UPCYCLED-FL: IMPROVING ACCURACY AND PRIVACY WITH LESS COMPUTATION IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) is a distributed learning paradigm that allows multiple decentralized edge devices to collaboratively learn toward a common objective without sharing local data. However, even though local data is not exposed directly, privacy concerns nonetheless exist as sensitive information can be inferred from intermediate computations. As the same data is repeatedly used over an iterative process, information leakage accumulates substantially over time, making it difficult to balance the trade-off between privacy and accuracy. In this paper we introduce `Upcycled-FL`, a novel federated learning framework, where first-order approximation is applied at every even iteration. Under such a scheme, half of the steps incur no privacy loss and require much less computation. This means less noise injection is needed for a given privacy guarantee, which in turn leads to higher accuracy. Theoretically, we establish the convergence rate performance of `Upcycled-FL` and provide privacy analysis based on objective and output perturbations. Experiments on real-world data show that `Upcycled-FL` consistently outperforms existing methods over heterogeneous data, and significantly improves privacy-accuracy trade-off, while reducing 48% of the training time on average.

1 INTRODUCTION

Federated learning (FL) has emerged as an important paradigm for learning models in a distributed fashion, whereby data is distributed across different edge devices and the goal is to jointly learn from the distributed data. This is facilitated by a central server and learning is conducted through an iterative process of interactions between the central server and local devices: at each iteration, each device performs certain computation using its local data; the local results are collected and aggregated by the central server; the aggregated result is then sent to local devices and used to update local results; and so on till the learning task is deemed accomplished.

Although the data of each device is not shared directly with the central server, sensitive information is nonetheless exposed by making inferences from the intermediate local computations. It is thus critical to ensure the learning process is privacy-preserving. Many techniques have been proposed to protect device's privacy in FL, including anonymization-based methods (e.g., k-anonymity (Samarati & Sweeney, 1998)), perturbation-based (e.g., differential privacy (Dwork, 2006)), and encryption-based methods (e.g., homomorphic encryption, secure multi-party computation). We consider differential privacy (DP) in our framework because: (i) it allows rigorous quantification of the total privacy leakage and is suitable for complex algorithms and tools such as FL; (ii) it can defend against attackers regardless of their background knowledge; (iii) it can provide heterogeneous privacy guarantees for devices with less computational costs; and (iv) it can be tailored to different types of privacy attacks such as membership/attribute inference attacks.

Differential privacy has been widely used in federated learning to provide privacy guarantees. Specifically, Zhang et al. (2022) uses the Gaussian mechanism for a federated learning problem and propose an incentive mechanism to encourage users to share their data and participate in the training process. Zheng et al. (2021) introduces f-differential privacy, a generalized version of Gaussian differential privacy, and propose a federated learning algorithm satisfying this new notion. Wang et al. (2020b) proposes a new mechanism called Random Response with Priori (RRP) to achieve local differential privacy and apply this mechanism to the text data by training a Latent Dirichlet Allocation (LDA) model using a federated learning algorithm. Triastcyn & Faltings (2019) adapts the Bayesian privacy

accounting method to the federated setting and propose joint accounting method for estimating client-level and instance-level privacy simultaneously and securely. Wei et al. (2020) presents a private scheme that adds noise to parameters at the random selected devices before aggregating and provides a convergence bound. Kim et al. (2021) combines the Gaussian mechanism with gradient clipping in federated learning to improve the privacy-accuracy tradeoff. Asoodeh et al. (2021) considers a different setting where only the last update is publicly released and the central server and other devices are assumed to be trustworthy.

In all these studies, the local data is used in every update and privacy leakage occurs at every iteration. While some of these studies aim to improve the privacy-accuracy trade-off, most have focused on finding a tighter bound on privacy loss (by adopting a different privacy notion/mechanism or privacy analysis tool), while the algorithmic property of the underlying federated learning remains unchanged. By contrast, our study aims to improve the privacy-accuracy trade-off by **modifying the federated learning algorithm itself**; this improvement on the algorithmic property is independent of the privacy notion/mechanism or the analysis method. Controlling information leakage by modifying an algorithm was also proposed in Zhang et al. (2018) in a fully distributed setting (i.e., without central server) in the case of the Alternating Direction Method of Multipliers (ADMM) framework, where data across devices are i.i.d. and the objective is convex. By contrast, in this paper we address heterogeneity across data and non-convex optimization objectives, in a federated learning setting.

