkilä, Antti Koskela, Kana Shimizu, Samuel Kaski, and Antti Honkela. Differentially private cross-silo federated learning. In *Privacy Preserving Machine Learning (PPML) and Privacy in Machine learning (PriML) Joint Workshop at NeurIPS 2020*, 2020.

[42] Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020.

[43] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. In *Workshop on Federated Learning for Data Privacy and Confidentiality, NeurIPS 2019*, 2019.

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

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

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

[47] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*, pages 1023–1034. IEEE, 2015.

[48] 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.

[49] 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.

[50] 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.

[51] Pallika Kanani, Virendra J Marathe, Daniel Peterson, Rave Harpaz, and Steve Bright. Private cross-silo federated learning for extracting vaccine adverse event mentions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 490–505. Springer, 2021.

[52] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[53] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[54] 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.

[55] 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.

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

[57] 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.

[58] 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.

[59] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[60] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020.

[61] 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. In *NeurIPS 2019 Workshop on Federated Learning*, 2020.

[62] 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.

[63] 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.

[64] 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.

[65] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[66] Andrew Lowy and Meisam Razaviyayn. Private federated learning without a trusted server: Optimal algorithms for convex losses. *arXiv preprint arXiv:2106.09779*, 2021.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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. In *NeurIPS 2018 Privacy Preserving Machine Learning Workshop*, 2018.

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

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

[75] 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.

[76] 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.

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

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

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

[80] 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.

[81] Liangqiong Qu, Niranjan Balachandar, and Daniel L Rubin. An experimental study of data heterogeneity in federated learning methods for medical imaging. *arXiv preprint arXiv:2107.08371*, 2021.

[82] Shah Atiqur Rahman, Christopher Merck, Yuxiao Huang, and Samantha Kleinberg. Unintrusive eating recognition using google glass. In *2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 108–111. IEEE, 2015.

[83] 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.

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

[85] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

[86] F Reith, ME Koran, G Davidzon, and G Zaharchuk. Application of deep learning to predict standardized uptake value ratio and amyloid status on 18f-florbetapir pet using adni data. *American Journal of Neuroradiology*, 41(6):980–986, 2020.

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

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

[89] 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.

[90] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021.

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

[92] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing*, 2013.

[93] Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[95] 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.

[96] 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.

[97] 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.

[98] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

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

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

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

[107] Han Zhao, Otilia Stretcu, Alexander J Smola, and Geoffrey J Gordon. Efficient multitask feature and relationship learning. In *Uncertainty in Artificial Intelligence*, pages 777–787. PMLR, 2020.

[108] 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.

[109] 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.

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

[111] 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 has been open sourced; see details in the appendix.
   (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 has been open sourced; see details in the appendix.
   (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]

Appendix

A Additional Background **18**

B Additional Discussions **19**

B.1 Limitations . . . 19

B.2 Potential Negative Societal Impact . . . 20

C Additional Experimental Details **20**

C.1 Datasets and Models . . . 20

C.2 License/Usage Information for Datasets . . . 22

C.3 Benchmark Methods and Implementation . . . 22

C.4 Training Settings . . . 23

C.5 Hardware . . . 24

C.6 Code . . . 24

D Additional Algorithmic Details **24**

D.1 Mean-Regularized Multi-Task Learning (MR-MTL) . . . 24

D.2 IFCA Preconditioning / Warm-Starting . . . 25

E Additional Analysis Details **27**

E.1 Notations . . . 27

E.2 MR-MTL Formulation . . . 27

E.3 Assumptions . . . 27

E.4 Omitted Details and Proofs . . . 27

E.5 The Case of Heterogeneous Privacy Requirements . . . 29

F Additional Details for Private Hyperparameter Tuning (§7) **31**

F.1 Additional Background on Privacy Costs of Hyperparameter Tuning . . . 31

F.2 Interpretation of Fig. 7 . . . 32

F.3 Implementing Hyperparameter Tuning under Silo-Specific Sample-Level DP . . . 32

G Future Directions for Auto-tuning / Online Estimation of λ **33**

H Additional Experimental Results **34**

H.1 Local Finetuning (Extension of Fig. 2) . . . 34

H.2 Utility of MR-MTL as a Function of λ (Extension of Fig. 5) . . . 34

H.3 Subsampled ADNI Dataset . . . 35

H.4 Additional Discussions . . . 36

A Additional Background

Rényi Differential Privacy (RDP). In this work, we make use of a relaxation of different privacy known as Rényi Differential Privacy [77] for tight privacy accounting.

Definition A.1 (Rényi Differential Privacy (RDP) [77])**.** A randomized algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is (α, ε)-RDP with order $\alpha > 1$ if for any adjacent datasets $x, x' \in \mathcal{X}^n$,

$$D_\alpha(M(x) \| M(x')) \leq \varepsilon, \tag{11}$$

where $D_\alpha(P\|Q)$ is the Rényi divergence[6] between distributions P and Q:

$$D_\alpha(P\|Q) \triangleq \frac{1}{\alpha - 1} \log \mathop{\mathbb{E}}_{x \sim P} \left[\left(\frac{P(x)}{Q(x)} \right)^{\alpha - 1} \right]. \tag{12}$$

Under Rényi DP, the privacy composition is simple: if every step of an algorithm satisfies (α, ε)-RDP, then over T steps the algorithm satisfies $(\alpha, T\varepsilon)$-RDP. The following lemma from [15, 17] provides a conversion from RDP to standard (ε, δ)-DP guarantees.

Lemma A.2 (Conversion from Rényi DP to approximate DP [15, 17])**.** *If a mechanism M satisfies $(\alpha, \varepsilon(\alpha))$-RDP, then for any $\delta > 0$, it also satisfies $(\varepsilon(\delta), \delta)$-DP where*

$$\varepsilon(\delta) = \inf_{\alpha > 1} \varepsilon(\alpha) + \frac{1}{\alpha - 1} \log \left(\frac{1}{\alpha \delta} \right) + \log \left(1 - \frac{1}{\alpha} \right). \tag{13}$$

Zero-Concentrated Differential Privacy (zCDP). A closely related privacy notion is zero-concentrated DP (zCDP [15]), where ρ-zCDP is equivalent to satisfying $(\alpha, \rho\alpha)$-Rényi DP simultaneously for all orders α. Thus, algorithms that satisfy zCDP guarantees are compatible with standard RDP accounting routines implemented in open-source libraries (e.g. TensorFlow Privacy [72]). In our work, we make use of zCDP and a related result for the Exponential Mechanism [87] for tight privacy composition when implementing private cluster selection for IFCA [33, 69] (see below and Appendix C.3).

Exponential Mechanism for Private Selection. The Exponential Mechanism (EM) is a standard algorithm for making private selection from a set of candidates based on their scores [75]. Specifically, there is a dataset $x \in \mathcal{X}^n$ requiring DP protection, and a scoring function $s : \mathcal{X}^n \times [G] \to \mathbb{R}$ that evaluates a set of candidates $g \in [G]$. We want to pick the candidate with the highest score (i.e. $\operatorname{argmax}_{g \in [G]} s(x, g)$) subject to $(\varepsilon, 0)$-DP for neighboring datasets x, x'. The mechanism M is defined by setting the probability of choosing any $g \in [G]$ as

$$\Pr[M(x) = g] = \frac{\exp\left(\frac{\varepsilon}{2\Delta} \cdot s(x, g)\right)}{\sum_{g' \in [G]} \exp\left(\frac{\varepsilon}{2\Delta} \cdot s(x, g')\right)}, \tag{14}$$

where Δ is the sensitivity of the scoring function. EM also satisfies $\frac{1}{8}\varepsilon^2$-zCDP [87] and thus $(\alpha, \frac{\alpha}{8}\varepsilon^2)$-RDP for all α. A variant of EM is the Permute-and-Flip mechanism [70].

The Exponential Mechanism can be implemented as "Report Noisy Max" with Gumbel noise: we can add independent noises drawn from the Gumbel distribution with scale $\frac{2\Delta}{\varepsilon}$ to the candidate scores $s(x, g)$ for all $g \in [G]$ and simply report the max noisy score. If the score function is a loss metric (where we want the minimum instead of the maximum), we can similarly implement "Report Noisy Min" by subtracting the Gumbel noises from the scores and report the minimum. The latter is used in our implementation for private cluster selection, where clients select clusters with lowest loss or error rate (i.e. $1 -$ accuracy); see Appendix C.3 for more details.

Privacy Budgeting. A typical accounting workflow, as used in our experiments, thus involves (1) composing the RDP guarantees of all private operations in the algorithm and (2) trying a list of α values that give the lowest ε for a target δ when converting back to (ε, δ)-DP that captures the overall privacy cost. For SGD training, we also use existing results on privacy amplification via subsampling [78, 1]: if a gradient step is (ε, δ) w.r.t. the dataset *without* amplification and the gradient is computed with a minibatch (assumed to be a random sample) of batch size $b = q/n$ where q is the sampling ratio and n is the size of the dataset, then the privacy of the gradient step is amplified to $(O(q\varepsilon), \delta)$-DP. In a silo-specific sample-level DP setup, the size of the dataset n_k at each silo thus

[6] The Rényi divergence at $\alpha = 1$ is defined as $D_1(P\|Q) \triangleq \mathop{\mathbb{E}}_{x \sim P} \left[\log \left(\frac{P(x)}{Q(x)} \right) \right] = \lim_{\alpha \to 1} D_\alpha(P\|Q)$, which is also the KL divergence.

determines the extent of the amplification, and thus even if silos target for the same (ε, δ) sample-level DP, they may end up adding different amounts of noise when running DP-SGD (mentioned in the **Training Setup** paragraph of §4). For this reason our experiments primarily focus on having the same privacy target for all silos.

Heterogeneous Differential Privacy. A related privacy notion is *heterogeneous DP* [5, 47], where each item within a dataset to be protected by DP may opt for a different (ε, δ) target. Our setting primarily focuses on different (ε, δ) values for disjoint datasets, and all items within a specific dataset share the same DP target.

See also Appendix F for additional background relating to Section §7 (private hyperparameter tuning).

B Additional Discussions

B.1 Limitations

We discuss below the limitations of this work in addition to Section §7.

When multiple records map to the same entity. In this paper we studied the application of silo-specific sample-level differential privacy in cross-silo federated learning. While this is an important initial step towards a more suitable privacy model for cross-silo FL (in contrast to the commonly studied client-level DP model), we assume that each entity that requires privacy protection has at most one record (training example) across silos (e.g. a single patient has one medical record at a hospital).

There are two characteristic cases where this assumption does not hold for all items in a silo:

- **Multiple records within a silo map to the same entity.** One example would be a student re-enrolling at the same school for multiple degree programs, thus creating multiple student records at the same silo. In such cases, the silo curator may need to carefully apply group privacy or other methods for ensuring entity-level privacy [55] to protect the entity rather than its records.
- **Multiple records across silos map to the same entity.** One example would be a person having multiple credit cards at different banks. This case is more intricate as it is harder to precisely account for the DP guarantee for this entity without knowing (1) the silos in which this entity has appeared and (2) the specific privacy targets for each of those silos. In this case, the silos may cooperate to run *private set intersection* (e.g. as considered in [67]) to privately identify this scenario, but this would by itself come at a privacy cost.

