# DIFFERENTIALLY PRIVATE META-LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Parameter-transfer is a well-known and versatile approach for meta-learning, with applications including few-shot learning, federated learning, and reinforcement learning. However, parameter-transfer algorithms often require sharing models that have been trained on the samples from specific tasks, thus leaving the task-owners susceptible to breaches of privacy. We conduct the first formal study of privacy in this setting and formalize the notion of *task-global differential privacy as a practical relaxation of more commonly studied threat models*. We then propose a new differentially private algorithm for gradient-based parameter transfer that not only satisfies this privacy requirement but also retains provable transfer learning guarantees in convex settings. Empirically, we apply our analysis to the problems of federated learning with personalization and few-shot classification, showing that allowing the relaxation to task-global privacy from the more commonly studied notion of *local privacy* leads to dramatically increased performance in recurrent neural language modeling and image classification.

## 1 INTRODUCTION

The field of *meta-learning* offers promising directions for improving the performance and adaptability of machine learning methods. At a high level, the key assumption leveraged by these approaches is that the *sharing* of knowledge gained from individual learning tasks can help catalyze the learning of similar unseen tasks. However, the collaborative nature of this process, in which task-specific information must be sent to and used by a *meta-learner*, also introduces inherent data privacy risks.

In this work, we focus on a popular and flexible meta-learning approach, *parameter transfer* via *gradient-based meta-learning* (GBML). This set of methods, which includes well-known algorithms such as MAML (Finn et al., 2017) and Reptile (Nichol et al., 2018), tries to learn a common initialization $\phi$ over a set of tasks $t = 1, \ldots, T$ such that a high-performance model can be learned in only a few gradient-steps on new tasks. Notably, information flows constantly between training tasks and the meta-learner as learning progresses; to make iterative updates, the meta-learner obtains feedback on the current $\phi$ by training task-specific models $\hat{\theta}_t$ with it.

Meanwhile, in many settings it is crucial to ensure that sensitive information in each task-specific dataset stays private. Examples of this include learning models for next-word prediction on cell phone data (McMahan et al., 2018), clinical predictions using hospital records (Zhang et al., 2019), and fraud detectors for competing credit card companies (Stolfo et al., 1997). In such cases, each data-owner can benefit from information learned from other tasks, but each also desires, or is legally required, to keep their raw data private. Thus, it is not sufficient to learn a well-performing $\phi$; it is equally imperative to ensure that a task's sensitive information is not obtainable by *anyone* else.

While parameter transfer algorithms can move towards this goal by peforming task-specific optimization locally, thus preventing direct access to private data, this provision is far from fail-safe in terms of privacy. A wealth of work has shown in the single-task setting that it is possible for an adversary with only access to the model to learn detailed information about the training set, such as the presence or absence of specific records (Shokri et al., 2017) or the identities of sensitive features given other covariates (Fredrikson et al., 2015). Furthermore, Carlini et al. (2018) showed that deep neural networks can effectively memorize user-unique training examples, which can be recovered even after only a single epoch of training. As such, in parameter-transfer methods, the meta-learner or any downstream participant can potentially recover data from a previous task.

However, despite these serious risks, privacy-preserving meta-learning has remained largely an unstudied problem. Our work aims to address this issue by applying *differential privacy* (DP) (Dwork and Roth, 2014), a well-established definition of privacy with rich theoretical guarantees and consistent empirical success at preventing leakages of data (Carlini et al., 2018; Fredrikson et al., 2015; Jayaraman and Evans, 2019). Crucially, although there are various threat models and degrees of DP one could consider in the meta-learning setting (as we outline in Section 2), we balance the well-documented trade-off between privacy and model utility by formalizing and focusing on a setting that we call *task-global* DP. This setting provides a strong privacy guarantee for each task-owner that sharing $\hat{\theta}_t$ with the meta-learner will not reliably reveal anything about specific training examples to *any* downstream agent. It also allows us to use the framework of Khodak et al. (2019a) to provide a DP GBML algorithm that enjoys provable learning guarantees in convex settings.

Finally, we show an application of our work by drawing connections to federated learning (FL). While standard methods for FL, such as FedAvg (McMahan et al., 2017), have inspired many works also concerning DP in a multi-user setup (Agarwal et al., 2018; Duchi et al.; Geyer et al., 2018; McMahan et al., 2018; Truex et al., 2019), we are the first to consider task-global DP as a useful variation on standard DP settings. Moreover, these works fundamentally differ from ours in that they do not consider a task-based notion of learnability, instead focusing on the global federated learning problem to learn a single global model. That being said, a federated setting involving per-user personalization (Chen et al., 2018; Smith et al., 2017) is a natural meta-learning application.

More specifically, our main contributions are:

1. We are the first to provide a taxonomy for the different notions of DP possible for meta-learning. In particular, we formalize on a variant we call *task-global* DP, showing and arguing that it adds a useful option to commonly studied settings in terms of trading privacy and accuracy.
2. We propose the first DP GBML algorithm, which we construct to satisfy this privacy setting. Further, we show a straightforward extension for obtaining a *group DP* version of our setting to protect multiple samples simultaneously.
3. While our privacy guarantees hold generally, we also prove learning-theoretic results in convex settings. Our learning guarantees scale with task-similarity, as measured by the closeness of the task-specific optimal parameters (Denevi et al., 2019; Khodak et al., 2019b).
4. We show that our algorithm, along with its theoretical guarantees, naturally carries over to federated learning with personalization. Compared to previous notions of privacy considered in works for DP federated learning (Agarwal et al., 2018; Bhowmick et al., 2019; Geyer et al., 2018; McMahan et al., 2018; Truex et al., 2019), we are, to the best of our knowledge, the first to simultaneously provide both privacy and learning guarantees.
5. Empirically, we demonstrate that our proposed privacy setting allows for strong performance on federated language-modeling and few-shot image classification tasks. For the former, we achieve close to the performance of non-private models and significantly improve upon the performance of models trained with local-DP guarantees, a previously studied notion that also provides protections against the meta-learner. Our setting reasonably relaxes this latter notion but can achieve roughly $2.4$ times the accuracy on a modified version of the Shakespeare dataset (Caldas et al., 2018) and $2.7$ times the accuracy on a modified version of Wiki-3029 (Arora et al., 2019). For image-classification, we show that we show that we can still retain significant benefits of meta-learning while applying task-global DP on Omniglot (Lake et al., 2011).

### 1.1 Related Work

**DP Algorithms in Federated Learning Settings.** Works most similar to ours focus on providing DP for federated learning. Specifically, Geyer et al. (2018) and McMahan et al. (2018) apply update clipping and the Gaussian Mechanism to achieve user-level global DP federated learning algorithms for language modeling and image classification tasks respectively. Their methods are shown to only suffer minor drops in accuracy compared to non-private training but they do not consider protections to inferences made by the meta-learner. Alternatively, Bhowmick et al. (2019) does achieve such protection by applying a theoretically rate-optimal local DP mechanism on the $\hat{\theta}_t$'s users send to the meta-learner. However, they sidestep hard minimax rates (Duchi et al.) by assuming the central server has limited side-information and allowing for a large privacy budget. In this work, though we achieve a relaxation of the privacy of Bhowmick et al. (2019), we do not restrict the adversary's

power. Finally, Truex et al. (2019) does consider a setting that coincides with task-global DP, but they focus primarily on the added benefits of applying MPC (see below) rather than studying the merits of the setting in comparison to other potential settings.

**Secure Multiparty Computation (MPC).** MPC is a cryptographic technique that allows parties to calculate a function of their inputs while also maintaining the privacy of each individual inputs. In GBML, sets of model updates may come in a batch from multiple tasks, and hence MPC can securely aggregate the batch before it is seen by the meta-learner. Though MPC itself gives no DP guarantees, it can combined with DP to increase privacy. This approach has been studied in the federated setting, e.g. by Agarwal et al. (2018), who apply SMC in the same difficult setting of Bhowmick et al. (2019), and Truex et al. (2019), who apply SMC similarly to a setting analogous to ours. On the other hand, MPC also comes with additional practical challenges such as peer-to-peer communication costs, drop outs, and vulnerability to collaborating participants. As such, combined with its applicability to multiple settings, including ours, we consider MPC to be an orthogonal direction.

## 2 Privacy in a Meta-Learning Context

In this section, we first formalize the meta-learning setting that we consider. We then describe the various threat models that arise in the GBML setup, before presenting the different DP notions that can be achieved. Finally, we highlight the specific model and type of DP that we analyze.