Specifically, we propose a novel federated learning framework called Upcycled Federated Learning (`Upcycled-FL`)[1], in which privacy leakage is controlled such that it only occurs during *half* of the updates. This is attained by modifying the *even* iterations of the learning algorithm with first-order approximation, which allows us to compute the update using existing results from previous iterations without using data. Moreover, the updates in even iterations only involve addition/subtraction operations on existing results, the computational costs can be reduced significantly.

We emphasize that the idea of upcycling is orthogonal to (1) the baseline FL algorithm, and (2) the DP algorithm. In this paper we adopt `FedProx`(Li et al., 2020) as the baseline framework, but the essence of our proposed updating rule can be applied to other FL frameworks, such as `FedNova` (Wang et al., 2020a) and `pFedMe` (T Dinh et al., 2020). We develop two private DP-enhanced `Upcycled-FL` algorithms by adopting objective perturbation/output perturbation as examples and quantify their privacy loss, while noting that most of the general DP algorithms can also be embedded in `Upcycled-FL`.

It's worth mentioning that, empirically `Upcycled-FL` also outperforms existing baseline algorithm under device and statistical heterogeneity. In real-world scenarios, local data are often non-identically distributed across different devices; different devices are also often equipped with different specifications and computation capabilities. Such heterogeneity often causes instability in the model performance and leads to divergence. Many approaches have been proposed to tackle this issue. For example, `FedAvg` (McMahan et al., 2017) uses a random selection of devices at each iteration to reduce the negative impact of statistical heterogeneity; however, it may fail to converge when heterogeneity increases. Other methods include `FedProx` (Li et al., 2020), a generalization and re-parameterization of `FedAvg` that adds a proximal term to the objective function to penalize deviations in the local model from the previous aggregation, and `FedNova` (Wang et al., 2020a) that re-normalizes local updates before updating to eliminate objective inconsistency. It turns out that `Upcycled-FL` exhibits superior performance in the presence of heterogeneity because gradients encapsulate information on data heterogeneity, the reusing of which leads to a boost in performance.

Our main contributions are summarized as follows.

- We propose `Upcycled-FL` (Algorithm 2), a novel federated learning framework in which information leakage only happens during half of updates and can better handle heterogeneous data.
- We conduct convergence analysis (Section 4, Theorem 4.6) and identify a sufficient condition for the convergence of `Upcycled-FL`.
- We develop a private version of `Upcycled-FL` using output perturbation and objective perturbation (Section 5, Theorem 5.2) and conduct privacy analysis to quantify the privacy guarantee.

[1]The word "upcycle" refers to reusing material so as to create higher-quality things than the original.

- We evaluate `Upcycled-FL` and `Private Upcycled-FL` on real-world data (Section 6). Results show that the proposed algorithms significantly outperform the existing algorithms in terms of the accuracy-privacy trade-off and especially for heterogeneous data.

2 Problem Formulation and Preliminaries

Consider a FL system consisting of a central server and a set $\mathcal{I}$ of agents. Each agent i has its local data $\mathcal{D}_i$ and they can be non-i.i.d across the agents. The goal of FL is to learn a model $\omega \in \mathbb{R}^d$ from data $\cup_{i\in\mathcal{I}}\mathcal{D}_i$ by solving the following optimization:

$$\min_\omega f(\omega) := \sum_{i\in\mathcal{I}} p_i F_i(\omega; \mathcal{D}_i) := \mathbb{E}_i(F_i(\omega; \mathcal{D}_i)), \tag{1}$$

where $p_i = \frac{|\mathcal{D}_i|}{\sum_{j\in\mathcal{I}}|\mathcal{D}_j|}$ is the size of agent i's data as a fraction of the total data samples, $\mathbb{E}_i(\cdot)$ is defined as the expectation over agents, $F_i(\omega; \mathcal{D}_i)$ is the local loss function associated with agent i and depends on local dataset $\mathcal{D}_i$. In this work, we allow $F_i(\omega; \mathcal{D}_i)$ to be possibly non-convex.