These cases are interesting avenues for future research on private cross-silo learning.[7]

Extending the analysis to deep learning cases. In Section §6 we use federated mean estimation as a simplified setting for analyzing the behavior of MR-MTL under silo-specific sample-level privacy. While the analysis provides adequate insights into the empirical phenomena in Fig. 5, it is a simple model that does not consider the dynamic aspects of the learning settings, including (1) the Gaussian random walk component of the model updates due to DP noise applied over many training rounds, (2) the effect of communication frequency on the effect of noise reduction, (3) the concept of "client drifts" (as considered in [52]) as a result of heterogeneity and how it interfaces with the DP noises, and (4) how overparameterization may affect all of the above.

Caveats of cross-silo learning with very large local datasets. In contrast to cross-device federated learning, cross-silo federated learning is typically characterized by having a limited number of clients, each with a large local dataset. The term "large" is relative because it describes the sufficiency of the datasets for fitting good local models of a *specific class*; for example, 500 examples are likely sufficient to fit linear regression of 10 parameters, but very likely insufficient to learn a transformer [98]. In this sense, many FL problems in practice – such as large commercial banks running regression on tabular data – will in fact have local training to be the *optimal* strategy, as long as there are sufficient local data and the data from other silos are not of the same local distribution. In these cases, one should expect MR-MTL to opt for $\lambda^* \approx 0$ as federated learning is not needed at all, and thus its advantages under privacy will also be minimal.

[7] The case of having individual records corresponding to multiple entities at once (e.g. one record for all family members) is slightly less interesting since sample-level privacy would protect all of the entities.

B.2 Potential Negative Societal Impact

Our work studies the empirical behaviors that arise when applying an alternative model of differential privacy to cross-silo federated learning, and we provide a strong baseline method (MR-MTL) that fares well in this setting. In this sense, our work sheds light on and facilitates the development of a previously underexplored area of differentially private federated learning. However, because MR-MTL requires selecting a good regularization stength λ, one potential negative impact is that users may excessively tune λ on a private dataset and inadvertently leak privacy via the choice of λ (perhaps qualitatively rather than quantitatively); for example, if a silo chose a large λ for better performance, then in principle its data would look somewhat more similar to the "average" of the silo datasets. Moreover, our privacy model requires that silos add their own independent noises for their own DP targets, and this requirement may not be followed correctly (either deliberately or inadvertently) to provide vacuous DP guarantees for people's data.

C Additional Experimental Details

C.1 Datasets and Models

Table A1 summarizes the datasets, tasks, and models considered in our experiments. In the following, we provide details on each.

Dataset	Task	# Clients (Silos)	Input Dim	Min n_k	Max n_k	Learner
Vehicle	Classification	23	100	872	1933	SVM
School	Regression	139	28	15	175	Linear
Google Glass (GLEAM)	Classification	38	180	699	776	SVM
Heterogenous CIFAR-10	Classification	30	$32 \times 32 \times 3$	1515	1839	ConvNet
Rotated & Masked MNIST	Classification	40	$28 \times 28 \times 1$	1500	1500	ConvNet
Subsampled ADNI	Regression	9	$32 \times 32 \times 1$	45	2685	ConvNet

Table A1: Summary of datasets, tasks, and models for our empirical studies. n_k denotes the number of training examples on client k.

Vehicle [26]. The Vehicle Sensor dataset is a binary classification dataset containing $K = 23$ data silos. Each silo (sensor) has acoustic and seismic measurements for a road segment, with each data point being a 100-dimensional feature vector describing the measurements when a vehicle passes through the road segment. The goal is to predict between two predetermined types of vehicles. We use a train/test split of 75%/25% following previous work [91], yielding $872 \leq n_k^{\text{train}} \leq 1933$ training examples on each client. We use simple linear SVMs for classification following [91, 57]. It is a suitable dataset for cross-silo FL because the number of silos K is small while each silo has sufficient data to fit a good local model, as opposed to cross-device datasets such as FEMNIST [16] where K is large but each silo has little data to learn a useful model. Moreover, we can use tight privacy budgets due to reasonably large local datasets (in terms of sufficiency for fitting a good local model) and SVMs (which are relatively noise-tolerant since decision boundaries only depend on support vectors). The dataset is accessible from the original authors.[8]

School [35, 9, 111]. The School dataset originated from the now-defunct Inner London Education Authority.[9] It is a regression dataset for predicting the exam scores of 15,362 students distributed across 139 secondary schools. Each school has records for between 22 and 251 students, and each student is described by a 28-dimensional feature vector capturing attributions such as the school ranking, student birth year, and whether the school provided free meals. We perform 80%/20% train/test split in Fig. 3 (with $15 \leq n_k^{\text{train}} \leq 175$ training examples in each silo), and additionally consider 50%/50% and 20%/80% train/test split in Fig. A8. We use simple linear regression models following previous work (e.g. [9, 111, 36]) to predict student scores. Like the Vehicle dataset, the School dataset is a natural cross-silo FL dataset with a limited number of clients K, each with roughly sufficient data to fit a reasonable local model. The dataset is available from [111].

[8] https://web.archive.org/web/20110515133717/http://www.ece.wisc.edu:80/~sensit/
[9] https://en.wikipedia.org/wiki/Inner_London_Education_Authority

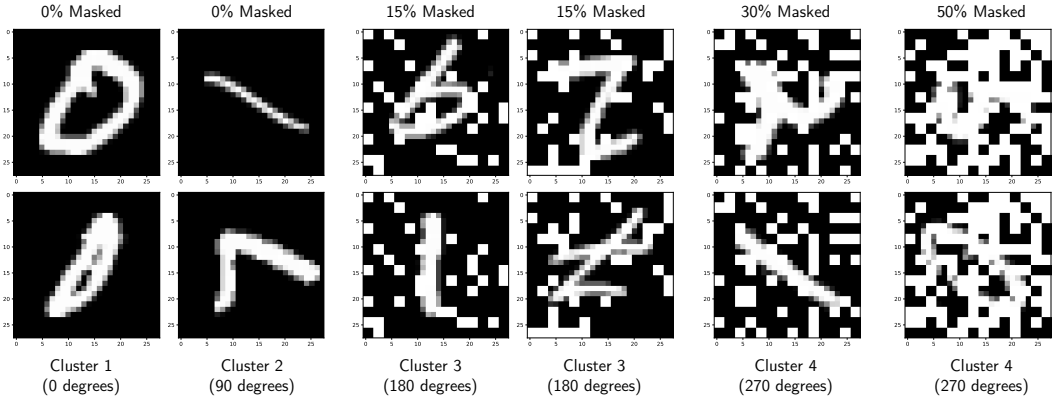

Figure A1: **Example images of the Rotated & Masked MNIST dataset**. Each column corresponds to two images from the same (random) client from the specified cluster (thus they have the same client-specific random mask). Labels for each column from left to right: (0, 0), (1, 7), (9, 1), (2, 7), (2, 1), (5, 0).

Google Glass Eating and Motion (GLEAM) [82]. We also benchmark on the GLEAM dataset, a real-world head motion tracking dataset for binary classification. The motion data is collected with Google Glass, and the task is to classify the activity of the wearer (eating or not). There are in total $K = 38$ silos (wearers) and 27800 data points, with each silo containing $699 \leq n_k \leq 776$ data points. Each data point is a 180-dimensional feature vector capturing head movement of the wearers. Linear models yield reasonable utility on GLEAM and thus we use linear SVMs following previous work [91]. Like Vehicle and School, this is a suitable dataset for cross-silo FL given a small K and relatively large local datasets.

Heterogeneous CIFAR-10 Dataset. We additionally evaluate on CIFAR-10 [53], with heterogeneous client data split following previous work [94, 90] (based on the code provided by [90]). The dataset has $K = 30$ clients (silos) in total, and data heterogeneity is generated with each client having a random number of samples from 5 randomly chosen classes out of the 10 classes. Each client has $1515 \leq n_k \leq 1839$ training examples. We use a simple convolutional network with the following layers: [Conv 3×3 with 32 channels, ReLU, MaxPool 2×2 with stride 2, Conv 3×3 with 64 channels, ReLU, MaxPool 2×2 stride 2, Linear]. No padding is used for convolutional layers.

Rotated & Masked MNIST. We adapt the original MNIST dataset [54] to study the effect of structured heterogeneity on MR-MTL. For the 60000/10000 train/test images, we first perform a shuffle and then evenly separate them into $K = 40$ clients (silos), each with 1500/250 train/test images with roughly uniform distribution on the labels. We then randomly separate the silos into 4 groups of 10, and apply rotations of $\{0°, 90°, 180°, 270°\}$ to each group respectively; silos within the same group have the same rotations applied to the images, thus forming 4 natural silo clusters. To add *intra-cluster* heterogeneity, we then apply *silo-specific* random masks of 2×2 white patches; that is, all images in the same silo has the same mask, and no two silos have the same mask with very high probability. The random masks are akin to those considered in [40]. The white patches of the random mask do not overlap, and the mask ratio is the probability of a patch being applied (so the specified percentage of masked area is an expectation). Examples of generated images are shown in Fig. A1. These image transformations introduce two types of heterogeneity identified by [50]: "covariate shift" (skew of feature distributions) and "concept drift" (same label, different features). The model architecture is the same as the one used for heterogeneous CIFAR-10.

Subsampled Alzheimer's Disease Neuroimaging Initiative (ADNI) Dataset [2]. We additionally benchmark on the ADNI dataset, which is a real-world dataset containing brain PET scans of Alzheimer's disease patients, patients with mild cognitive impairment, and healthy people taken from multiple institutions [81]. It is a regression dataset for predicting the SUVR value (a scalar ranged roughly between 0.8 and 2) from the PET scan images of a brain. We simplified the full dataset for faster training by subsampling the axial slices generated for each brain PET scan (96 slices for scan), turning them into a gray scale image, downsampling them to size 32×32, and randomly splitting them into a 75%/25% train/test sets. There are in total $K = 9$ silos containing a total of 11040 images; each silo corresponds to a different equipment that took the PET scans and contains $45 \leq n_k^{\text{train}} \leq 2685$ training examples. See Fig. 4 of [86] and Fig. 4 of [81] for sample images. The

model architecture is a simple convolutional network with the following layers: [Conv 5×5 with 32 channels, ReLU, MaxPool 2×2 with stride 2, Conv 5×5 with 64 channels, ReLU, MaxPool 2×2 stride 2, Linear]. No padding is used for convolutional layers.

C.2 License/Usage Information for Datasets

Vehicle. The Vehicle dataset was made publicly available by the original authors as a research dataset [26] and license information was unavailable. It has been subsequently used in many work (e.g. [91]).

School. The original entity that collected the School dataset [35] is defunct and license information was unavailable. The dataset has been made publicly available [111] and used extensively in previous work (e.g. [111, 107]).

Google Glass (GLEAM). The GLEAM dataset was made publicly available by the original authors and can be used for any non-commercial purposes. See this URL[10] for license and usage information.

Heterogeneous CIFAR-10. The original CIFAR-10 dataset is available under the MIT license.

Rotated & Masked MNIST. The original MNIST dataset is available under the CC BY-SA 3.0 license.