### 2.1 Parameter Transfer Meta-Learning

In parameter transfer meta-learning, we assume that there is a set of learning tasks $t = 1, \ldots, T$, each with its corresponding disjoint training set $D_t$. Each $D_t$ contains $m_t$ training examples $\{z_{t,i}\}_{i=1}^{m_t}$ where each $z_{t,i} \in \mathcal{X} \times \mathcal{Y}$. The goal within each task is to learn a function $f_{\hat{\theta}_t} : \mathcal{X} \to \mathcal{Y}$ parameterized by $\hat{\theta}_t \in \Theta \subset \mathbb{R}^d$ that performs "well," generally in the sense that it has low within-task population risk in the distributional setting. The meta-learner's goal is to learn an initialization $\phi \in \Theta$ that leads to a well-performing $\hat{\theta}_t$ within-task. In GBML this $\phi$ is learned via an iterative process that alternates between the following two steps: (1) a within-task procedure where a batch of task-owners $B$ receives the current $\phi$ and each $t \in B$ uses $\phi$ as an initialization for running a within-task optimization procedure, obtaining $\hat{\theta}_t(D_t, \phi)$; (2) a meta-level procedure where the meta-learner receives these model updates $\{\hat{\theta}_t\}_{t \in B}$ and aggregates them to determine an updated $\phi$.

### 2.2 Threat Models for GBML

As in any privacy endeavor, before discussing particular mechanisms, a key specification must be made in terms of what threat model is being considered. In particular, it must be specified both (1) who the potential adversaries are and (2) what information needs to be protected.

**Potential adversaries.** For a single task-owner, adversaries may be either solely recipients of $\phi$ (i.e. other task-owners) or recipients of either $\phi$ or $\hat{\theta}_t$ (i.e. also the meta-learner). In the latter case, we consider only a honest-but-curious meta-learner, who does not deviate from the agreed upon algorithm but may try to make inferences from $\hat{\theta}_t$. In both cases, concern is placed not only about these other participants' intentions, but also their own security against access by malicious outsiders.

**Data to be protected.** A system can choose either to protect information contained in single records $z_{t,i}$ one-at-a-time or to protect entire datasets $D_t$ simultaneously. This distinction between *record-level* and *task-level* privacy can be practically important. Multiple $z_{t,i}$ within $D_t$ may reveal the same secret (e.g., a cell-phone user has sent their SSN multiple times), or the entire distribution of $D_t$ could reveal sensitive information (e.g., a user has sent all messages in a foreign language). In these cases, record-level privacy may not be sufficient. However, given that privacy and utility are often at odds, we often seek the weakest notion of privacy needed in order to best preserve utility.

In related work, focus has primarily been placed on task-level protections. However, these works usually fall into two extremes, either obtaining strong learning but having to trust the meta-learner (McMahan et al., 2018; Geyer et al., 2018) or trusting nobody but also obtaining low performance (Bhowmick et al., 2019). In response, we try to bridge the gap between these threat models by considering a model that makes a relaxation from task-level to record-level privacy but retains

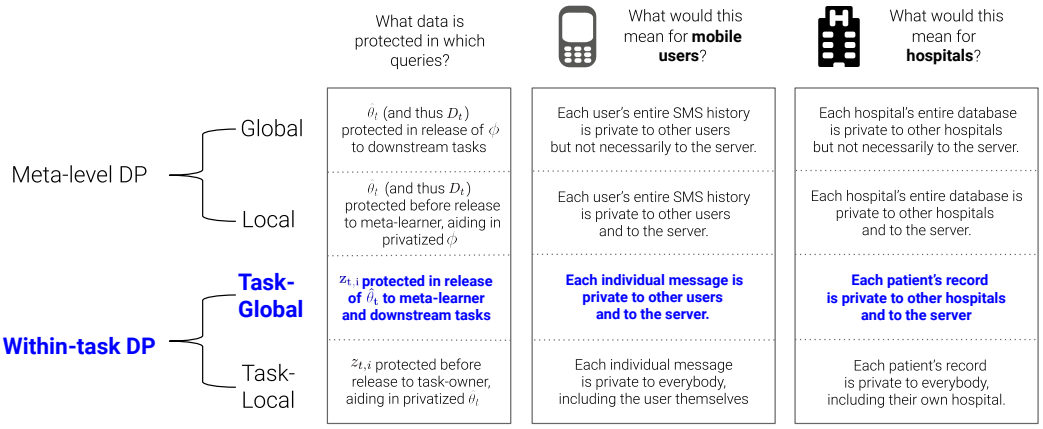


Figure 1: Summary of the privacy protections guaranteed by local and global DP at the different levels of the meta-learning problem (with our notion in blue). On the right, we show what each specification would mean in two practical federated scenarios: mobile users and hospital networks.

protections for each task-owner against *all* other parties. This relaxation can be reasonably justified in practical situations, as while task-level guarantees are strictly stronger, they may also be unnecessary. In particular, record-level guarantees are likely to be sufficient whenever single records each pertain to different individuals. For example, for hospitals, what we care about is providing privacy to the individual patients and not aggregate hospital information. For cell-phones, if one can bound the number of texts that could contain the *same* sensitive information, then an straightforward extension of our setting and methods, which protects up to $k$ records simultaneously, could also be sufficient.

### 2.3 Differential Privacy (DP) in a Single-Task Setting

In terms of actually achieving privacy guarantees for machine learning, a de-facto standard has been to apply DP, a provision which strongly limits what one can infer about the examples a given model was trained on. Assuming a training set $D = \{z_1, \ldots, z_m\}$, two common types of DP are considered.

**Differential Privacy (Global DP).** A randomized mechanism $\mathcal{M}$ is $(\varepsilon, \delta)$-*differentially private* if for all measurable $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all datasets $D, D'$ that differ by at most one element:

$$\mathbb{P}[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(D') \in \mathcal{S}] + \delta$$

If this holds for $D, D'$ differing by at most $k$ elements, then $(\varepsilon, \delta)$ $k$-*group DP* is achieved.

**Local Differential Privacy.** A randomized mechanism $\mathcal{M}$ is $(\varepsilon, \delta)$-*locally differentially private* if for any two possible training examples $z, z' \in \mathcal{X} \times \mathcal{Y}$ and measurable $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$:

$$\mathbb{P}[\mathcal{M}(z) \in \mathcal{S}] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(z') \in \mathcal{S}] + \delta$$

Global DP guarantees the difficulty of inferring the presence of a specific record in the training set by observing $\mathcal{M}(D)$. It assumes a trusted *aggregator* running $\mathcal{M}$ gets direct access to $D$ and privatizes the final output. Meanwhile, local DP assumes more strictly that the aggregator also cannot be trusted, thus requiring a random mechanism to be applied individually on each $z$ *before* training. However, it generally results in worse model performance, suffering from hard minimax rates (Duchi et al.).

### 2.4 Differential Privacy for a GBML Setting

In meta-learning, there exists a hierarchy of agents and statistical queries, so we cannot as simply define global and local DP. Here, both the meta-level sub-procedure ,$\{\hat{\theta}_t\}_{t\in B} \to \phi$, and the within-task sub-procedure, $\{z_{t,i}\}_{i=1}^{m_t} \to \hat{\theta}_t$, can be considered individual queries and a DP algorithm can implement either to be DP. Further, for each query, the procedure may be altered to satisfy either local DP or global DP. Thus, there are four fundamental options that follow from standard DP definitions.

Table 1: Broad categorization of the DP settings considered by our work in meta-learning and notable past works in the federated setting.

| Previous Work | Notion of DP | Privacy for $\phi$ | Privacy for $\theta_t$ |
|---|---|---|---|
| McMahan et al. (2018) | Global | Task-level | - |
| Geyer et al. (2018) | Global | Task-level | - |
| Bhowmick et al. (2019) | Local, Global | Task-level | Task-level |
| Agarwal et al. (2018) | Local + MPC | Task-level | Task-level |
| Truex et al. (2019) | Task-Global + MPC | Record-level | Record-level |
| Our work | Task-Global | Record-level | Record-level |

(1) *Global DP:* Releasing $\phi$ will at no point compromise information regarding any specific $\hat{\theta}_t$.
(2) *Local DP:* Additionally, each $\hat{\theta}_t$ is protected from being revealed to the meta-learner.
(3) *Task-Global DP:* Releasing $\hat{\theta}_t$ will at no point compromise any specific $z_{t,i}$.
(4) *Task-Local DP:* Additionally, each $z_{t,i}$ is protected from being revealed to task-owner.