FL Algorithm. Let ω_i^t be agent i's local model parameter at time t. In FL, the model is learned through an iterative process: at each time step t, (1) *local computations:* each active agent updates its local model ω_i^t using its local data $\mathcal{D}_i$; (2) *local models broadcasts:* local models (or gradients) are then uploaded to the central server; (3) *model aggregation:* the central server aggregates results received from agents to update the global model parameter $\overline{\omega}^t = \sum_{i\in\mathcal{I}} p_i\omega_i^t$; (4) *model updating:* the aggregated model is sent back to agents and is used for updating local models at $t+1$. In this work, we will focus on `FedProx` framework and propose our algorithm based on it. The details of `FedProx` are given in Algorithm 1.

Algorithm 1 `FedProx` (Li et al., 2020)

1: **Input:** $\mu > 0$, $\{\mathcal{D}_i\}_{i\in\mathcal{I}}$, $\overline{\omega}^0$
2: **for** $t = 1, 2, ..., T$ **do**
3: The central server sends the current global model parameter $\overline{\omega}^t$ to all the agents.
4: A subset of agents get active and each active agent updates its local model by finding (approximate) minimizer of local loss function:

$$\omega_i^{t+1} = \arg\min_\omega F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^t||^2.$$

5: Each agent sends its local model to server.
6: The central server updates the global model by aggregating all local models:

$$\overline{\omega}^{t+1} = \sum_{i\in\mathcal{I}} p_i\omega_i^{t+1}.$$

During the iterative procedure, each agent's local computations are exposed to third parties: its models/gradients need to be uploaded to central server, and the global model calculated based on them are shared with all agents. Privacy concerns arise as private data, though not exposed directly, may nonetheless be inferred from this process. Thus, it is critical to develop private FL algorithms that attain high accuracy. In this work, we adopt differential privacy as the notion of privacy.

Differential Privacy (Dwork, 2006). Differential privacy (DP) centers around the idea that the output of a certain computational procedure should be statistically similar given singular changes to the input, thereby preventing meaningful inference from observing the output. To illustrate the guarantee of DP, consider an attacker aiming at inferring private information of a target individual, whose data may or may not have contributed to a computation. The attacker is able to observe the computational outcome, and may have access to any arbitrary side information DP ensures that regardless of what side information the attacker has, they can learn almost nothing new about the target individual from the computational outcome.

In FL, the information exposed by each agent i includes all intermediate computations $\{\omega_i^t\}_{t=1}^{2M}$. Consider a randomized FL algorithm $\mathcal{A}(\cdot)$ that generates a sequence of private local models $\{\widehat{\omega}_i^t\}_{t=1}^{2M}$, we say it satisfies DP for agent i over $2M$ iterations if the following holds for any possible output $O \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$, and for any two neighboring local datasets $\mathcal{D}_i, \mathcal{D}_i'$:

$$\Pr(\{\widehat{\omega}_i^t\}_{t=0}^{2M} \in O|\mathcal{D}_i) \leq \exp(\varepsilon) \cdot \Pr(\{\widehat{\omega}_i^t\}_{t=0}^{2M} \in O|\mathcal{D}_i') + \delta.$$

where $\varepsilon \in [0, \infty)$ bounds the privacy loss, and $\delta \in [0, 1]$ loosely corresponds to the probability that algorithm fails to bound the privacy loss by ε. Two datasets are neighboring datasets if they are different in at most one data point.

3 PROPOSED ALGORITHM: UPCYCLED-FL

Main idea: Fundamentally, the accumulation of privacy loss over iterations stems from the fact that the agent's data $\mathcal{D}_i$ is used in every update. If the updates can be made without directly using this original data, but only from computational results that already exist, then the privacy loss originating from these updates will be zero, and meanwhile the computational cost may be reduced significantly. Based on this idea, we propose `Upcycled-FL`, which modifies `FedProx` such that the earlier computations are repeatedly used for new updates.

Upcycling Information: Next, we present `Upcycled-FL` and illustrate how total information leakage is reduced under this method.