Subsampled ADNI. As per the Data Use Agreement of the ADNI dataset:[11]

> Data used in preparation of this manuscript were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this manuscript. A complete listing of ADNI investigators can be found at this URL.[12]

The data sharing and publication policy of the ADNI dataset can be found at this URL.[13] Access to the dataset must be approved by ADNI.

C.3 Benchmark Methods and Implementation

We provide more details on our benchmark personalization methods below.

Local finetuning [99, 104, 19] is one of the simplest but most effective personalization methods: once clients obtain a shared model via federated training (e.g. FedAvg), they can personalize it with additional training steps over their local dataset. This simple strategy has been shown to work very well empirically [99, 104, 19], with the work of [19] providing theoretical support that it can asymptotically achieve comparable performance to other more sophisticated methods. Moreover, in contrast to other local adaptation methods like distillation [104], local finetuning's privacy footprint under our privacy model can be easily controlled (by limiting the number of finetuning and total training steps) without qualitatives change in its behavior. In our experiments, local finetuning is implemented as FedAvg followed by local training, each taking 50% of the total number of training rounds to ensure an identical privacy budget as other baseline methods.

Ditto [57] is the current state-of-the-art method for personalization with provable benefits of robustness and fairness. It is closely related to MR-MTL because it similarly trains personalized models while ℓ^2-regularizing them towards a global model, but it differs from MR-MTL in that its global model can be obtained by a standalone solver. In particular, when no privacy is added, Ditto's modularity allows it to perfectly interpolate between local and FedAvg training with its regularization strength λ. In our experiments, we implement Ditto with the FedAvg solver and use a minimal number of iterations over the local datasets to avoid excessive privacy overhead.

[10] http://www.healthailab.org/data.html

[11] https://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Data_Use_Agreement.pdf

[12] https://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

[13] https://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_DSP_Policy.pdf

Mocha [91] is a multi-task learning framework tailored for federated settings. During training, it simultaneously learns the personalized models as well as a client-relationship matrix which can model both positive and negative client relationships (in contrast, clustering methods only focus on positive relationships). One disadvantage of Mocha is that it applies to convex problems only. In particular, the original paper uses a dual formulation for efficient training, but for fair comparison and compatibility with privacy primitives (especially DP-SGD [92, 11, 1]), we implemented Mocha in its primal form and trained it with SGD in our experiments. Similarly, we align Mocha to other baselines in terms of the number of training steps to prevent privacy overhead.

IFCA [33] (and the conceptually similar **HypCluster** [69]) is a simple clustering framework proposed as an extension to FedAvg. In every round of IFCA training, the server sends k models (cluster centroids) to all clients; each client locally evaluates them over its local training data and selects the one with the lowest loss. Each client then locally trains on the selected model and returns updates only for this model, along with its index. Clients that selected the same model indices can be viewed as belonging to the same cluster. We use IFCA as the representative clustering method due to its performance, simplicity, and practicality.

The privacy overhead of IFCA comes in the form of *private cluster selection*: when each client evaluates the incoming models (cluster centroids) on the local datasets, this procedure must be privatized as the selection itself may leak information about the dataset. Private selection can be implemented via the exponential mechanism [75] with sharp accounting via a bounded range analysis [87] (discussed in Appendix A), but one must decide (e.g. as a privacy budget $\varepsilon_{\text{select}}$ for the same δ) how to share the selection cost with DP-SGD under a fixed total privacy budget $\varepsilon_{\text{total}}$.

Mitigating strategies for private cluster selection. In our experiments, we observed that if private selection is implemented naively, it can incur a prohibitive privacy overhead and destroys the final utility (e.g. Fig. 4). There are two important reasons: (1) unlike DP-SGD, no privacy amplification applies to private selection, and (2) the sensitivity of the training loss (which is used to select cluster centroids) is unbounded in general and must be clipped to a reasonable value (e.g. ≤ 1). We propose two mitigation strategies:

1. **Use accuracy instead of loss.** For the cluster selection metric (i.e. the score function $s(x, g)$ where x is the local dataset and g is a particular cluster centroid), we use the error rate ($1 -$ accuracy) instead of the loss (which is used by the original authors [33, 69]). The rationale is that accuracy is a low-sensitivity function, particularly in cross-silo settings: one can show that, by enumerating the cases where the differing example between the neighboring datasets x and x' are correctly/incorrectly classified under addition/removal/replacement notions of DP, the sensitivity Δ_{acc} of s is bounded as
$$\Delta_{\text{acc}} = \max_{g} \max_{x,x'} |s(x, g) - s(x, g')| \leq \frac{1}{n-1} \tag{15}$$
where n is the size of the local dataset. Since n can be large in cross-silo settings, the sensitivity can be orders of magnitude smaller than that of the loss function. With small sensitivity, we heuristically set the per-round selection privacy budget to a very small $\varepsilon_{\text{select}} = 0.03 \cdot \varepsilon_{\text{total}}$. Despite this, however, the cost of private selection can still grow quickly over the entire training process and considerably eat into DP-SGD's privacy budget.
2. **Truncate the number of cluster selection rounds.** Our second strategy is to simply run less rounds of cluster selection (e.g. to 10% of total number of training rounds, as in Fig. 3). This follows from the empirical analysis of [33] as well as our own experimental observation that clusters tend to converge quickly, though in some cases, clusters may not fully converge within 10% of training rounds.

Despite these strategies, however, the private selection cost can still lead to a steep utility hit. Note also that the new hyperparameter $\varepsilon_{\text{select}}$ can be tuned; for a fixed total budget $\varepsilon_{\text{total}}$, a small $\varepsilon_{\text{select}}$ means the budget for DP-SGD $\varepsilon_{\text{train}}$ is not affected by much, but the selected clusters would be very noisy and inaccurate; a large $\varepsilon_{\text{select}}$ leads to less noisy clusters (and thus smaller intra-cluster heterogeneity), but DP-SGD will correspondingly use larger noise and hurt optimization.

C.4 Training Settings

Optimizers. For simplicity of hyperparameter tuning and experimental controls, we use minibatch DP-SGD for client local training without local or server momentum for all experiments (in fact,

FedAvgM [43] and FedAdam [85] were not found to be helpful on Vehicle and School). While there are more efficient solvers for the Vehicle and School datasets since we are dealing with convex problems, we want compatibility with DP-SGD [92, 11, 1] as well as tight privacy accounting with privacy amplification via subsampling (i.e. via minibatch training).

Hyperparameters. For all datasets and all methods, we set silos to train for 1 local epoch in every round (except Ditto [57] which takes 2 local epochs). For Vehicle, GLEAM, School, Heterogeneous CIFAR-10, Rotated & Masked MNIST, and subsampled ADNI respectively, the local batch size across all silos are fixed with $B = 64, 64, 32, 100, 100, 64$, and the clipping norm for per-example gradients are heuristically set to $c = 6, 6, 1, 8, 1, 0.5$. Vehicle uses $T = 400$ rounds for most experiments (except Fig. 2 which trains for $T = 200$ rounds); School, Google Glass, Heterogeneous CIFAR-10, and Rotated & Masked MNIST use $T = 200$; and ADNI uses $T = 500$.

For multi-task learning methods (MR-MTL, Ditto [57], Mocha [91]), we sweep the regularization strength across a grid of $\lambda \in [0.0001, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10]$ to find the best λ^* wherever applicable (e.g. Figs. 3, 5 and A8). To compensate for the change in the gradient magnitude, we also sweep different client learning rates across a grid of $\eta \in [0.001, 0.003, 0.01, 0.03, 0.1, 0.3]$; for fair comparison, the same grid of η is swept for methods that do not involve λ (e.g. IFCA, local finetuning).[14] For Fig. 2 and Fig. A2, the learning rate is fixed to $\eta = 0.01$. For all datasets, the chosen privacy parameter δ satisfy $\delta < n_k^{-1.1}$ where n_k is the local training dataset size.

Evaluation Protocol. For all datasets, we evaluate methods by the average test metric (accuracy or MSE) across the silos, weighted by their respective test sample counts. Weighted averaging allows the final test metric to reflect a method's performance over the individual test samples of the combined dataset across silos, thus fairer and more aligned (compared to uniform averaging of silo test metrics) with our privacy model where each test sample represents an entity requiring protection.

C.5 Hardware

Experiments for Vehicle, School, and Google Glass (GLEAM) are run on commodity CPUs and experiments for Heterogeneous CIFAR-10, Rotated & Masked MNIST and ADNI are run on four NVIDIA RTX A6000 GPUs.

C.6 Code

Our experiments are implemented in Python with NumPy [39], JAX [14] and Haiku [42]. For private training, JAX is used to vectorize DP-SGD over per-example gradients [93]. Code is available at `https://github.com/kenziyuliu/private-cross-silo-fl`.

D Additional Algorithmic Details

D.1 Mean-Regularized Multi-Task Learning (MR-MTL)

Algorithm A1 describes the canonical instantiation of MR-MTL [94]. Its key ingredient is the mean-regularization (Line 6 of Algorithm A1) that forces the local personalized models w_k to be close to their mean $\bar{w}$. Silo-specific sample-level privacy is added by privatizing the local gradients as in DP-SGD [92, 11, 1].

Privacy of MR-MTL. Since the iterates of $\bar{w}^{(t)}$ are already differentially private (as they are the average of the private iterates $w_k^{(t)}$), the additional regularization term $\frac{\lambda}{2}\|w_k^{(t)} - \bar{w}_k^{(t-1)}\|_2^2$ (and hence MR-MTL) does not incur privacy overhead compared to local and FedAvg training.

Weighted vs Unweighted Model Updates. It is customary to weigh the model updates from each client (silo) by its training example counts, as in the original FedAvg implementation [71]. However, note that under silo-specific sample-level privacy with *addition/removal* notions of DP, the example

[14] As discussed in Section §7, releasing the results from repeated experiments (possibly with different hyperparameters) may technically compromise the privacy of the datasets [62, 79]. In our case, we are primarily interested in understanding the behaviors and tradeoffs that emerge under silo-specific sample-level DP, and thus for experimental control and ease of comparison we do not account for the privacy costs from hyperparameter tuning and repetitions. We also use only public datasets in our experiments.

Algorithm A1 Mean-Regularized Multi-Task Learning

1: **Input:** Initial client models $\{w_k^{(0)}\}_{k\in[K]}$, and mean model $\bar{w}^{(0)}$.
2: **for** training round $t = 1, ..., T$ **do**
3: Server sends $\bar{w}^{(t-1)}$ to every client.
4: **for** client $k = 1, ..., K$ **in parallel do**
5: Set model iterate: $w_k^{(t)} \leftarrow w_k^{(t-1)}$.
6: For every batch (x, y), client updates $w_k^{(t)}$ using SGD or DP-SGD with batch loss $\ell(w_k^{(t)}, x, y)$ and gradient

$$\nabla_{w_k^{(t)}} \left[\ell\left(w_k^{(t)}, x, y\right) + \frac{\lambda}{2} \left\| w_k^{(t)} - \bar{w}^{(t-1)} \right\|_2^2 \right].$$

7: Return model update $\Delta_k^{(t)} = w_k^{(t)} - w_k^{(t-1)}$.
8: Server updates $\bar{w}^{(t)} = \bar{w}^{(t-1)} + \frac{1}{K}\sum_{k=1}^{K} \Delta_k^{(t)}$ (may weigh $\Delta_k^{(t)}$ by client example counts).
9: **Output:** Personalized models $w_k^{(T)}$ for all $i \in [K]$.

counts on each silo may itself leak sensitive information (e.g. when a silo only has one record). This is less of an issue with the *replacement* notion of DP, since neighboring datasets would have the same example counts. In our experiments we use weighted model updates and thus implicitly assume the replacement notion of DP. Nevertheless, the resulting privacy guarantees are constant factors apart and empirically we did not observe significant changes in performance when using unweighted aggregation.