To form analogies to single-task DP, the examples in the meta-level procedure are the model updates and the aggregator is the meta-learner. For the within-task procedure, the examples are actually the individual records and the aggregator is the task-owner. As such, (1) is implemented by the meta-learner, (2) and (3) are implemented by the task-owner, and (4) is implemented by record-owners. By immunity to post-processing, the guarantees for (3) and (4) also automatically apply to the release of any future iteration of $\phi$, thus protecting against future task-owners as well. Meanwhile, though (1) and (2) by definition protect the identities of individual $\hat{\theta}_t$, they actually satisfy a task-level threat model by doing so. Intuitively, not being able to infer anything about $\hat{\theta}_t$ implies that nothing can be inferred about the $D_t$ that was used to generate it.

Using the terminology we introduce in Section 2.4, previous works for DP in federated settings can be categorized as in Table 1. While these works do not assume a multi-task setting, we can still naturally use the terms *global/local* and *task-global/task-local* to analogously refer to releasing the global model (by the central server) and user-specific updates (by users' devices) respectively.

## 3 Differentially Private Parameter-Transfer

### 3.1 Algorithm

We now present our DP GBML method, which is written out in its online (regret) form in Algorithm 1. Here, we observe that both within-task optimization and meta-optimization are done using some form of gradient descent. The key difference between this algorithm and traditional GBML is that since task-learners must send back privatized model updates, each now applies an DP gradient descent procedure to learn $\hat{\theta}_t$ when called. However, at meta-test time the task-learner will run a *non-private* descent algorithm to obtain the parameter $\bar{\theta}_t$ used for inference, as this parameter may remain locally. To obtain learning-theoretic guarantees, we use a variant of Algorithm 1 in which the DP algorithm is an SGD procedure (Bassily et al., Algorithm 1) that adds a propertly scaled Gaussian noise vector at each iteration. A stability result due to Bassily et al. regarding the population loss of this algorithm's output allows us to provide bounds on the transfer risk due to our meta-algorithm.

### 3.2 Privacy Guarantees

We run a certified $(\varepsilon, \delta)$-DP version of SGD (Bassily et al., Algorithm 1) within each task. Therefore, this guarantees that the contribution of each task-owner, a $\hat{\theta}_t$ trained on their data, carries global DP guarantees with respect to the meta-learner. Additionally, since DP is preserved under post-processing, the release of any future calculation stemming from $\hat{\theta}_t$ also carries the same DP guarantee.

### 3.3 Learning Guarantees

Our learning result follows the setup of Baxter (2000), who formalized the LTL problem as using task-distribution samples $\mathcal{P}_1, \ldots, \mathcal{P}_T \sim \mathcal{Q}$ from some meta-distribution $\mathcal{Q}$ and samples indexed by

**Algorithm 1:** Online version of our $(\varepsilon, \delta)$-meta-private parameter-transfer algorithm.

Meta-learner picks first meta-initialization $\phi_1 \in \Theta$.
**for** task $t \in [T]$ **do**
  Meta-learner sends meta-initialization $\phi_t$ to task $t$.
  Task-learner runs OGD starting from $\theta_{t,1} = \phi_t$ on losses $\{\ell_{t,i}\}_{i=1}^m$, suffering regret $\sum_{i=1}^m \ell_{t,i}(\theta_{t,i}) - \min_{\theta\in\Theta} \sum_{i=1}^m \ell_{t,i}(\theta)$.
  Task-learner $t$ runs $(\varepsilon, \delta)$-DP descent algorithm on losses $\{\ell_{t,i}\}_{i=1}^m$ to get $\hat{\theta}_t$.
  Task-learner sends $\hat{\theta}_t$ to meta-learner.
  Meta-learner constructs loss $\ell_t(\phi) = \frac{1}{2}\|\hat{\theta}_t - \phi_t\|_2^2$.
  Meta-learner picks meta-initialization $\phi_{t+1}$ using an OCO algorithm on $\ell_1, \ldots, \ell_t$.

$i = 1, \ldots, m$ from those tasks to improve performance when a new task $\mathcal{P}$ is sampled from $\mathcal{Q}$ and we draw $m$ samples from it. In the setting of parameter-transfer meta-learning we are learning functions parameterized by real-valued vectors $\theta \in \Theta \subset \mathbb{R}^d$, so our goal will follow that of Denevi et al. (2019) and Khodak et al. (2019b) in seeking bounds on the transfer-risk – the distributional performance of a learned parameter on a new task from $\mathcal{Q}$ – that improve with task similarity.

The specific task-similarity metric we consider is the average deviation of the risk-minimizing parameters of tasks sampled from the distribution $\mathcal{Q}$ are close together. This will be measured in-terms of the following quantity: $V^2 = \min_{\phi\in\Theta} \frac{1}{2}\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\|\theta_{\mathcal{P}} - \phi\|_2^2$, for $\theta_P \in \arg\min_{\theta\in\Theta} \ell_{\mathcal{P}}(\theta)$ a risk-minimizer of task-distribution $\mathcal{P}$. This quantity is roughly the variance of risk-minimizing task-parameters and is a standard quantifier of improvement due to meta-learning (Denevi et al., 2019; Khodak et al., 2019b). For example, Denevi et al. (2019) show excess transfer-risk guarantees of the form $\mathcal{O}\left(\frac{V}{\sqrt{m}} + \sqrt{\frac{\log T}{T}}\right)$ when $T$ tasks with $m$ samples are drawn from the distribution. This guarantee ensures that as we see more tasks our transfer risk becomes roughly $V/\sqrt{m}$, which if the tasks are similar, i.e. $V$ is small, implies that LTL improves over single-task learning.

In Algorithm 1, each user $t$ obtains a within-task parameter $\bar{\theta}_t$ by running (non-private) OGD on a sequence of losses $\ell_{t,1}, \ldots, \ell_{t,m}$ and averaging the iterates. The regret of this procedure, when averaged across the users, implies a bound on the expected excess transfer risk of new task from $\mathcal{Q}$ when running OGD from a learned initialization (Cesa-Bianchi et al., 2004). Thus our goal is to bound this regret in terms of $V$; here we follow the Average Regret-Upper-Bound Analysis (ARUBA) framework of Khodak et al. (2019b) and treat meta-update procedure itself as an online algorithm optimizing a bound on the performance measure (regret) of each within-task algorithm. As OGD's regret depends on the squared distance $\frac{1}{2}\|\theta_t^* - \phi_t\|_2^2$ of the optimal parameter from the initialization $\phi_t$, with no privacy concerns one could simply update $\phi_t$ using $\theta_t^* \in \arg\min_{\theta\in\Theta} \sum_{i=1}^m \ell_{t,i}(\theta)$ to recover guarantees similar to those in Denevi et al. (2019) and Khodak et al. (2019b).

However, this approach requires sending $\theta_t^*$ to the meta-learner, which is not private; instead in Algorithm 1 we send $\hat{\theta}_t$, which is the output of noisy SGD. To apply ARUBA, we need an additional assumption – that the losses satisfy the following quadratic growth (QG) property: for some $\alpha > 0$,

$$\frac{\alpha}{2}\|\theta - \theta_{\mathcal{P}}\|_2^2 \leq \ell_{\mathcal{P}}(\theta) - \ell_{\mathcal{P}}(\theta_{\mathcal{P}}) \quad \forall\, \theta \in \Theta \tag{1}$$

Here $\theta_{\mathcal{P}}$ is the risk minimizer of $\ell_{\mathcal{P}}$. This assumption, which Khodak et al. (2019a) show is reasonable in settings such as logistic regression, amounts to a statistical non-degeneracy assumption on the parameter-space – that parameters far away from the risk-minimizer do not have low-risk. Note that QG is significantly weaker than strong convexity, which previous work (Finn et al., 2019) has assumed to hold for task losses but does not hold for applicable cases such as few-shot least-squares or logistic regression if the number of task-samples is smaller than the data-dimension.