We modify `FedProx` by applying *first-order approximation* to **even** iterations. Specifically, at even iteration $2m$, we expand $F_i(\omega; \mathcal{D}_i)$ at ω_i^{2m-1} (local model in the previous iteration):

$$F_i(\omega; \mathcal{D}_i) = F_i(\omega_i^{2m-1}; \mathcal{D}_i) + \nabla F_i(\omega_i^{2m-1}; \mathcal{D}_i)^T(\omega - \omega_i^{2m-1}) + \mathcal{O}(||\omega - \omega_i^{2m-1}||^2),$$

We approximate term $\mathcal{O}(||\omega - \omega_i^{2m-1}||^2)$ by $\frac{\lambda_m}{2}||\omega - \omega_i^{2m-1}||^2$ for some constant $\lambda_m \geq 0$, then the even update under such an approximation becomes:

$$\begin{aligned}
\omega_i^{2m} &= \arg\min_{\omega} F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^{2m-1}||^2 \\
&= \arg\min_{\omega} \nabla F_i(\omega_i^{2m-1}; \mathcal{D}_i)^T\omega + \frac{\mu}{2}||\omega - \overline{\omega}^{2m-1}||^2 + \mathcal{O}(||\omega - \omega_i^{2m-1}||^2) \\
&\approx \arg\min_{\omega} \nabla F_i(\omega_i^{2m-1}; \mathcal{D}_i)^T\omega + \frac{\mu}{2}||\omega - \overline{\omega}^{2m-1}||^2 + \frac{\lambda_m}{2}||\omega - \omega_i^{2m-1}||^2. \qquad (2)
\end{aligned}$$

Note that in the above even update, the only term that depends on dataset $\mathcal{D}_i$ is $\nabla F_i(\omega_i^{2m-1}; \mathcal{D}_i)$, which can be derived directly from the previous odd iteration. Specifically, according to first-order condition, the following holds at odd iterations:

$$\omega_i^{2m-1} = \arg\min_{\omega} F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^{2m-2}||^2 \Longrightarrow \nabla F_i(\omega_i^{2m-1}; \mathcal{D}_i) + \mu(\omega_i^{2m-1} - \overline{\omega}^{2m-2}) = 0.$$

Plug $\nabla F_i(\omega_i^{2m-1}; \mathcal{D}_i)$ into even updates equation 2, we have

$$\omega_i^{2m} \approx \arg\min_{\omega} \mu(\overline{\omega}^{2m-2} - \omega_i^{2m-1})^T\omega + \frac{\lambda_m}{2}||\omega - \omega_i^{2m-1}||^2 + \frac{\mu}{2}||\omega - \overline{\omega}^{2m-1}||^2. \qquad (3)$$

Solving equation 5 by first-order condition, even updates are reduced to:

$$\omega_i^{2m} \approx \omega_i^{2m-1} + \frac{\mu}{\mu + \lambda_m}(\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}). \qquad (4)$$

It turns out that with first-order approximation, dataset $\mathcal{D}_i$ is not used in the even updates. Instead, the even updates only involve *addition/subtraction operations* on the existing results from previous iterations (i.e., $\omega_i^{2m-1}, \overline{\omega}^{2m-1}, \overline{\omega}^{2m-2}$): the computational cost is reduced significantly. Note that the first-order approximation is only applied to even iterations, the odd iterations should remain the same as `FedProx` to ensure Eqn. equation 3 holds. The entire updating procedure of `Upcycled-FL` is summarized in Algorithm 2.

Because $\mathcal{D}_i$ is only used in odd iterations, information leakage only happens during odd updates. Intuitively, the reduced information leakage would require less perturbation to attain a certain level of privacy guarantee, which further results in the higher accuracy and improved privacy-accuracy tradeoff. In the following sections, we first analyze the convergence property of `Upcycled-FL` and then introduce the privacy mechanisms to satisfy differential privacy.

4 CONVERGENCE ANALYSIS

In this section, we analyze the convergence property of `Upcycled-FL`. Note that we do not require local functions $F_i(\cdot)$ to be convex. Moreover, we consider practical settings where data are *non-i.i.d* across different devices. Similar to Li et al. (2020), we introduce a measure below to quantify the dissimilarity between devices in network.