D.2 IFCA Preconditioning / Warm-Starting

In Section §5, we considered an extension to MR-MTL where training is "warm-started" by a small number of rounds of clustering (via IFCA [33, 69]), followed by MR-MTL training *within* each formed cluster. Pseudocode for this procedure is shown in Algorithm A2.

Privacy of IFCA Preconditioning. Observe that as with MR-MTL, the gradient steps satisfy silo-specific sample-level DP regardless or whether the steps are made on the cluster models or the personalized models. IFCA preconditioning introduces privacy overhead in the form of private cluster selection (Line 6 of Algorithm A2; discussed in Appendices A and C.3), which splits the total privacy budget with DP-SGD. As a result the noise scale for DP-SGD would be increased and can be numerically determined.

Algorithm A2 IFCA-Preconditioned MR-MTL

Input: Initial client models $\{w_k^{(0)}\}_{k\in[K]}$, number of clusters G, initial cluster models $\{\bar{w}_g^{(0)}\}_{g\in[G]}$, total number of rounds T, and the number of initial clustering rounds T_{cluster}.

`# IFCA preconditioning rounds`

for IFCA training round $t = 1, ..., T_{\text{cluster}}$ **do**

  Server sends cluster models $\{\bar{w}_g^{(t-1)}\}_{g\in[G]}$ to every client.

  for client $k = 1, ..., K$ **in parallel do**

    Use Exponential Mechanism (Appendix A) to select best cluster $\bar{w}_{g^*(k)}^{(t-1)}$ from $\{\bar{w}_g^{(t-1)}\}$ with loss/error rate function s and local dataset (X_k, Y_k).

    Set model iterate: $w_k^{(t)} \leftarrow \bar{w}_{g^*(k)}^{(t-1)}$.

    For every batch (x, y), client updates $w_k^{(t)}$ using SGD or DP-SGD with batch loss $\ell(w_k^{(t)}, x, y)$ and gradient $\nabla_{w_k^{(t)}} \ell\left(w_k^{(t)}, x, y\right)$.

    Return model update $\Delta_k^{(t)} = w_k^{(t)} - \bar{w}_{g^*(k)}^{(t-1)}$ and selected cluster index $g^*(k)$.

  for each cluster $g \in [G]$ **do**

    Server applies (weighted) model updates to cluster g with the associated client indices:

$$\bar{w}_g^{(t)} = \bar{w}_g^{(t-1)} + \frac{1}{|\mathcal{K}_g|} \sum_{k\in\mathcal{K}_g} \Delta_k^{(t)}, \quad \text{where } \mathcal{K}_g \equiv \{k \in [K] \mid g^*(k) = g\}. \tag{16}$$

`# MR-MTL rounds with regularization towards frozen cluster centroids`

for MR-MTL training round $t = T_{\text{cluster}} + 1, ..., T$ **do**

  Server sends cluster models $\{\bar{w}_g^{(t-1)}\}_{g\in[G]}$ to every client.

  for client $k = 1, ..., K$ **in parallel do**

    Client retrieves the last selected cluster centroid $\bar{w}_{g^*(k)}^{(t-1)}$.

    Set model iterate: $w_k^{(t)} \leftarrow w_k^{(t-1)}$ (**using the personalized model from last round**).

    For every batch (x, y), client updates $w_k^{(t)}$ using SGD or DP-SGD [92, 11, 1] with batch loss $\ell(w_k^{(t)}, x, y)$ and gradient

$$\nabla_{w_k^{(t)}} \left[\ell\left(w_k^{(t)}, x, y\right) + \frac{\lambda}{2}\left\|w_k^{(t)} - \bar{w}_{g^*(k)}^{(t-1)}\right\|_2^2\right].$$

    Return model update $\Delta_k^{(t)} = w_k^{(t)} - \bar{w}_{g^*(k)}^{(t-1)}$ and the cluster index g^*.

  for each cluster $g \in [G]$ **do**

    Server applies (weighted) model updates to cluster g with the associated client indices:

$$\bar{w}_g^{(t)} = \bar{w}_g^{(t-1)} + \frac{1}{|\mathcal{K}_g|} \sum_{k\in\mathcal{K}_g} \Delta_k^{(t)}, \quad \text{where } \mathcal{K}_g \equiv \{k \in [K] \mid g^*(k) = g\}. \tag{17}$$

Output: Personalized models $w_k^{(T)}$ for all $i \in [K]$.

E Additional Analysis Details

In this section we provide additional details for the analysis presented in Section §6. We also extend the analysis to the case with varying n, σ, σ_{DP} for each silo in Appendix E.5.

E.1 Notations

Unless otherwise specified, we used the following notations throughout the analysis:

- K denotes the total number of silos (clients) with indices $k \in [K]$;
- n denotes the number of examples on each silo;
- w_k denotes the true center of silo k's data distribution;
- $X_k \equiv \{x_{k,i}\}_{i\in[n]}$ denotes the local dataset with n data points;
- $\hat{w}_k$ denotes the best local estimator of w_k;
- $\bar{w}$ denotes the average of local estimators;
- $\hat{w}_{\setminus k} \triangleq \frac{1}{K-1}\sum_{j\neq k, j\in[K]} \hat{w}_j$ denotes the external average estimators from the perspective of k;
- σ^2 denotes the sampling variance of the local data X_k;
- τ^2 denotes the sampling variance of the local data centers w_k (hence a measure of data heterogeneity across silos); and
- σ_{DP}^2 denotes the DP noise variance on each silo to satisfy silo-specific sample-level privacy.

E.2 MR-MTL Formulation

The general formulation of the mean-regularized MTL objective may be expressed as

$$\min_{w_k, k\in[K]} \frac{1}{K}\sum_{k=1}^{K} h_k(w_k) \quad \text{with} \quad h_k(w) \triangleq F_k(w) + \frac{\lambda}{2}\|w - \bar{w}\|_2^2, \tag{18}$$

where $F_k(\cdot)$ is the local objective for client k, $\bar{w} \triangleq \frac{1}{K}\sum_{k=1}^{K} w_k$ is the average model, and $\lambda \geq 0$ is the regularization strength. A larger λ enforces the models to be closer to each other, and $\lambda = 0$ reduces the problem to local training. In particular, unlike Ditto [57], MR-MTL may *not* recover FedAvg [71] under SGD as $\lambda \to \infty$, since λ essentially changes the ratio between the local objective gradients and the regularization gradients. For the purposes of our analysis, we assume F_k is strongly convex for all $k \in [K]$.

E.3 Assumptions

Our characterization of MR-MTL on makes the following simplifying assumptions.

Assumption E.1. All clients (silos) have the same number of data points n, data sampling variance σ^2, and DP noise variance σ_{DP}^2.

Assumption E.2. A sufficiently large clipping c can be selected such that $\|x_{k,i}\|_2 \leq c$ with high probability.

Note that Assumption E.1 primarily serves to make results cleaner and easily interpretable (see Appendix E.5 for extensions). Assumption E.2 is mild since Gaussians have strong tail decay.

E.4 Omitted Details and Proofs

Lemma 11 of [68]. Our analysis (particularly Lemma 6.2) makes use of the following lemma to determine the posterior of an unknown parameter given several independent Gaussian observations.

Lemma E.3 (Lemma 11 of [68])**.** *Let $\theta \in \mathbb{R}$ be a constant (non-informative prior). Let $\{\phi_k \triangleq \theta + z_k\}_{k\in[K]}$ with $z_k \sim \mathcal{N}(0, \sigma_k^2)$ be m independent noisy observations of θ with variances $\{\sigma_k^2\}_{k\in[K]}$. Then, conditioned on $\{\phi_k\}_{k\in[K]}$, with $\sigma_\theta^2 \triangleq (\sum_{k\in[K]} 1/\sigma_k^2)^{-1}$, we have*

$$\theta = \sigma_\theta^2 \sum_{k\in[K]} \frac{\phi_k}{\sigma_k^2} + z, \text{ where } z \sim \mathcal{N}(0, \sigma_\theta^2). \tag{19}$$

This lemma allows us to express the local true centers w_k conditioned on the local empirical estimates $\hat{w}_k$, the external empirical estimates $\hat{w}_{\setminus k}$, and the local datasets $\{X_k\}_{k\in[K]}$.

Below we present omitted proofs for the main results presented in Section §6. We restate the lemmas and theorems for convenience.

Lemma 6.1. The minimizer of the silo-specific objective $h_k(w)$ (see Eq. (3), Eq. (18)) is

$$\hat{w}_k(\lambda) = \alpha \cdot \hat{w}_k + (1-\alpha)\cdot \hat{w}_{\setminus k}, \quad \text{where} \quad \alpha \triangleq \frac{K+\lambda}{(1+\lambda)K} \in (1/K, 1]. \tag{20}$$

Proof of Lemma 6.1. First note that $\bar{w}$ is independent of λ. Since we assume F_k is convex, h_k is also convex, and the proof follows from taking the derivative of h_k and setting it to 0:

$$\frac{\partial h_k(w)}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}(w - x_{k,i}) + \lambda(w-\bar{w}) = (w-\hat{w}_k) + \lambda(w-\bar{w}), \tag{21}$$

$$\hat{w}_k(\lambda) = \frac{1}{1+\lambda}\hat{w}_k + \frac{\lambda}{1+\lambda}\bar{w} = \frac{K+\lambda}{(1+\lambda)K}\cdot\hat{w}_k + \frac{\lambda(K-1)}{(1+\lambda)K}\hat{w}_{\setminus k}. \tag{22}$$

Note here that in the non-private case, the local estimator is given by the empirical average: $\hat{w}_k = \frac{1}{n}\sum_{i=1}^{n} x_{k,i}$. In the private case, under Assumption E.2, the local estimator is $\hat{w}_k = \frac{1}{n}\left(\xi_k + \sum_{i=1}^{n} x_{k,i}\right)$ where $\xi_k \sim \mathcal{N}(0, \sigma_{\text{DP}}^2)$ is the one-shot privacy noise, and we can similarly arrive at the same $\hat{w}_k(\lambda)$ expression. □

Lemma 6.2. Let $\sigma_{\text{loc}}^2 \triangleq \sigma^2/n + \sigma_{\text{DP}}^2/n^2$ denote the local variance on each silo as a result of data sampling and DP noise. Then, given $\hat{w}_k$ and $\hat{w}_{\setminus k}$, w_k can be expressed as

$$w_k = \mu_k + \zeta_k \tag{23}$$

$$\text{where} \quad \zeta_k \sim \mathcal{N}(0, \sigma_w^2), \tag{24}$$

$$\text{with} \quad \sigma_w^2 \triangleq \left(\frac{1}{\sigma_{\text{loc}}^2} + \frac{K-1}{K\tau^2+\sigma_{\text{loc}}^2}\right)^{-1} \tag{25}$$

$$\text{and} \quad \mu_k \triangleq \sigma_w^2\left(\frac{1}{\sigma_{\text{loc}}^2}\cdot\hat{w}_k + \frac{K-1}{K\tau^2+\sigma_{\text{loc}}^2}\cdot\hat{w}_{\setminus k}\right). \tag{26}$$