We are now able to state our main theoretical result, a proof of which is given in Appendix B. The result follows from a bound on the task-average regret (TAR) across all tasks of a simple online meta-learning procedure that treats the update $\hat{\theta}_t$ sent by each task as an approximation of the optimal parameter in hindsight $\theta_t^*$. Since this parameter determines regret on that task, by reducing the meta-update procedure to OCO on this sequence of functions in a manner similar to (Khodak et al., 2019a), we are able to show a task-similarity-dependent bound on the TAR. Following this the

statistical guarantee stems from a nested online-to-batch conversion, a standard procedure to convert low-regret online-learning algorithms to low-risk distribution-learning algorithms.

**Theorem 3.1.** *Suppose $\mathcal{Q}$ is a distribution over task-distributions $\mathcal{P}$ over $G$-Lipschtz, $\beta$-Lipschitz-smooth, 1-bounded convex loss functions $\ell : \Theta \mapsto \mathbb{R}$ over parameter space $\Theta$ with diameter $D$, and let each $\mathcal{P}$ satisfy the quadratic growth property* (1)*. Suppose the distribution $\mathcal{P}_t$ of each task is sampled i.i.d. from $\mathcal{Q}$ and we run Algorithm 1 with the $(\varepsilon, \delta)$-DP procedure of Bassily et al., Algorithm 1 to obtain $\hat{\theta}_t$ as the average iterate for the meta-update step. Then if $\eta = V/(G\sqrt{m})$ for $V^2 = \min_{\phi \in \Theta} \frac{1}{2}\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\|\theta_{\mathcal{P}} - \phi\|_2^2$ and $\beta \leq \frac{G}{D}\min\left(\frac{\sqrt{m}}{2}, \frac{\varepsilon m}{4\sqrt{d \log \frac{1}{\delta}}}\right)$ we have the following bound on the expected transfer risk when a new task $\mathcal{P}$ is sampled from $\mathcal{Q}$, $m$ samples are drawn i.i.d. from $\mathcal{P}$, and we run OGD with learning rate $\eta$ starting from $\bar{\phi} = \frac{1}{T}\sum_{t=1}^{T}\phi_t$ and use the average $\bar{\theta}$ of the resulting iterates as the learned parameter:*

$$\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\bar{\theta}) \leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V}{\sqrt{m}} + \frac{D^2}{VT\sqrt{m}} + \frac{D}{V\alpha}\max\left(\frac{\sqrt{\frac{d}{\varepsilon}\log\frac{1}{\delta}}}{m^{\frac{3}{2}}}, \frac{1}{m}\right)\right)$$

*Here $\theta^*$ is any element of $\Theta$ and the outer expectation is taken over $\ell_{t,i} \sim \mathcal{P}_t \sim \mathcal{Q}$ and the randomness of the within-task DP mechanism. Note that this procedure is $(\varepsilon, \delta)$-DP.*

Theorem 3.1 shows that one can usefully run a DP-algorithm as the within-task method in meta-learning and still obtain improvement due to task-similarity. Specifically, the standard term of $1/\sqrt{m}$ is multiplied by $V$, which is small if the tasks are related via the closeness of their risk minimizers. Thus we can use meta-learning to improve within-task performance relative to single-task learning. We also obtain a very fast convergence of $1/T\sqrt{m}$ in the number of tasks. However, we do gain some $o(1/\sqrt{m})$ terms due to the quadratic growth approximation and the privacy mechanism. Note that the assumption that both the functions and its gradients are Lipschitz-continuous are standard and required by the noisy SGD procedure of Bassily et al..

This theorem also gives us a relatively straightforward extension if the desire is to provide $(\varepsilon, \delta)$-group-DP. Since any privacy mechanism that provides $(\varepsilon, \delta)$-DP also provides $(k\varepsilon, ke^{(k-1)\epsilon}\delta)$-DP guarantees for groups of size $k$ (Dwork and Roth, 2014), we immediately have the following corollary.

**Corollary 3.1.** *Under the same assumptions and setting as Theorem 3.1, achieving $(\varepsilon, \delta)$-group DP is possible with the same guarantee except replacing $\sqrt{\frac{d}{\varepsilon}\log(\frac{1}{\delta})}$ with $\sqrt{k^2 d + kd[\frac{1}{\epsilon}\log(\frac{k}{\delta}) - 1]}$.*

For constant $k$, this allows us to enjoy the stronger guarantee while maintaining largely the same learning rates. This is a useful result given that in some settings, it may be desired to simultaneously protect small groups of size $k << m_t$, such as protecting entire families for hospital records.

## 4 Empirical Results

We present results that show it is possible to learn useful deep models in federated scenarios while still preserving privacy against all other participants. Specifically, we evaluate the performance of models that have been trained with a *task-global* DP algorithm in comparison to models that are trained both non-privately and those satisfying *local* DP. We evaluate performance on federeated language modeling and few-shot image classification, applying a practical batched variant of Algorithm 1.

**Datasets:** We train a LSTM-RNN for next word prediction on two federated datasets: (1) The Shakespeare dataset as preprocessed by (Caldas et al., 2018), and (2) a dataset constructed from $3,000$ Wikipedia articles drawn from the Wiki-3029 dataset (Arora et al., 2019), where each article is used as a different task. For each dataset, we set a fixed number of tokens per task, discard tasks with fewer tokens than the specified, and discard samples from those tasks with more. We set the number of tokens per task to $800$ for Shakespeare and to $1,600$ for Wikipedia, divide tokens into sequences of length $10$, and we refer to these modified datasets as Shakespeare-800 and Wiki-1600.

For few-shot image classification, we use the Omniglot dataset (Lake et al., 2011) with 5-shot-5-way test tasks. As has been done on Reptile (Nichol et al.), we use more training shots at meta-training (trying $m = 10, 20, 30$) than at meta-test time. Though tasks could be sampled indefinitely, we set a

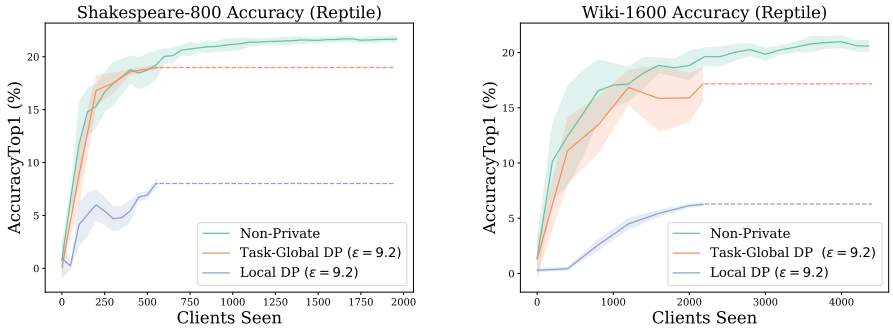


Figure 2: Performance of different versions of Reptile on a next-word-prediction task for two federated datasets. We report the test accuracy on unseen tasks and repeat each experiment 10 times. Solid lines correspond to means, colored bands indicate 1 standard deviation, and dotted lines are for comparing final accuracies (private algorithms can only be trained until privacy budget is met).

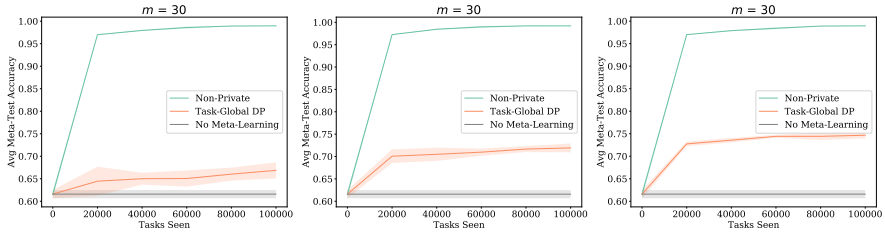


Figure 3: Performance of task-global DP Reptile on 5-shot-5-way Omniglot. $10^5$ sampled test- tasks were used for evaluation and experiments were repeated 3 times.

fixed budget of tasks at $T = 10^6$ to reflect to a realistic setting in which a finite number of training tasks constrains our learning and privacy–for both local and task-global DP– the more tasks that can be grouped in a meta-batch means that less total noise can be added at each iteration, but this also means fewer iterations can be taken. We note that this setting is enough to achieve convergence at 99% accuracy for non-private training.

**Meta Learning Algorithm.** We study the performance of our method when applied to the batch version of Reptile (Nichol et al., 2018) (which, in our setup, reduces to personalized Federated Averaging when the meta-learning rate is set to 1.0). For the language modelling tasks, we tune various configurations of task batch size for all methods and for the non-private baseline, and also allow for multiple visits per client. Additionally, for language modeling, we implement gradient clipping and exponential decay on the meta learning rate. For Omniglot, we use largely the same parameters as Nichol et al. (2018) but we explore the effects of altering the number of training shots, the $L_2$ clipping threshold, the Adam Learning Rate, and the meta-batch size. We defer a more complete discussion of hyperparameter tuning to Appendix C.