Definition 4.1 (B-Dissimilarity (Li et al., 2020))**.** *The local function F_i is B-dissimilar if $\forall \omega$, we have $\mathbb{E}_i[||\nabla F_i(\omega)||^2] \leq ||\nabla f(\omega)||^2 B^2$.*

where $\mathbb{E}_i(\cdot)$ denotes the expectation over devices (Eqn. 1). Parameter $B \geq 1$ captures the statistical heterogeneity across different devices: when all devices are homogeneous with i.i.d data, we have $B = 1$ for all local functions; the larger value of B, the more dissimilarity among devices.

Assumption 4.2. *Local functions F_i are B-dissimilar and L-Lipschitz smooth.*

Note that B-dissimilarity can be satisfied if the divergence between the gradient of the local function and that of the aggregated global function is bounded, as stated below.

Lemma 4.3. *$\forall i$, there exists B such that F_i is B-dissimilar if $||\nabla F_i(\omega) - \nabla f(\omega)|| \leq \kappa_i, \forall \omega$ for some κ_i.*

Assumption 4.4. *$h_i(\omega; \overline{\omega}^t) := F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^t||^2$ are ρ-strongly convex.*

The above assumptions are fairly standard. They first appeared in Li et al. (2020) and are adopted in subsequent works.

Algorithm 2 Proposed framework: `Upcycled-FL`

1: **Input:** $\lambda_m > 0$, $\mu > 0$, $\{\mathcal{D}_i\}_{i \in \mathcal{I}}$, $\overline{\omega}^0$
2: **for** $m = 1, 2, ..., M$ **do**
3: The central server sends the current global model parameter $\overline{\omega}^{2m-2}$ to all the agents.
4: A subset of agents are selected to be active and each active agent updates its local model by finding minimizer of local loss function:

$$\omega_i^{2m-1} \leftarrow \arg\min_{\omega} F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^{2m-2}||^2.$$

5: Agents send local models to central server.
6: The central server updates the global model by aggregating all local models:

$$\overline{\omega}^{2m-1} = \textstyle\sum_{i \in \mathcal{I}} p_i \omega_i^{2m-1}.$$

7: The central server sends the aggregated global model parameter $\overline{\omega}^{2m-1}$ to all the agents.
8: Each active agent updates its local model using existing information

$$\omega_i^{2m} \leftarrow \omega_i^{2m-1} + \frac{\mu}{\mu + \lambda_m}(\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}).$$

9: Active agents send their local models to central server, who then updates the global model by aggregating all local models

$$\overline{\omega}^{2m} = \textstyle\sum_{i \in \mathcal{I}} p_i \omega_i^{2m}.$$

Note that strongly convex assumption is not on local objective $F_i(\omega; \mathcal{D}_i)$, but the regularized function $F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^t||^2$, i.e., the assumption can be satisfied by selecting a sufficiently large $\mu > 0$. As shown in Section 6, our algorithm still converges empirically even when Assumption 4.2, 4.4 don't hold (e.g., DNN). Next, we provide theoretical guarantee on the convergence of `Upcycled-FL` by showing that the global objective function decreases over two consecutive odd iterations.

Lemma 4.5. *Let $\mathcal{S}_m$ be the set of K randomly selected local devices got updated (i.e., active devices) at iterations $2m-1$ and $2m$, and $\mathbb{E}_{\mathcal{S}_m}[\cdot]$ be expectation with respect to the choice of devices. Then under Assumptions 4.2 and 4.4, we have*

$$\mathbb{E}_{\mathcal{S}_m}[f(\overline{\omega}^{2m+1})] \leq f(\overline{\omega}^{2m-1}) - \boldsymbol{C_1}||\nabla f(\overline{\omega}^{2m-1})||^2 + \boldsymbol{C_2} h_m^1 + \boldsymbol{C_3} h_m^2,$$

where $h_m^1 = ||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}||$, $h_m^2 = ||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}||^2$. The details of $\boldsymbol{C_1}$, $\boldsymbol{C_2}$, $\boldsymbol{C_3}$ (expressed as functions of $L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}, \frac{\mu}{\lambda_m}$) are in Appendix D Equation 10 - 12.