Proof of Lemma 6.2. When given θ, we can express $w_k = \theta + z_k$ where $z_k \sim \mathcal{N}(0,\tau^2)$, and by symmetry when given w_k,

$$\theta = w_k + z_k \quad \text{where} \quad z_k \sim \mathcal{N}(0,\tau^2). \tag{27}$$

Note also that when given the local dataset X_k, $\hat{w}_k$ is a noisy observation of w_k with Gaussian noise from data sampling and added privacy:

$$\hat{w}_k = w_k + \hat{z}_k \quad \text{where} \quad \hat{z}_k \sim \mathcal{N}(0, \sigma_{\text{loc}}^2). \tag{28}$$

Moreover, when given the local datasets $\{X_j\}_{j\in[K], j\neq k}$, $\hat{w}_{\setminus k}$ can be viewed as a noisy observation of θ with Gaussian noise from the silo heterogeneity and the empirical mean of the local estimators:

$$\hat{w}_{\setminus k} = \theta + \hat{z}_{\setminus k} \quad \text{where} \quad \hat{z}_{\setminus k} \sim \mathcal{N}\left(0, \frac{\tau^2+\sigma_{\text{loc}}^2}{K-1}\right). \tag{29}$$

Combining (29) with (27) we have, when given w_k and the local datasets $\{X_j\}_{j\in[K], j\neq k}$,

$$\hat{w}_{\setminus k} = w_k + z_{\setminus k} \quad \text{where} \quad z_{\setminus k} \sim \mathcal{N}\left(0, \tau^2 + \frac{\tau^2+\sigma_{\text{loc}}^2}{K-1}\right) \equiv \mathcal{N}\left(0, \frac{K\tau^2+\sigma_{\text{loc}}^2}{K-1}\right). \tag{30}$$

Invoking Lemma E.3 on Eq. (28) and Eq. (30) we have the desired μ_k and σ_w^2.

A key observation for this derivation is that the local datasets $\{X_k\}_{k\in[K]}$ form the Markov blankets of $\hat{w}_k$ and $\hat{w}_{\setminus k}$ (as they are sampled *after* w_k are sampled, and the estimators $\hat{w}_k$ and $\hat{w}_{\setminus k}$ are computed based on the datasets). Thus, given $\{X_k\}_{k\in[K]}$, $\hat{w}_k$ and $\hat{w}_{\setminus k}$ can be viewed as independent observations of w_k and Lemma E.3 applies. □

Theorem 6.3 (Optimal MR-MTL estimate). Let λ^* be the optimal λ that minimizes the generalization error. Then,

$$\lambda^* = \operatorname*{argmin}_{\lambda} \mathbb{E}\left[(w_k - \hat{w}_k(\lambda))^2 \mid \hat{w}_k, \hat{w}_{\setminus k}, \{X_k\}_{k\in[K]}\right] = \frac{\sigma_{\text{loc}}^2}{\tau^2} = \frac{1}{n\tau^2}\left(\sigma^2 + \frac{\sigma_{\text{DP}}^2}{n}\right). \tag{31}$$

Proof of Theorem 6.3. The objective of finding λ^* equates to finding an expression for λ such that when w_k is viewed as a noisy observation of $\hat{w}_k(\lambda)$ (which is an interpolation of $\hat{w}_k$ and $\hat{w}_{\setminus k}$ from Lemma 6.1), the coefficients for $\hat{w}_k$ and $\hat{w}_{\setminus k}$ in $\hat{w}_k(\lambda)$ matches those of w_k (Lemma 6.2). That is, we want λ^* such that

$$\frac{K+\lambda^*}{(1+\lambda^*)K} = \frac{\sigma_w^2}{\sigma_{\text{loc}}^2} = \frac{1}{\sigma_{\text{loc}}^2}\cdot\left(\frac{1}{\sigma_{\text{loc}}^2} + \frac{K-1}{K\tau^2+\sigma_{\text{loc}}^2}\right)^{-1}. \tag{32}$$

Simpifying gives $\lambda^* = \sigma_{\text{loc}}^2/\tau^2$. Note that $\hat{w}_k(\lambda^*)$ is the (conditional) MMSE estimator of w_k. □

Note that Eq. (31) measures the *generalization* error because w_k is the (unobserved) true center of the local data distribution around which testing data points would be drawn.

Proposition 6.5 (Optimal error gap to local training). Denote the error of the local estimate as

$$\mathcal{E}_{\text{loc}} \triangleq \mathbb{E}\left[(w_k - \hat{w}_k)^2 \mid X_k\right], \tag{33}$$

and let $\Delta_{\text{loc}} \triangleq \mathcal{E}_{\text{loc}} - \mathcal{E}^*$ be its error gap to the optimal estimate. Then,

$$\Delta_{\text{loc}} = \left(1 - \frac{1}{K}\right)\cdot\frac{\sigma_{\text{loc}}^4}{\sigma_{\text{loc}}^2+\tau^2}. \tag{34}$$

Proposition 6.6 (Optimal error gap to FedAvg). Denote the error of the global (FedAvg) estimate as

$$\mathcal{E}_{\text{fed}} \triangleq \mathbb{E}\left[(w_k - \bar{w})^2 \mid \{X_k\}_{k\in[K]}\right], \tag{35}$$

and let $\Delta_{\text{fed}} \triangleq \mathcal{E}_{\text{fed}} - \mathcal{E}^*$ be its error gap to the optimal estimate. Then,

$$\Delta_{\text{fed}} = \left(1 - \frac{1}{K}\right)\cdot\frac{\tau^4}{\sigma_{\text{loc}}^2+\tau^2}. \tag{36}$$

Proofs of Propositions 6.5 and 6.6. From Corollary 6.4 we know that the error of the optimal estimator $\hat{w}_k(\lambda^*)$ of w_k is $\mathcal{E}^* = \sigma_w^2 = \frac{\sigma_{\text{loc}}^2(\sigma_{\text{loc}}^2+K\tau^2)}{K(\sigma_{\text{loc}}^2+\tau^2)}$. For $\mathcal{E}_{\text{loc}}$, we know from earlier that the local estimator $\hat{w}_k = \hat{w}_k(0)$ of w_k has an MSE/variance of $\mathcal{E}_{\text{loc}} = \sigma_{\text{loc}}^2$. For $\mathcal{E}_{\text{fed}}$, we can view $\bar{w}$ as a noisy observation of w_k with MSE/variance $\mathcal{E}_{\text{fed}} = \frac{\sigma_{\text{loc}}^2+(K-1)\tau^2}{K}$. The results of Propositions 6.5 and 6.6 thus follow from re-arranging terms of $\mathcal{E}_{\text{loc}} - \mathcal{E}^*$ and $\mathcal{E}_{\text{fed}} - \mathcal{E}^*$ respectively. □

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,

$$\mathcal{E}_\lambda = \left(1 - \frac{1}{K}\right)\frac{\sigma_{\text{loc}}^2+\lambda^2\tau^2}{(\lambda+1)^2} + \frac{\sigma_{\text{loc}}^2}{K}. \tag{37}$$

Proof of Lemma 6.7. The result follows from the variance of $\hat{w}_k(\lambda)$ when viewed as a noisy observation of w_k. Specifically, from Lemma 6.1 we know that $\hat{w}_k(\lambda)$ is an interpolation (parameterized by $\alpha \triangleq \frac{K+\lambda}{(1+\lambda)K}$) between $\hat{w}_k$ and $\hat{w}_{\setminus k}$. We thus have (with some rearrangement),

$$\mathcal{E}_\lambda = \alpha^2\cdot\sigma_{\text{loc}}^2 + (1-\alpha)^2\cdot\frac{K\tau^2+\sigma_{\text{loc}}^2}{K-1} = \left(1 - \frac{1}{K}\right)\frac{\sigma_{\text{loc}}^2+\lambda^2\tau^2}{(\lambda+1)^2} + \frac{\sigma_{\text{loc}}^2}{K}. \tag{38}$$

□

The proof of Theorem 6.8 follows directly from Lemma 6.7 by subtracting the common terms between private and non-private MR-MTL estimates.

E.5 The Case of Heterogeneous Privacy Requirements

In the main analysis, we assumed for simplicity that each silo has the same values of local dataset size n, local data sampling variance σ^2, and DP noise variance σ_{DP}^2. In this section, we consider the

case where each silo k has custom values for the above, denoted as n_k, σ_k^2, and $\sigma_{k,\mathrm{DP}}^2$, to arrive at a silo-specific "local variance" $\tilde{\sigma}_k^2$:

$$\tilde{\sigma}_k^2 \triangleq \frac{\sigma_k^2}{n} + \frac{\sigma_{k,\mathrm{DP}}^2}{n^2}. \tag{39}$$

With a slight abuse of notation, let us use $C + \mathcal{N}(\mu, \sigma^2)$ to denote drawing an iid sample of Gaussian noise with mean μ and variance σ^2 and add it to some value C. Then, following Lemma 6.2 and the same set of conditions, we can express the local model $\hat{w}_k$ and the non-local model $\hat{w}_{\setminus k}$ as:

$$\hat{w}_k = w_k + \mathcal{N}(0, \tilde{\sigma}_k^2), \tag{40}$$

$$\hat{w}_{\setminus k} = \theta + \mathcal{N}\left(0, \frac{\tau^2}{K-1} + \frac{\sum_{j \neq k} \tilde{\sigma}_j^2}{(K-1)^2}\right) \tag{41}$$

$$= w_k + \mathcal{N}\left(0, \tau^2 + \frac{\tau^2}{K-1} + \frac{\sum_{j \neq k} \tilde{\sigma}_j^2}{(K-1)^2}\right) \tag{42}$$

$$= w_k + \mathcal{N}\left(0, \frac{K\tau^2}{K-1} + \frac{\sum_{j \neq k} \tilde{\sigma}_j^2}{(K-1)^2}\right), \tag{43}$$

and the true local center w_k can be expressed in terms of $\hat{w}_k$ and $\hat{w}_{\setminus k}$ with silo-specific local variances:

Lemma E.4. *Given $\hat{w}_k$ and $\hat{w}_{\setminus k}$, w_k can be expressed as*

$$w_k = \bar{\mu}_k + \mathcal{N}(0, \bar{\sigma}_k^2) \tag{44}$$

with

$$\bar{\sigma}_k^2 \triangleq \left(\frac{1}{\tilde{\sigma}_k^2} + \frac{(K-1)^2}{(K-1)K\tau^2 + \sum_{j \neq k} \tilde{\sigma}_j^2}\right)^{-1}, \tag{45}$$

$$\bar{\mu}_k \triangleq \frac{\bar{\sigma}_k^2}{\tilde{\sigma}_k^2} \cdot \hat{w}_k + \frac{(K-1)^2 \bar{\sigma}_k^2}{(K-1)K\tau^2 + \sum_{j \neq k} \tilde{\sigma}_j^2} \cdot \hat{w}_{\setminus k}. \tag{46}$$

Proof. The proof follows similarly from Lemma 6.2: we apply Lemma E.3 to the expressions of $\hat{w}_k$ and $\hat{w}_{\setminus k}$ as noisy observations of w_k. □

From here we can derive the best λ_k^* *specific to each silo*.