**Privacy Considerations.** For the *task-global* DP models, we set $\delta = 10^{-3} < \frac{1}{m^{1.1}}$ on each task and we implement DP-SGD (for language modelling) and DP-Adam (for Omniglot) within-task using the tools provided by *TensorFlow Privacy*[1]. Although these algorithms differ from the one presented in Section 3, they let us explore *task-global* privacy in a realistic setting. We use the the *RDP accountant* to track our privacy budgets. Finally, for the language modeling datasets, we make sure that all tasks are sampled without replacement with a fixed batch size until all are seen. This is necessary since multiple visits to a single client results in degradation of the privacy guarantee for that client. We instead aim to provide the same guarantee for each client. For local-DP, though this notion of DP is stronger, we explore the same privacy budgets so as to obtain guarantees that are

[1]https://github.com/tensorflow/privacy

of the same *confidence*. Here, we essentially run the DP-FedAvg algorithm from (McMahan et al., 2018) with some key changes. First, to get local DP instead of global, we add Gaussian noise to each clipped set of model updates before returning them to the central server instead of after aggregation. Second, we again iterate through tasks without replacement.

**Results.** Figure 3 shows the performance of both the non-private and *task-global* private versions of Reptile (Nichol et al., 2018) for the language modelling tasks. As expected, in neither case does the private version reach the same accuracy of the non-private version of the algorithm. Nonetheless, the private version still comes within $88\%$ of the non-private accuracy for Shakespeare-800 and within $82\%$ for Wiki-1600. Meanwhile achieving local DP results in only about $42\%$ and $30\%$ of the non-private accuracy on both datasets. In practice, these differences can be toggled by changing the privacy budget for the algorithm or trading off more training iterations for larger noise multipliers.

For Omniglot, not applying meta-learning in this setting results in meta-test accuracy of $61.6\%$. Thus, while performance is indeed compromised compared to non-private learning, applying task-global DP *does* result in meta-learning benefits for test-time tasks. In settings where privacy is a concern, this increase in performance is still significantly advantageous for the "task-owners"– test-time tasks (who hold less data) on average see up to $14\%$ improvements while the train-time tasks are guaranteed reasonably low (single-digit) $\varepsilon$. Intuitively, larger training-task datasets make it easier to apply privacy within-task, and in accordance with our learning guarantees, adding training shots indeed closes the gap in performance between task-global DP Reptile and non-private Reptile. In comparison, applying Local-DP for a similar hyperparameter range consistently decreases performance at test-time. However, the non-meta-learning baseline is a theoretical lower bound for Local-DP, as one could set the clipping threshold or meta-learning rate close to $0$ to recover the effects of no meta-learning.

## 5 Conclusions

In this work, we have outlined and studied the issue of privacy in the context of meta-learning. Focusing on the class of gradient-based parameter-transfer methods, we used differential privacy to address the privacy risks posed to task-owners by sharing task-specific models with a central meta-learner. To do so, we formalized and considered the notion of *task-global* differential privacy, which guarantees that individual examples from the tasks are protected from all downstream agents (and particularly the meta-learner). Working in this privacy model, we developed a differentially private algorithm that guarantees both this protection as well as learning-theoretic results in the convex setting. Finally, we demonstrate how this notion of privacy can translate into useful deep learning models for non-convex language modelling and image-classification tasks.

## References

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems 31*, pages 7564–7575. Curran Associates, Inc., 2018.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Nikunj Saunshi, and Orestis Plevrakis. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Thakurta. Private stochastic convex optimization with optimal rates. URL `https://arxiv.org/abs/1908.09970`.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12: 149–198, 2000.

Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning, 2019. `https://arxiv.org/abs/1812.00984`.

Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings, 2018. URL `http://arxiv.org/abs/1812.01097`.

Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets, 2018. URL http://arxiv.org/abs/1802.08232.

Nicoló Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning for recommendation. *CoRR*, abs/1802.07876, 2018. URL http://arxiv.org/abs/1802.07876.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization, 2019. URL http://arxiv.org/abs/1903.10399.

John Duchi, Martin Wainwright, and Michael Jordan. Minimax optimal procedures for locally private estimation. In *Journal of the American Statistical Association*.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3&4):211–407, 2014. doi: 10.1561/0400000042.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham M. Kakade, and Sergey Levine. Online meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

Robin C. Geyer, Tassilo J. Klein, and Moin Nabi. Differentially private federated learning: A client level perspective, 2018. URL https://openreview.net/forum?id=SkVRTj0cYQ.

Bargav Jayaraman and David Evans. When relaxations go bad: "differentially-private" machine learning, 2019. URL http://arxiv.org/abs/1902.08874.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2016.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019a.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, 2019b. To Appear.

Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *CogSci*, 2011.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private language models. In *ICLR*, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. URL http://arxiv.org/abs/1803.02999.

Reza Shokri, Marco Stronati, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of 2017 IEEE Symposium on Security and Privacy*, pages 3–18, 2017.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems 31*, 2017.

Salvatore J. Stolfo, David W. Fan, Wenke Lee, Andreas L. Prodromidis, and Philip K. Chan. Credit card fraud detection using meta-learning: Issues and initial results 1. In *Working notes of AAAI Workshop on AI Approaches to Fraud Detection and Risk Management.*, 1997.

Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, and Rui Zhang. A hybrid approach to privacy-preserving federated learning, 2019. URL http://arxiv.org/abs/1812.03224.

Xi Sheryl Zhang, Fengyi Tang, Hiroko Dodge, Jiayu Zhou, and Fei Wang. Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records, 2019. URL https://arxiv.org/abs/1905.03218.

**Algorithm 1:** Online version of our $(\varepsilon, \delta)$-meta-private parameter-transfer algorithm.

Meta-learner picks first meta-initialization $\phi_1 \in \Theta$.
**for** task $t \in [T]$ **do**
  Meta-learner sends meta-initialization $\phi_t$ to task $t$.
  Task-learner runs OGD starting from $\theta_{t,1} = \phi_t$ on losses $\{\ell_{t,i}\}_{i=1}^m$, suffering regret $\sum_{i=1}^m \ell_{t,i}(\theta_{t,i}) - \min_{\theta\in\Theta} \sum_{i=1}^m \ell_{t,i}(\theta)$.
  Task-learner $t$ runs $(\varepsilon, \delta)$-DP descent algorithm on losses $\{\ell_{t,i}\}_{i=1}^m$ to get $\hat{\theta}_t$.
  Task-learner sends $\hat{\theta}_t$ to meta-learner.
  Meta-learner constructs loss $\ell_t(\phi) = \frac{1}{2}\|\hat{\theta}_t - \phi_t\|_2^2$.
  Meta-learner picks meta-initialization $\phi_{t+1}$ using an OCO algorithm on $\ell_1, \ldots, \ell_t$.

## A Local-Meta-Level DP and task-global DP

**Remark A.1.** *If a GBML algorithm achieves $(\varepsilon, \delta)$-local DP at the meta-level, it is also guaranteed to be $(\varepsilon, \delta)$-DP at a task-global level.*

*Proof.* According to the definition of local DP, a mechanism $\mathcal{M}$ that achieves $(\varepsilon, \delta)$-local DP for releasing $\phi$ must satisfy for any $\hat{\theta}_t, \hat{\theta}'_t \in \Theta$ and $\mathcal{S} \subseteq \Theta$:

$$\mathbb{P}[\mathcal{M}(\hat{\theta}_t) \in \mathcal{S}] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(\hat{\theta}'_t) \in \mathcal{S}] + \delta$$

Here $\hat{\theta}_t$ can also be seen as a function, possibly stochastic, of $D_t$, or more formally, $\hat{\theta}_t = \mathcal{A}_\phi(D_t)$ where $\phi$ is an initialization and $\mathcal{A}_\phi : (\mathcal{X} \times \mathcal{Y})^m \to \Theta$. Thus, by also setting $\hat{\theta}'_t = \mathcal{A}_\phi(D'_t)$, we automatically get for any $D_t, D'_t$

$$\mathbb{P}[\mathcal{M}(\mathcal{A}_\phi(D_t)) \in \mathcal{S}] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(\mathcal{A}_\phi(D'_t)) \in \mathcal{S}] + \delta$$

This holds by definition when $\mathcal{A}_\phi$ is deterministic since $\hat{\theta}_t$ and $\hat{\theta}'_t$ are single elements from $\Theta$. When $\hat{\theta}_t$ and $\hat{\theta}'_t$ are stochastic, this bound also holds since it holds even in the worst case for any single pair of elements in $\Theta$. Further, the bound holds no matter how many elements differ between $D_t$ and $D'_t$, as long as $\mathcal{A}_\phi$ outputs something in $\Theta$. Thus, if we treat $\mathcal{M}(\mathcal{A}_\phi(\cdot))$ as one mechanism, we get the given proposition.