Lemma 4.5 characterizes the relation of values of objective function over *two consecutive odd* iterations. It is easy to verify $\boldsymbol{C2}, \boldsymbol{C3} \geq 0$. By rearranging and telescoping, we get the following convergence rate of `Upcycled-FL`.

Theorem 4.6 (Convergence rate of `Upcycled-FL`)**.** *Under the same assumptions of Lemma 4.5, if $\boldsymbol{C1} > 0$, we have*

$$\begin{aligned}\min_{m \in [M]} \mathbb{E}[||\nabla f(\overline{\omega}^{2m-1})||^2] &\leq \frac{1}{M} \sum_{m=0}^{M} \mathbb{E}[||\nabla f(\overline{\omega}^{2m-1})||^2] \\ &\leq \frac{f(\omega^0) - f(\omega^*)}{M\boldsymbol{C1}} + \frac{\sum_{m=0}^{M} \boldsymbol{C_2} h_m^1}{M\boldsymbol{C_1}} + \frac{\sum_{m=0}^{M} \boldsymbol{C_3} h_m^2}{M\boldsymbol{C_1}},\end{aligned}$$

where ω^0 and ω^* denote the initial and the optimal model parameters.

Theorem 4.6 indicates tunable μ and λ_m are key parameters that control the convergence (rate) and robustness of `Upcycled-FL`. Specifically, μ penalizes the deviation of local model ω_i^{2m} from global

aggregated model $\overline{\omega}^{2m-1}$, while λ_m penalizes the deviation of local model ω_i^{2m} from local model in the previous update ω_i^{2m-1}. Note that the condition $\boldsymbol{C_1} := C1\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}\Big) > 0$ does not depend on λ_m, i.e., with the proper choice of local functions F_i and proximal term μ, Theorem 4.6 will hold for **any** λ_m. However, λ_m affects convergence rate by impacting $\boldsymbol{C_2} := C_2\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}, \frac{\mu}{\mu+\lambda_m}\Big)$ and $\boldsymbol{C_3} := C_3\Big(L, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}, \frac{\mu}{\mu+\lambda_m}\Big)$. Specifically, as $\frac{\lambda_m}{\mu}$ gets larger, $\boldsymbol{C_2}$ and $\boldsymbol{C_3}$ become smaller, thus convergence rate bound is tighter. We will empirically examine the impacts of μ and λ_m in Section 6.

The convergence rate also depends on data heterogeneity, captured by dissimilarity B. When $B = 0$ (i.i.d. clients), $\boldsymbol{C_1} > 0$ must hold, while as B increases, $\boldsymbol{C_1}$ becomes negative. Nevertheless, in Section 6 we empirically show `Upcycled-FL` will still converge on highly heterogeneous datasets.

Assumption 4.7. $||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}|| \leq h, \forall m$ *and* $||\nabla f(\omega)|| \leq d, \forall \omega$.

Assumption 4.7 is rather mild; it only requires that the difference of aggregated weights between two consecutive iterations and the gradient $||\nabla f(\omega)||$ are bounded. Under this assumption, we have the following corollary.

Corollary 4.8 (Convergence to the stationary point)**.** *Under the same assumptions of Theorem 4.6 and Assumption 4.7, for fixed μ, K, if λ_m is taken such that $\frac{\mu}{\mu+\lambda_m} = \mathcal{O}\Big(\frac{1}{\sqrt{M}}\Big)$, then the convergence rate of `Upcycled-FL` reduces to $\mathcal{O}(\frac{1}{\sqrt{M}})$.*

Corollary 4.8 provides a guidance on how to choose the value of λ_m to guarantee the convergence of `Upcycled-FL`, i.e., by taking an increasing sequence of $\{\lambda_m\}_{m=1}^M$. Intuitively, increasing λ_m during the training helps stabilize the algorithm, because the deviation of local models from previous update is penalized more under a larger λ_m.