Theorem E.5 (Optimal λ for silo k)**.** *The optimal choice of the* MR-MTL *regularization strength for silo k is given by*

$$\lambda_k^* = \frac{\tilde{\sigma}_k^2}{\tau^2 + \frac{1}{K}\left(\frac{\sum_{j \neq k} \tilde{\sigma}_j^2}{K-1} - \tilde{\sigma}_k^2\right)} \tag{47}$$

Proof. The result follows from re-arranging and solving for λ_k^* in the following equation, as in Theorem 6.3:

$$\frac{K + \lambda_k^*}{(1 + \lambda_k^*)K} = \frac{\bar{\sigma}_k^2}{\tilde{\sigma}_k^2} \tag{48}$$

where μ_k is defined in Lemma E.4. □

Theorem E.5 suggests that if silos have varying local variance (as a result of varying local dataset sizes, inherent data variances, and silo-specific sample-level DP requirements), then it is optimal to have each silo choose its own λ_k^*. In particular, the term $\frac{\sum_{j \neq k} \tilde{\sigma}_j^2}{K-1} - \tilde{\sigma}_k^2$ suggests that the optimal λ_k^* depends on how much of an "outlier" is silo k compared to the rest of silos: if it has a smaller local variance, then it benefits from a smaller λ_k^* since other silos are noisy; if it has a larger local variance, then λ_k^* would be larger to encourage silo k to "conform". Note that under this simple analysis, one could have $\lambda_k^* < 0$ (when $\tilde{\sigma}_k^2 > K\tau^2 + \frac{1}{K-1}\sum_{j \neq k} \tilde{\sigma}_j^2$); this case stems from having a gradually larger $\tilde{\sigma}_k^2$ and it suffices to choose $\lambda^* \to \infty$ (i.e. fall back to FedAvg training).

F Additional Details for Private Hyperparameter Tuning (§7)

In Section §7 we took an alternative view on the benefits of personalization by examining the *privacy cost* of tuning the hyperparameter responsible for navigating the tradeoff between costs from heterogeneity and privacy. In this section, we expand on the omitted details and provide more discussions.

F.1 Additional Background on Privacy Costs of Hyperparameter Tuning

We consider the typical hyperparameter tuning procedure (denoted as HPO) where a base algorithm M, such as one run of a machine learning algorithm to produce a model, is executed h times with different hyperparameters and the best result is recorded. The hyperparameters may be high-dimensional; e.g. vector of learning rate and batch size. Typically, h is a constant (e.g. we sweep a fixed grid of learning rates and batch sizes).

On a high level, the works of Liu and Talwar [62] and Papernot and Steinke [79] show that, with a constant h, one can construct M where the privacy guarantee of the HPO output is (roughly) a factor of h weaker than the original privacy guarantee. Specifically, [79] gives the following proposition that applies to both pure DP and Rényi DP (and thus related to approximate DP).

Proposition F.1 (Proposition 17 of [79])**.** *For any $\varepsilon > 0$ and any $\alpha > 1$, there exist some algorithm M that satisfy $(\varepsilon, 0)$-DP and the corresponding HPO algorithm A that runs M for h times and returns only the best result, such that*

1. *A is not $(\tilde{\varepsilon}, 0)$-DP for any $\tilde{\varepsilon} < h\varepsilon$, and*
2. *A is not $(\alpha, \tilde{\varepsilon}(\alpha))$-Rényi DP for any $\tilde{\varepsilon}(\alpha) < \hat{\varepsilon}(\alpha)$ where*
$$\hat{\varepsilon}(\alpha) = h\varepsilon - \frac{h\log(1+\exp(-\varepsilon))}{\alpha - 1}. \tag{49}$$

Both the pure DP and RDP results suggest that running the typical HPO procedure with a fixed h can be as bad as running M and h times and releasing *all* results.

The authors of [62] and [79] provided a mitigating strategy where one simply makes h (the number of tuning runs) a random variable. The authors of [79] provide an improved analysis and considered two distributions for h in particular: the truncated negative binomial (TNB) distribution and the Poisson distribution.

Here, we focus on TNB as it provides a spectrum of results for privacy-utility tradeoff, allowing us to choose several points on this spectrum that offer small (but realistic) privacy overhead. We combine it our analysis in Section §6 to motivate our discussions on the practical complications of deploying personalization methods.

Specifically, the probability mass function of the TNB distribution is given by

$$f(h) = \begin{cases} \frac{(1-\gamma)^h}{\gamma^{-\eta}-1}\prod_{\ell=0}^{h-1}\left(\frac{\ell+\eta}{\ell+1}\right) & \text{for } \eta \neq 0, \\ \frac{(1-\gamma)^h}{h\log(1/\gamma)} & \text{for } \eta = 0, \end{cases} \text{ with expected value } \mathbf{E}[h] = \begin{cases} \frac{\eta(1-\gamma)}{\gamma(1-\gamma^\eta)} & \text{for } \eta \neq 0, \\ \frac{1/\gamma-1}{\log(1/\gamma)} & \text{for } \eta = 0, \end{cases}$$

where η and γ are parameters of the TNB distribution. The privacy of the HPO output using h sampled from the TNB distribution is given by the following theorem from [79].

Theorem F.2 (Theorem 2 of [79])**.** *For any $\eta > 1$, any $\gamma \in (0, 1)$, and any algorithm M that satisfy both $(\alpha_1, \varepsilon(\alpha_1))$-RDP and $(\alpha_2, \varepsilon(\alpha_2))$-RDP with $\alpha_1 > 1$ and $\alpha_2 \geq 1$, the HPO algorithm A that runs M for h times and returns only the best result with $h \sim \mathrm{TNB}(\eta, \gamma)$ satisfies $(\alpha_1, \tilde{\varepsilon})$-RDP with*
$$\tilde{\varepsilon} = \varepsilon(\alpha_1) + (1+\eta)\left(1 - \frac{1}{\alpha_2}\right)\varepsilon(\alpha_2) + \frac{(1+\eta)\log(1/\gamma)}{\alpha_2} + \frac{\log \mathbf{E}[h]}{\alpha_1 - 1}. \tag{50}$$

Theorem F.2 tells us that by drawing h from the TNB distribution, the privacy cost of the HPO procedure is a *constant* factor of the original cost of M for pure DP (when $\alpha_1, \alpha_2 \to \infty$), or up to a factor of $\log \mathbf{E}[h]$ for approximate DP. This substantially improves the default HPO procedure.

F.2 Interpretation of Fig. 7

In Fig. 7, we considered 6 pairs of (η, γ) to arrive at the same $\mathbf{E}[h] = 10$ (i.e. on average we want to try 10 values of λ when tuning MR-MTL). Intuitively, for a fixed $\mathbf{E}[h]$, we can expect a smaller η (and consequently a smaller γ) to give a tighter privacy guarantee from Theorem F.2 (i.e. the privacy cost of hyperparameter tuning is smaller). At the same time, however, the TNB distribution would be more concentrated around $h = 1$ with a smaller η, which means we may end up trying only 1 hyperparameter even when $\mathbf{E}[h]$ is large. (η, γ) thus provides an implicit privacy-utility tradeoff.

In Fig. 7 we chose a list of η values that favors strong privacy while being realistic, since picking $\eta \to -1$ (or just $\eta < -0.5$) would give a even smaller privacy overhead, but it will perform very poorly utility-wise and was not considered by [79]. We then applied Theorem F.2 with these (η, γ) values to the the expected error of MR-MTL as a function of λ under mean estimation, by computing and applying the extra noise needed to account for the privacy cost of tuning. Here, the closed form of the expected error of MR-MTL is derived in Lemma 6.7, but in Fig. 7 we ran numerical simulations with 500 repetitions.

Fig. 7 (a) and (c) suggests that there are settings in which, after applying Theorem F.2, the best λ^* one could find would already give an error that is larger than simply running FedAvg and local training *without any tuning*, respectively (the **green** curves). Fig. 7 (b) suggests that in less extreme cases, the private hyperparameter tuning cost would significantly diminish the utility improvement of MR-MTL.

F.3 Implementing Hyperparameter Tuning under Silo-Specific Sample-Level DP

Under silo-specific sample-level privacy, it can be intricate to implement the improved, randomized hyperparameter tuning protocol described above in Appendix F.1. For silo k to satisfy its own $(\varepsilon_k, \delta_k)$ sample-level DP target, it must draw h *independently* to all other silos, and it must only return the best result of hyperparameter tuning at the *end of training* (instead of returning the best model iterate at every round).

Using MR-MTL as an example, the above has several implications:

1. Each silo should now maintain a list of personalized models, each corresponding to a choice of λ.
2. Depending on the specific distribution of h, silos may or may not end up trying the same set of λ values.
3. During training rounds, silos must return model updates for – and consequently regularize their personalized models towards – a "pivot" model using some public $\bar{\lambda}$.

The analysis in Appendix E.5 suggests that having silo-specific choices of λ (#2 above) should not have adverse effects, but the above adaptations in general may influence final utility or convergence properties in iterative learning settings.

Other more sophisticated implementation (potentially tailored to the specific personalization method) may also be possible. It would be an interesting future direction to study the best approaches for implementing private hyperparameter tuning under silo-specific sample-level privacy.

G Future Directions for Auto-tuning / Online Estimation of λ

Apart from tuning λ for MR-MTL (or other hyperparameters for other personalization methods) non-adaptively with grid search or random search, another approach is to leverage some form of online estimation as in [97, 8, 80] (mentioned in Section §7). That is, λ is adjusted at every training round or iteration such that it gradually converges to a good λ^* (in expectation).

This line of approaches would be promising if their privacy overhead (if any) can be contained within a small factor of the original privacy guarantee—smaller than that offered by the randomized tuning procedure discussed above in Appendix F.1. However, since the hyperparameters of interest are those that navigate the tradeoff between costs from heterogeneity and privacy (such as λ for MR-MTL), there are several factors that make auto-tuning particularly challenging in our setting:

- **A universal and practical measure of heterogeneity is generally unavailable**. For MR-MTL, our analysis in Section §6 suggests that the optimal λ^* depends on a measure of data heterogeneity across silos (τ^2). However, in practice, the level of heterogeneity is abstract and hard to quantify even with non-private oracle access to the silo datasets. Past work has devised custom measures of heterogeneity such as the variance of client (silo) model or gradient updates [52, 59], or the difference between the true global loss and the aggregate of the true local losses, but these measures may not be consistent throughout training [52, 59] such that they cannot provide a useful estimate of the heterogeneity which should be invariant during training. These metrics may also simply not be obtainable in practice [60]. The work of [91] considers the use of a client-relationship matrix, but such matrix may be too large (K^2 elements) and too noisy under privacy as we may need to directly or implicitly noise every coordinate of the matrix. Directly estimating the dataset statistics may also be challenging, since silos may apply custom dataset transformations (e.g. standardization or augmentation) that can drastically alter such statistics without affecting the learning dynamics.
- **The estimated heterogeneity may have high variance.** In addition to the above, heterogeneity would generally be measured on the client distribution (as some form of "variance" across clients, as in [52]), but there is generally a limited number of clients in cross-silo settings, meaning that this estimated "variance" as a measure of heterogeneity may itself have high variance. Moreoever, such measures must also be estimated privately, which means that it would have even higher variance from the additional noise. This could be problematic since the best hyperparameter for personalization may depend on such heterogeneity measure (as is the case for MR-MTL in simplified settings), and the precision of a potential auto-tuning precedure may suffer as a result.
- **Auto-tuning procedures may need to be developed specifically for each method.** For example, local finetuning may use the number of federated training rounds as the hyperparameter for determining "how much" to personalize, but such hyperparameter would likely be auto-tuned in a drastically different way compared to auto-tuning for λ for MR-MTL. It is in general an open question as to whether method-specific auto-tuning procedures will always outperform the simple randomized procedure described in Appendix F.1, in terms of both precision and privacy overhead.