□

## B Proofs of Learning Guarantees

Throughout this section we assume all subsets are convex and in $\mathbb{R}^d$ unless explicitly stated. In the online learning setting we will use the shorthand $\nabla_t$ to denote the subgradient of $\ell_t : \Theta \mapsto \mathbb{R}$ evaluated at action $\theta_t \in \Theta$. For any $x_1, \ldots, x_T \in \mathbb{R}^d$ we will use $x_{1:t}$ to refer to the sum of the first $t$ of them.

In this section we first prove (Theorem B.1) a general averaged-regret bound following the ARUBA framework of Khodak et al. (2019b). We then combine an algorithmic stability based $(\varepsilon, \delta)$-DP generalization bound for noisy SGD of Bassily et al. with a quadratic growth assumption (Karimi et al., 2016; Khodak et al., 2019a) to show that such an algorithm returns a meta-update parameter $\hat{\theta}$ that is close $\theta^*$ and thus suffices to show a meaningful task-averaged-regret guarantee (Corollary B.1). We conclude by using this bound to derive a guarantee in the statistical LTL setting (Corollary B.2).

**Setting B.1.** *We assume all functions $\ell_{t,i} : \Theta \mapsto [0, 1]$ are convex and $G$-Lipschitz for some $G \geq 1$ and that $\Theta$ has $\ell_2$-diameter $D \geq 1$. We define the following quantities:*

- *convenience coefficients $\sigma = G\sqrt{m}$*

- *the sequence of update parameters* $\{\hat{\theta}_t \in \Theta\}_{t\in[T]}$ *with mean* $\hat{\phi} = \frac{\hat{\theta}_{1:T}}{T}$
- *a sequence of reference parameters* $\{\theta'_t \in \Theta\}_{t\in[T]}$ *with mean* $\phi' = \frac{\theta'_{1:T}}{T}$
- *a sequence* $\{\theta^*_t \in \Theta\}_{t\in[T]}$ *of optimal parameters in hindsight*
- $\kappa \geq 1, \Delta^* \geq 0$ *s.t.* $\sigma \sum_{t=1}^T \mathbb{E}\,\|\theta^*_t - \phi_t\|_2^2 \leq \Delta^* + \kappa\sigma \sum_{t=1}^T \mathbb{E}\,\|\hat{\theta}_t - \phi_t\|_2^2$
- $\nu \geq 1, \Delta' \geq 0$ *s.t.* $\sigma \sum_{t=1}^T \mathbb{E}\,\|\hat{\theta}_t - \hat{\phi}\|_2^2 \leq \Delta' + \nu\sigma \sum_{t=1}^T \mathbb{E}\,\|\theta'_t - \phi'\|_2^2$
- *positive task-similarity* $V^2 = \frac{1}{2}\sum_{t=1}^T \mathbb{E}\,\|\theta'_t - \phi'\|_2^2$
- *learning-rate* $\eta = \frac{H}{G\sqrt{m}}$ *for some* $H > 0$

**Theorem B.1.** *In Setting B.1 define the regret upper-bound* $\hat{\mathbf{R}}_t = \frac{\|\theta^*_t - \phi_t\|_2^2}{2\eta} + \eta G^2 m$ *and the averaged regret upper-bound* $\hat{\bar{\mathbf{R}}} = \frac{1}{T}\sum_{t=1}^T \hat{\mathbf{R}}_t$*. Then in Algorithm 1 if the meta-learner uses FTL or AOGD to pick the meta-initialization and the within-task descent algorithm has regret upper-bounded by* $\hat{\mathbf{R}}_t$ *we have the following bound:*

$$\mathbb{E}\,\hat{\bar{\mathbf{R}}} \leq \frac{\Delta^* + \kappa\Delta'}{HT} + \left(\frac{D^2\kappa}{2H}\frac{1+\log T}{T} + H + \frac{\kappa\nu V^2}{H}\right) G\sqrt{m}$$

*Here the expectation is taken over the randomness of the DP mechanism.*

*Proof.* We apply the standard FTRL regret of OGD, e.g. Theorem A.1 in Khodak et al. (2019a), and the logarithmic regret of FTL and AOGD, e.g. Theorem A.2 in Khodak et al. (2019a):

$$\begin{aligned}
T\,\mathbb{E}\,\hat{\bar{\mathbf{R}}} &= \mathbb{E}\left(\sum_{t=1}^T \frac{\|\theta^*_t - \phi_t\|_2^2}{2\eta} + \eta G^2 m\right) \\
&= \frac{\Delta^*}{H} + \sigma\sum_{t=1}^T \mathbb{E}\left(\frac{\kappa}{2H}\|\hat{\theta}_t - \phi_t\|_2^2 + H\right) \\
&= \frac{\Delta^*}{H} + H\sigma T + \frac{\kappa\sigma}{2H}\sum_{t=1}^T \mathbb{E}\left(\|\hat{\theta}_t - \phi_t\|_2^2 - \|\hat{\theta}_t - \hat{\phi}\|_2^2 + \|\hat{\theta}_t - \hat{\phi}\|_2^2\right) \\
&\leq \frac{\Delta^*}{H} + H\sigma T + \frac{D^2\kappa\sigma}{2H}(1+\log T) + \frac{\kappa\Delta'}{H} + \frac{\kappa\nu\sigma}{2H}\sum_{t=1}^T \mathbb{E}\,\|\theta'_t - \phi'\|_2^2
\end{aligned}$$

□

**Setting B.2.** *In Setting B.1, assume loss functions* $\ell_{t,1}, \ldots, \ell_{t,m}$ *are generated by picking some distribution* $\mathcal{P}_t$ *over valid losses and then sampling* $m$ *of them i.i.d. Assume further that the expected loss of every such distribution satisfies* $\alpha$*-quadratic-growth (*$\alpha$*-QG): for some* $\alpha > 0$*, any* $\theta \in \Theta$*, and* $\theta'$ *the closest minimizer of* $\mathbb{E}\,\ell$ *to* $\theta$ *we have*

$$\frac{\alpha}{2}\|\theta - \theta'\|_2^2 \leq \mathbb{E}(\ell(\theta) - \ell(\theta'))$$

*Furthermore, assume that these losses are* $\beta$*-strongly-smooth:*

$$\ell(\theta) \leq \ell(\theta') + \langle\nabla\ell(\theta'), \theta - \theta'\rangle + \frac{\beta}{2}\|\theta - \theta'\|_2^2$$

*Finally, assume that* $\theta'$ *is unique for every* $\mathcal{P}_t$*.*

**Lemma B.1.** *Let* $\ell_1, \ldots, \ell_m : \Theta \mapsto [0,1]$ *be a sequence of convex losses drawn i.i.d. from some distribution* $\mathcal{D}$ *with risk* $\mathbb{E}\,\ell$ *being* $\alpha$*-QG and let* $\theta^* \in \arg\min_{\theta\in\Theta}\sum_{i=1}^m \ell_i(\theta)$ *be any of the optimal actions in hindsight. Then the closest minimum* $\theta'$ *of* $\mathbb{E}\,\ell$ *to* $\theta^*$ *satisfies*

$$\frac{1}{2}\,\mathbb{E}\,\|\theta^* - \theta'\|^2 \leq \frac{2}{\alpha}\sqrt{\frac{1+\log m}{m}}$$

*Proof.* Taking expectations of the result of Lemma B.4 in Khodak et al. (2019a), we have for $\delta = \frac{2}{\sqrt{m}}$ that

$$\frac{\alpha}{2}\,\mathbb{E}\,\|\theta^* - \theta'\|_2^2 \leq \sqrt{\frac{8}{m}\log\frac{2}{\delta}} + \delta \leq \sqrt{\frac{4}{m}(1+\log m)}$$

□

**Lemma B.2.** *Let $\ell_1, \ldots, \ell_m : \Theta \mapsto [0,1]$ be a sequence of $\beta$-strongly-smooth, $G$-Lipschitz convex losses drawn i.i.d. from some distribution $\mathcal{D}$ with risk $\mathbb{E}\,\ell$ being $\alpha$-QG and let $\hat{\theta} \in \Theta$ be the average iterate of running Algorithm 1 of Bassily et al. with the appropriate parameters for obtaining $(\varepsilon, \delta)$-DP. If $\beta \leq \frac{G}{D}\min\left(\frac{\sqrt{m}}{2}, \frac{\varepsilon m}{4\sqrt{d\log\frac{1}{\delta}}}\right)$ then the closest minimum $\theta'$ of $\mathbb{E}\,\ell$ to $\hat{\theta}$ satisfies*

$$\frac{1}{2}\,\mathbb{E}\,\|\hat{\theta} - \theta'\|_2^2 \leq \mathbb{E}(\ell(\hat{\theta}) - \ell(\theta')) \leq \frac{10GD}{\alpha}\max\left(\frac{\sqrt{d\log\frac{1}{\delta}}}{\varepsilon m}, \frac{1}{\sqrt{m}}\right)$$

*Proof.* The result follows by directly substituting Theorem 3.2 of Bassily et al. into the definition of $\alpha$-QG:

$$\frac{\alpha}{2}\,\mathbb{E}\,\|\hat{\theta} - \theta'\|_2^2 \leq \mathbb{E}(\ell(\hat{\theta}) - \ell(\theta')) \leq 10GD\max\left(\frac{\sqrt{d\log\frac{1}{\delta}}}{\varepsilon m}, \frac{1}{\sqrt{m}}\right)$$

□

**Proposition B.1.** *In Setting B.2 we have $\kappa = \nu = 3$ and*

$$\Delta^* = \frac{33G^2DT}{\alpha}\max\left(\sqrt{\frac{d}{\varepsilon m}\log\frac{1}{\delta}}, \sqrt{1+\log m}\right) \quad \textit{and} \quad \Delta' = \frac{10G^2DT}{\alpha}\max\left(\sqrt{\frac{d}{\varepsilon m}\log\frac{1}{\delta}}, 1\right)$$

*Proof.* We apply the triangle inequality, Jensen's inequality, and Lemmas B.1 and B.2 to get

$$\begin{aligned}
&\frac{\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\,\|\theta_t^* - \phi_t\|_2^2 \\
&\quad\leq \frac{3\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\left(\|\theta_t^* - \theta_t'\|_2^2 + \|\theta_t' - \hat{\theta}_t\|_2^2 + \|\hat{\theta}_t - \phi_t\|_2^2\right) \\
&\quad\leq 3\sigma\sum_{t=1}^{T}\left(\frac{6}{\alpha}\sqrt{\frac{1+\log m}{m}} + \frac{5GD}{\alpha}\max\left(\frac{\sqrt{d\log\frac{1}{\delta}}}{\varepsilon m}, \frac{1}{\sqrt{m}}\right) + \frac{1}{2}\,\mathbb{E}\,\|\hat{\theta}_t - \phi_t\|_2^2\right) \\
&\quad\leq \frac{33G^2DT\sqrt{m}}{\alpha}\max\left(\frac{\sqrt{d\log\frac{1}{\delta}}}{\varepsilon m}, \sqrt{\frac{1+\log m}{m}}\right) + \frac{3\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\,\|\hat{\theta}_t - \phi_t\|_2^2
\end{aligned}$$

We further have by the triangle inequality and Lemma B.2 that

$$\begin{aligned}
\frac{\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\,\|\hat{\theta}_t - \hat{\phi}\|_2^2 &\leq \frac{3\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\left(\|\hat{\theta}_t - \theta_t'\|_2^2 + \|\theta_t' - \phi'\|_2^2 + \|\phi' - \hat{\phi}\|_2^2\right) \\
&\leq 3\sigma\sum_{t=1}^{T}\mathbb{E}\,\|\hat{\theta}_t - \theta_t'\| + \frac{3\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\,\|\theta_t' - \phi'\|_2^2 \\
&\leq \frac{10G^2DT\sqrt{m}}{\alpha}\max\left(\frac{\sqrt{d\log\frac{1}{\delta}}}{\varepsilon m}, \frac{1}{\sqrt{m}}\right) + \frac{3\sigma}{2}\sum_{t=1}^{T}\mathbb{E}\,\|\theta_t' - \phi'\|_2^2
\end{aligned}$$

□

**Corollary B.1.** *In Setting B.2, if we run Algorithm 1 using OGD with learning rate $H$ and Algorithm 1 of Bassily et al. as the within-task $(\varepsilon, \delta)$-DP method then for $H = V$ we have the following bound on the expected task-averaged regret:*

$$\mathbb{E}\,\bar{\mathbf{R}} \leq \mathbb{E}\,\hat{\bar{\mathbf{R}}} = \frac{63G^2DT}{V\alpha}\max\left(\sqrt{\frac{d}{\varepsilon m}\log\frac{1}{\delta}}, \sqrt{1+\log m}\right) + \frac{3GD^2\sqrt{m}}{V}\frac{1+\log T}{T} + 10GV\sqrt{m}$$

*Proof.* Substitute Proposition B.1 into Theorem B.1 and simplify. □

**Corollary B.2.** *In Setting B.2 and under the assumptions of Corollary B.1, if the distribution $\mathcal{P}_t$ of each task is sampled i.i.d. from some environment $\mathcal{Q}$ and then we have the following bound on the expected transfer risk when a new task $\mathcal{P}$ is sampled from $\mathcal{Q}$, $m$ samples are drawn i.i.d. from $\mathcal{P}$, and we run OGD with $\eta = \frac{H}{G\sqrt{m}}$ starting from $\bar{\phi} = \frac{1}{T}\phi_{1:T}$ and use the average $\bar{\theta}$ of the resulting iterates as the learned parameter:*

$$\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\bar{\theta}) \leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V}{\sqrt{m}} + \frac{D^2}{VT\sqrt{m}} + \frac{D}{V\alpha}\max\left(\frac{\sqrt{\frac{d}{\varepsilon}\log\frac{1}{\delta}}}{m^{\frac{3}{2}}}, \frac{1}{m}\right)\right)$$

*Here $\theta^*$ is any element of $\Theta$ and the outer expectation is taken over $\ell_{t,i} \sim \mathcal{P}_t \sim \mathcal{Q}$ and the randomness of the DP mechanism.*

*Proof.* The result follows from two applications of the standard in-expectation online-to-batch argument, e.g. Proposition A.1 of Khodak et al. (2019a), followed by an application of Corollary B.1:

$$\begin{aligned}
\mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\bar{\theta}) &\leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}\left(\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{\hat{\mathbf{R}}(\bar{\phi})}{m}\right) \\
&\leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \frac{\hat{\mathbf{R}}(\phi^*)}{m} + \frac{\mathbb{E}\,\hat{\bar{\mathbf{R}}}}{Tm} \\
&\leq \mathbb{E}_{\mathcal{P}\sim\mathcal{Q}}\mathbb{E}_{\ell\sim\mathcal{P}}\ell(\theta^*) + \tilde{\mathcal{O}}\left(\frac{V}{\sqrt{m}} + \frac{D^2}{VT\sqrt{m}} + \frac{D}{V\alpha}\max\left(\frac{\sqrt{\frac{d}{\varepsilon}\log\frac{1}{\delta}}}{m^{\frac{3}{2}}}, \frac{1}{m}\right)\right)
\end{aligned}$$

□

## C Experiment Details

**Datasets:** We train a next word predictor for two federated datasets: (1) The Shakespeare dataset as preprocessed by (Caldas et al., 2018), and (2) a dataset constructed from Wikipedia articles, where each article is used as a different task. For each dataset, we set a fixed number of tokens per task, discard tasks with less tokens than the specified, and discard samples from those tasks with more. For Shakespeare, we set the number of tokens per task to $800$ tokens, leaving $279$ tasks for meta-training, $31$ for meta-validation, and $35$ for meta-testing. For Wikipedia, we set the number of tokens to $1,600$, which corresponds to having $2,179$ tasks for meta-training, $243$ for meta-validation, and $606$ for meta-testing. For the meta-validation and meta-test tasks, $75\%$ of the tokens are used for local training, and the remaining $25\%$ for local testing.