H Additional Experimental Results

H.1 Local Finetuning (Extension of Fig. 2)

In Fig. 2, we made the observation that local finetuning [99, 104, 19] as a personalization strategy (warm-starting from a global model and continue training using local data) may not always improve utility as expected. Extending on Fig. 2, we examine the behavior of local finetuning starting at different stages of training (under a fixed total number of rounds) as well as under varying privacy budgets in Fig. A2. We observe that the phenomenon illustrated in Fig. 2 is indeed reflected across different settings: as soon as local finetuning begins, the DP utility gap can quickly widen, and the utility of finetuning under privacy can roughly reduce to the utility of local training.

Note, however, that these observations serve to give insights into an interesting behavior of local finetuning under silo-specific sample-level DP and *do not* preclude the possibility that local finetuning can still outperform both local training and FedAvg. Indeed, local finetuning was observed to outperform local training in, e.g., Fig. 3 (d) and Fig. A8.

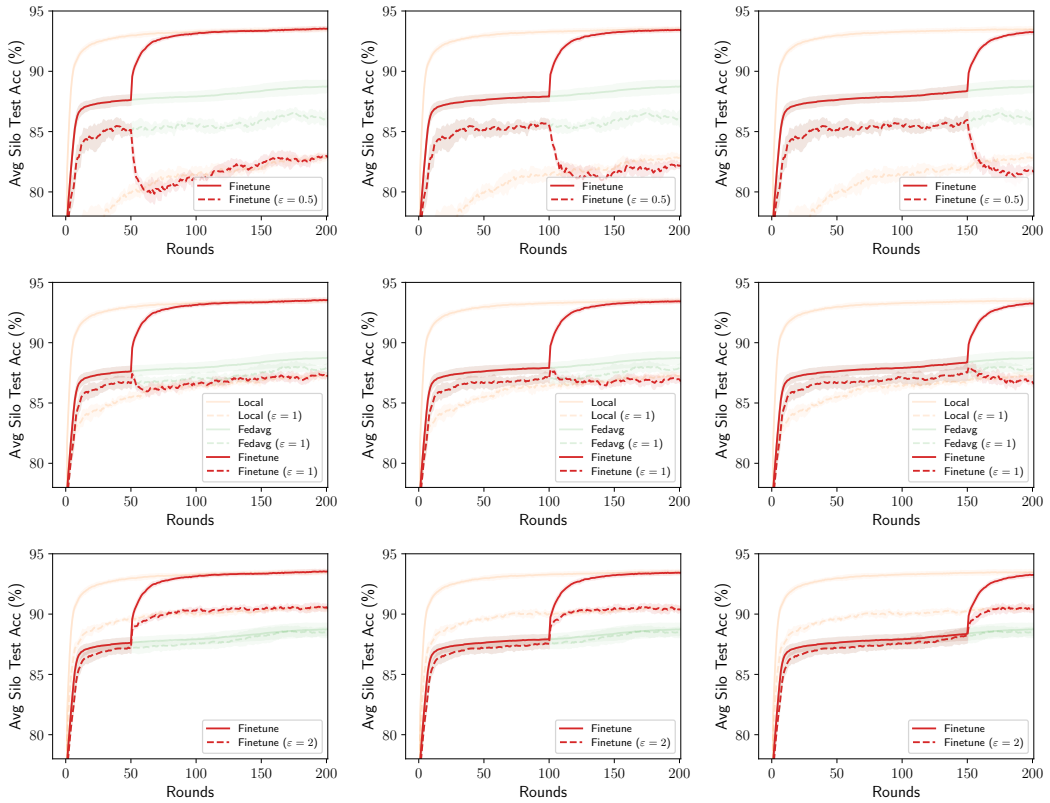

Figure A2: (Extension of Fig. 2) **Behavior of local finetuning** starting at different stages of training (25%, 50%, 75%) and under varying privacy budgets ($\varepsilon \in [0.5, 1, 2], \delta = 10^{-7}$) for $T = 200$ rounds on the **Vehicle** dataset.

H.2 Utility of MR-MTL as a Function of λ (Extension of Fig. 5)

We also extend on Fig. 5 to consider varying privacy budgets, and the results are shown in Fig. A3 (Vehicle), Fig. A4 (School), and Fig. A5 (Heterogeneous CIFAR-10). Note that on School, the test metric is MSE thus lower is better. There are several notable observations:

- λ^* **decreases as** ε **grows (weaker privacy)**: With weaker privacy (larger ε), we have a smaller optimal λ^* (i.e. the location of the utility "bump" under DP gradually shifts to the left). This behavior is characterized by Theorem 6.3, where under stronger privacy, silos benefit from more federation as a means to reduce the effect of privacy noise.

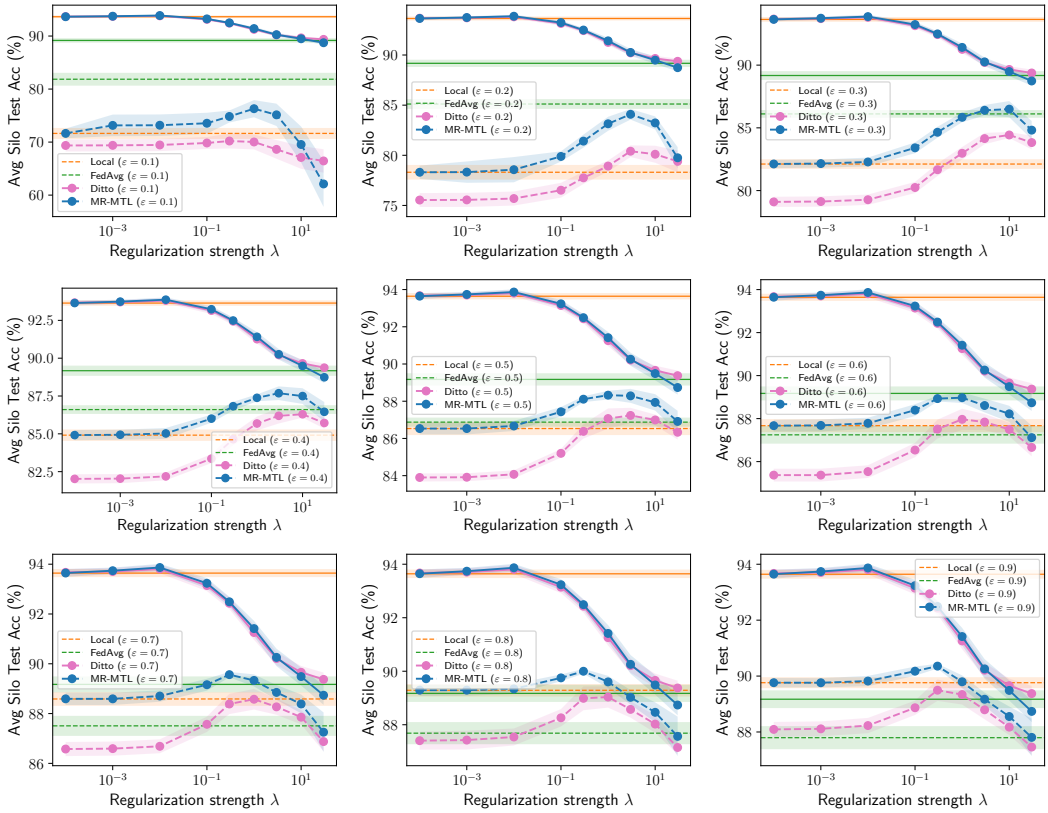

Figure A3: (Extension of Fig. 5) **Behavior of MR-MTL as a function of λ across varying privacy budgets** ($\varepsilon \in [0.1, 0.2, ..., 0.9]$, $\delta = 10^{-7}$) for $T = 400$ rounds on the **Vehicle** dataset. Solid lines refer to the non-private runs (same across all plots).

- **MR-MTL does not always outperform FedAvg** (at high privacy regimes): One minor caveat is that Proposition 6.6 says that the utility gap from MR-MTL to FedAvg is always nonnegative under federated mean estimation; however, as discussed in Sections §5 and §6, MR-MTL does not approach FedAvg with larger λ in general learning settings since the MR-MTL objective may become too hard to solve via (DP-)SGD. Thus, despite its utility advantage over local training at λ^*, it may never reach the performance of FedAvg. See, e.g., subplots of $\varepsilon \in [0.1, 0.2]$ in Fig. A3.
- **Utility advantage of MR-MTL (λ^*) over local/FedAvg changes with ε**: Observe that with larger ε, the utility advantage of MR-MTL (λ^*) is larger compared to FedAvg and is smaller compared to local training. This behavior is characterized by Propositions 6.5 and 6.6.

Note also that for Heterogeneous CIFAR-10 (Fig. A5), Ditto [57] exhibits a slightly different interpolation behavior compared to MR-MTL (e.g. it has larger optimal λ^* under privacy), though at λ^* its utility underperforms that of MR-MTL under privacy.

H.3 Subsampled ADNI Dataset

We further evaluate the suite of personalization methods on the subsampled ADNI dataset, and the results are shown in Fig. A6. There are several notable observations:

1. MR-MTL is essentially recovering local training due to the high degree of heterogeneity present in this dataset, with $\lambda^* \approx 0$ for most ε values except around $\varepsilon \approx 3.5$ (annotated with arrows in Fig. A6 (a, b)).
2. Unlike results on other datasets, the privacy regime of interest where MR-MTL outperforms both local/FedAvg (around $\varepsilon \approx 4$) is extremely narrow, compared to $\varepsilon \approx 0.5$ for Vehicle (Fig. 3 (a)), $\varepsilon \approx 6$ for School (Fig. 3 (b)), and $\varepsilon \approx 1.5$ for GLEAM (Fig. 3 (c)). The utility advantage of MR-MTL is also insignificant.