For Omniglot, we follow the standard set-up by splitting labels into training and testing and forming training tasks by randomly drawing labels from the training set. At evaluation time, we draw from the test set.

**Model Structure:** Our model first maps each token to an embedding of dimension $200$ before passing it through an LSTM of two layers of $200$ units each. The LSTM emits an output embedding, which is scored against all items of the vocabulary via dot product followed by a softmax. We build the vocabulary from the tokens in the meta-training set and fix its length to $10,000$. We use a sequence length of $10$ for the LSTM and, just as (McMahan et al., 2018), we evaluate using `AccuracyTop1` (i.e., we only consider the predicted word to which the model assigned the highest probability) and

consider all predictions of the unknown token as incorrect. For Omniglot, we use the architecture from Nichol et al. (2018) to also match the one from Finn et al. (2017). We evaluate in the standard transductive setting.

**Hyperparameters:** For the language-modeling experiments, we tune the hyperparameters on the set of meta-validation tasks. For all datasets and all versions of the meta-learning algorithm, we tune hyperparameters in a two step process. We first tune all the parameters that are not related to refinement: the meta learning rate, the local (within-task) meta-training learning rate, the maximum gradient norm, and the decay constant. Then, we use the configuration with the best accuracy pre-refinement and then tune the refinement parameters: the refine learning rate, refine batch size, and refine epochs.

All other hyperparameters are kept fixed for the sake of comparison: full batch steps were taken on within-task data, with the maximum number of microbatches used for the task-global DP model. The parameter search spaces are given in Tables 2, 3, 4. In these tables, the final hyperparameters we used are in bold.

For Omniglot, we largely based our hyperparameters on the choices of Nichol et al. (2018) for 5-way classification. We vary $m$, the number of training shots, but we continue to take 5 SGD steps of size $m$ within task and we leave the test-time SGD procedure exactly the same. However, we do tune for privacy clipping thresholds $\{0.001, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$, Adam Learning Rates $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$, and meta-batch sizes of $\{5, 15, 25, 50\}$.

Table 2: Hyperparameter Search Space for Non-Private Training

| Hyperparameter | Shakespeare-800 | Wiki-1600 |
|---|---|---|
| Visits Per Task | $\{1, 2, 3, 4, 5, 6, \mathbf{7}, 8, 9\}$ | $\{1, \mathbf{2}, 3\}$ |
| Tasks Per Round | $\{\mathbf{5}, 10\}$ | $\{\mathbf{5}, 10\}$ |
| Within-Task Epochs | $\{1, 3, \mathbf{5}, 7, 9\}$ | $\{1, \mathbf{3}, 5, 7, 9\}$ |
| Meta LR | $\{1, \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, \mathbf{8}, 8\sqrt{2}\}$ | $\{1, \sqrt{2}, 2, \mathbf{2\sqrt{2}}, 4, 4\sqrt{2}, 8, 8\sqrt{2}\}$ |
| Meta Decay Rate | $\{0, 0.001, 0.005, \mathbf{0.01}, 0.025, 0.05, 0.1\}$ | $\{0, \mathbf{0.001}, 0.005, 0.01, 0.025\}$ |
| Within-Task LR | $\{\sqrt{2}, \mathbf{2}, 2\sqrt{2}, 4, 4\sqrt{2}, 8\}$ | $\{1, \sqrt{2}, \mathbf{2}, 2\sqrt{2}, 4, 4\sqrt{2}, 8\}$ |
| $L_2$ Clipping | $\{0.3, \mathbf{0.5}, 0.7, 0.8, 1.0\}$ | $\{0.3, 0.5, 0.7, 0.8, \mathbf{1.0}\}$ |
| Refine LR | $\{0.1, \mathbf{0.15}, 0.3, 0.5, 0.7, 0.8\}$ | $\{\mathbf{0.1}, 0.15, 0.3, 0.5, 0.7, 0.8\}$ |
| Refine Mini-batch Size | $\{10, 20, \mathbf{30}, 60\}$ | $\{\mathbf{10}, 20, 30, 60, 120\}$ |
| Refine Epochs | $\{\mathbf{1}, 2, 3\}$ | $\{\mathbf{1}, 2, 3\}$ |

Table 3: Hyperparameter Search Space for Task-Global DP Training

| Model | Shakespeare-800 | Wiki-1600 |
|---|---|---|
| Visits Per Task | $\mathbf{1}$ | $\mathbf{1}$ |
| Tasks Per Round | $\{\mathbf{5}, 10\}$ | $\{5, \mathbf{10}\}$ |
| Within-Task Epochs | $\mathbf{1}$ | $\mathbf{1}$ |
| Meta LR | $\{\sqrt{2}, 2, \mathbf{2\sqrt{2}}, 4, 4\sqrt{2}, 8, 8\sqrt{2}\}$ | $\{1, \sqrt{2}, 2, 2\sqrt{2}, 4, \mathbf{4\sqrt{2}}, 8\}$ |
| Meta Decay Rate | $\{0, \mathbf{0.001}, 0.005, 0.01, 0.025, 0.05, 0.1\}$ | $\{0.0, 0.001, 0.005, 0.01, \mathbf{0.025}, 0.05\}$ |
| Within-Task LR | $\{1, \sqrt{2}, 2\sqrt{2}, 4, 4\sqrt{2}, \mathbf{8}\}$ | $\{1, \sqrt{2}, 2\sqrt{2}, 4, \mathbf{4\sqrt{2}}, \mathbf{8}\}$ |
| $L_2$ Clipping | $\{0.4, 0.5, 0.6, 0.7, 0.8, \mathbf{0.9}, 1.0\}$ | $\{0.3, 0.4, 0.5, 0.7, 0.8, \mathbf{0.9}, 1.0\}$ |
| Refine LR | $\{0.1, 0.15, 0.3, \mathbf{0.5}, 0.7, 0.8\}$ | $\{0.1, 0.15, 0.3, \mathbf{0.5}, 0.7, 0.8\}$ |
| Refine Mini-batch Size | $\{10, \mathbf{20}, 30, 60\}$ | $\{10, 20, 30, \mathbf{60}, 120\}$ |
| Refine Epochs | $\{1, 2, \mathbf{3}\}$ | $\{1, \mathbf{2}, 3\}$ |

Table 4: Hyperparameter Search Space for Local-DP Training

| Model | Shakespeare-800 | Wiki-1600 |
|---|---|---|
| Visits Per Task | $\mathbf{1}$ | $\mathbf{1}$ |
| Tasks Per Round | $\{\mathbf{5}, 10, 20\}$ | $\{10, \mathbf{20}, 40, 80\}$ |
| Within-Task Epochs | $\{1, \mathbf{2}, 3\}$ | $\{1, \mathbf{2}, 3\}$ |
| Meta LR | $\{\sqrt{2}, 2, 2\sqrt{2}, 4, \mathbf{4\sqrt{2}}, 8, 8\sqrt{2}\}$ | $\{2\sqrt{2}, 4, \mathbf{4\sqrt{2}}, 8, 8\sqrt{2}\}$ |
| Meta Decay Rate | $\{0, 0.005, 0.01, 0.025, \mathbf{0.05}, 0.1\}$ | $\{0.0, 0.001, 0.005, 0.01, \mathbf{0.025}, 0.05\}$ |
| Within-Task LR | $\{1, \sqrt{2}, 2\sqrt{2}, \mathbf{4}, 4\sqrt{2}, 8\}$ | $\{1, \sqrt{2}, \mathbf{2\sqrt{2}}, 4, 4\sqrt{2}, 8\}$ |
| $L_2$ Clipping | $\{0.005, \mathbf{0.01}, 0.025, 0.05, 0.1, 0.25, 0.5\}$ | $\{0.005, 0.01, 0.025, \mathbf{0.05}, 0.1, 0.25\}$ |
| Refine LR | $\{0.1, 0.15, 0.3, 0.5, \mathbf{0.7}, 0.8\}$ | $\{0.1, 0.15, 0.3, 0.5, \mathbf{0.7}, 0.8\}$ |
| Refine Mini-batch Size | $\{\mathbf{10}, 20, 30, 60\}$ | $\{\mathbf{10}, 20, 30, 60, 120\}$ |
| Refine Epochs | $\{1, \mathbf{2}, 3\}$ | $\{1, 2, \mathbf{3}\}$ |