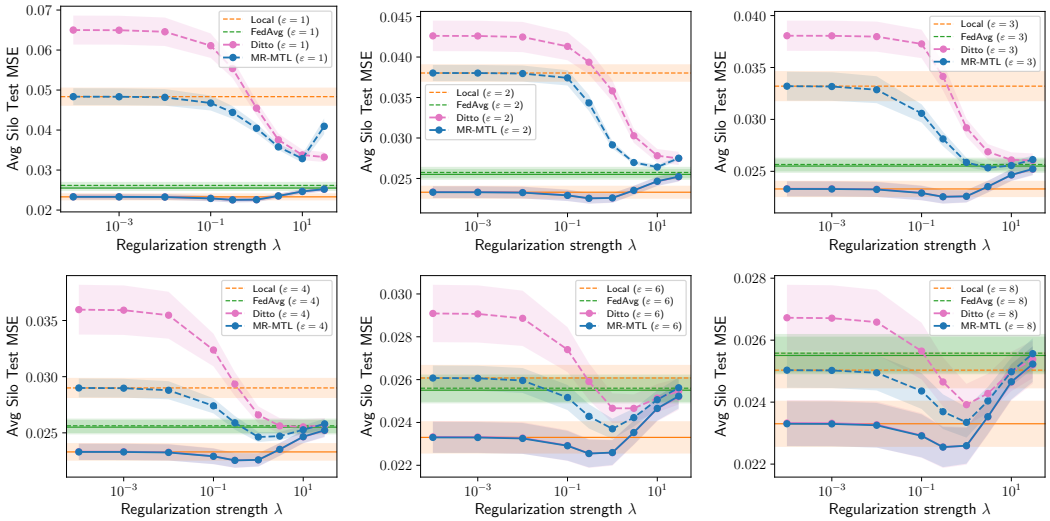

Figure A4: (Extension of Fig. 5) **Behavior of MR-MTL as a function of λ across varying privacy budgets** ($\varepsilon \in [1, 2, 3, 4, 6, 8], \delta = 10^{-3}$) for $T = 200$ rounds on the **School** dataset. Cf. Fig. 3 (b) and Fig. A8, where the privacy regime of interest is around $\varepsilon \approx 6$. Solid lines refer to the non-private runs (same across all plots).

3. The underlying heterogeneity structure of the dataset is rather pathological in that it is not ameanable to *both* mean-regularization and clustering. First, observe from Fig. A6 (a) that MR-MTL still opts for a small λ^* in high-privacy regimes even when FedAvg performs better; one would expect MR-MTL to opt for a larger λ for lower DP noise (at a cost of higher heterogeneity). Second, from Fig. A6 (b, c), we observe that both clustering and cluster-preconditoning did not lead to significant utility improvements, although the latter reduces the degree of heterogeneity and allows MR-MTL to opt for a larger λ^* in the privacy regime of interest.

In general, determining how to better model such heterogeneity (especially in high privacy regimes) is an interesting direction of future work.

H.4 Additional Discussions

Behavior on larger privacy budgets (Fig. A7). We extend the results on Vehicle (Fig. 3 (a, b)) by considering larger privacy budgets, and the results are shown in Fig. A7. We note that as privacy requirement loosens, personalized learning (where each silo maintains its own model) tend to perform better than the case where models are shared across clients (FedAvg and IFCA). This is expected as clients in cross-silo datasets tend to have sufficient data for learning a good local model. Moreover, this also means MR-MTL remains competitive as it can use a smaller λ.

Effect of dataset subsampling (Fig. A8). Subsampling local datasets would in principle turn the cross-silo learning setting closer to a cross-device learning setting, in which local training becomes less attractive as due to insufficient local data and may opt for federated training despite data heterogeneity. In Fig. A8, we examine this behavior on the School dataset by using different train/test split ratios of 80%/20%, 50%/50%, and 20%/80%. We observe that with less local training data: (1) FedAvg performs better since silos benefit from federation despite data heterogeneity; (2) the cross-over point between local training and FedAvg shifts to larger ε (or they may not be a cross-over point); and (3) MR-MTL may no longer provide an optimal point on the personalization spectrum, since small local datasets with silo-specific sample-level DP necessitate a larger λ (cf. Theorem 6.3) but MR-MTL may not recover FedAvg under (DP-)SGD. In these cases, client-level DP protection (as is commonly used for cross-device FL) may be more appropriate.

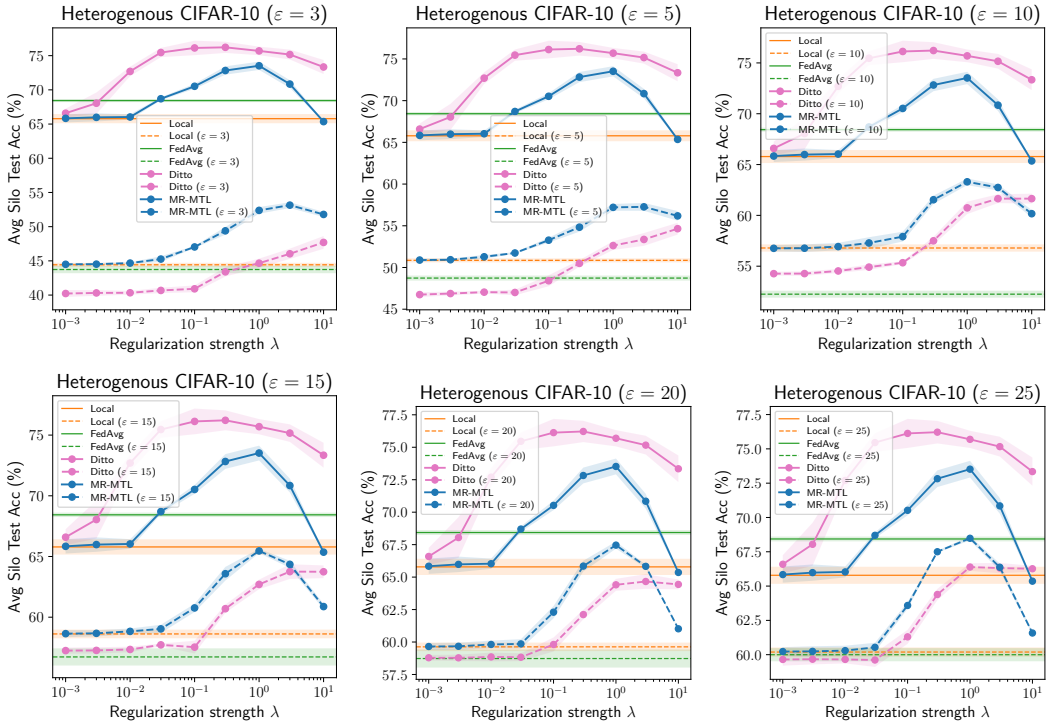

Figure A5: (Extension of Fig. 5) **Behavior of MR-MTL as a function of λ across varying privacy budgets** ($\delta = 10^{-4}$) for $T = 200$ rounds on **Heterogeneous CIFAR-10**. Solid lines refer to non-private runs (same across all plots) and dashed/dotted lines refer to private runs (with silo-specific sample-level DP).

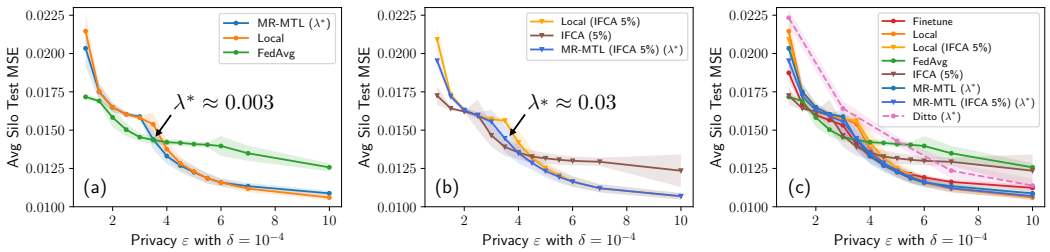

Figure A6: **Test accuracy** (mean $\pm$ std with 3 seeds) vs **privacy budgets** ε for various personalization methods on the **subsampled ADNI** dataset with $T = 500$ rounds. The heterogeneity present in this dataset is unique in that it is not ameanable to both mean-regularization (a) and clustering (b, c), though clustering can mitigate the level of heterogeneity by allowing a larger λ^* for MR-MTL. In such cases of high heterogeneity, local performance tends to be superior for most privacy budgets, and we find that MR-MTL recovers this behavior. Determining how to better model such heterogeneity (especially in high privacy regimes) is an interesting direction of future work.

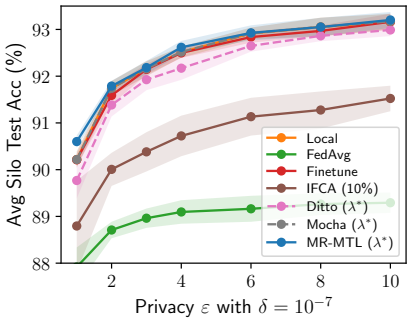

Figure A7: (Extension of Fig. 3 (b)) Results on the **Vehicle** dataset for low privacy regimes ($1 \leq \varepsilon \leq 10$). As privacy becomes weaker, the need for federation diminishes and personalization methods (including local training) perform better. MR-MTL can recover local training by using a small λ.

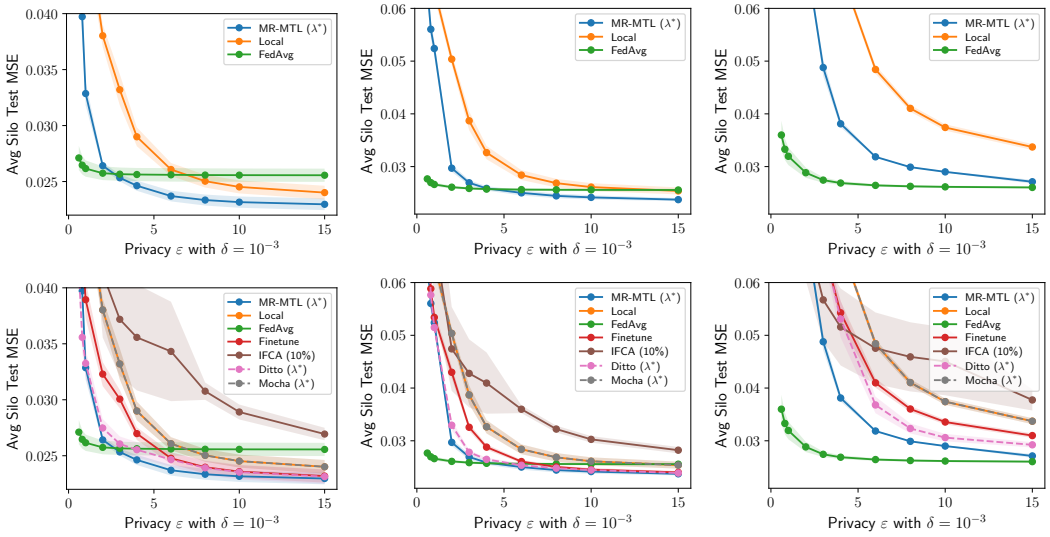

Figure A8: **Test MSE vs total privacy budgets on the School dataset** (80%/20%, 50%/50%, and 20%/80% train/test split of the local dataset by columns from left to right). The top row compares MR-MTL to local training/FedAvg, which form the endpoints of the personalization spectrum with constant privacy costs, and the bottom row compares against other personalization methods. Under data subsampling (as little as 20% training data in the 3rd column), we obtain a setting closer to cross-device FL where FedAvg outperforms personalization since clients benefit from others' training data despite their heterogeneity.