5 PRIVATE UPCYCLED-FL

In this section, we present a privacy-preserving version of `Upcycled-FL`. Many perturbation mechanisms can be adopted to achieve differential privacy such as *objective perturbation* (Chaudhuri et al., 2011; Kifer et al., 2012), *output perturbation* (Chaudhuri et al., 2011; Zhang et al., 2017), *gradient perturbation* (Bassily et al., 2014; Wang et al., 2017), etc. Next, we will use output perturbation and objective perturbation for illustrating that our algorithm has a better performance than Private `FedProx` and Private `FedAvg`(other methods can also be adopted as discussed in Section 7). Note that both methods are used for generating private updates at odd iterations, which can be used directly for even updates, as detailed below.

Output perturbation: the private odd updates $\widehat{\omega}_i^{2m-1}$ are generated by first *clipping* the local models ω_i^{2m-1} and then adding a noise random vector n_i^m to the clipped model:

$$\textbf{Clip odd update:} \quad \xi(\omega_i^{2m-1}) = \omega_i^{2m-1} / \max\Big(1, \frac{||\omega_i^{2m-1}||_2}{\tau}\Big)$$

$$\textbf{Perturb with noise:} \quad \widehat{\omega}_i^{2m-1} = \xi(\omega_i^{2m-1}) + n_i^m$$

where parameter $\tau > 0$ is the clipping threshold; the clipping ensures that if $||\omega_i^{2m-1}||_2 \leq \tau$, then update is preserved, otherwise it is scaled to be of norm τ.

Objective perturbation: a random linear term $\langle n_i^m, \omega \rangle$ is added to the objective function in odd $(2m+1)$-th iteration, and the private local model $\widehat{\omega}_i^{2m+1}$ is found by solving a *perturbed* optimization:

$$\textbf{Perturb objective function:} \quad \widehat{\omega}_i^{2m+1} = \arg\min_{\omega} F_i(\omega; \mathcal{D}_i) + \frac{\mu}{2}||\omega - \overline{\omega}^{2m}||^2 + \langle n_i^m, \omega \rangle,$$

Given noisy $\widehat{\omega}_i^{2m-1}$ generated by either method, local models in even iterations can be updated, i.e.,

$$\widehat{\omega}_i^{2m} = \widehat{\omega}_i^{2m-1} + \frac{\mu}{\mu + \lambda_m}(\widehat{\overline{\omega}}^{2m-1} - \widehat{\overline{\omega}}^{2m-2}), \tag{5}$$

where the aggregated models $\widehat{\overline{\omega}}^{2m-1} = \sum_{i \in \mathcal{I}} p_i \widehat{\omega}_i^{2m-1}$ and $\overline{\omega}^{2m-2} = \sum_{i \in \mathcal{I}} p_i \widehat{\omega}_i^{2m-2}$ are all calculated based on the noisy local models.

Privacy Analysis. Next, we conduct privacy analysis and theoretically quantify the total privacy loss of private `Upcycled-FL`. Because even updates are computed directly using already private intermediate results (Eqn. equation 5) without using dataset $\mathcal{D}_i$, no privacy leakage occurs at even iterations. This can be formally stated as the following lemma.

Lemma 5.1. *For any $m = 1, 2, \cdots$, if the total privacy loss up to $2m-1$ can be bounded by ε_m, then the total privacy loss up to the $2m$-th iteration can also be bounded by ε_m.*

Lemma 5.1 is straightforward; it can be derived directly by leveraging a property of differential privacy called *immunity to post-processing* (Dwork et al., 2014), i.e., a differentially private output followed by any data-independent computation remains satisfying differential privacy. In particular, consider settings where local loss function $F_i(\omega_i, \mathcal{D}_i) := \frac{1}{|\mathcal{D}_i|}\sum_{d\in\mathcal{D}_i}\hat{F}_i(\omega_i, d)$ for some $\hat{F}_i$. Then the guarantee of privacy is presented in Theorem 5.2 (output perturbation) and 5.3 (objective perturbation) below. The total privacy loss in the following theorem is composed using moments accountant method (Abadi et al., 2016).

Theorem 5.2. *Consider the private `Upcycled-FL` over $2M$ iterations under output perturbation with noise $n_i^m \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, then for any $\varepsilon \geq \frac{M\tau^2}{2\sigma^2|\mathcal{D}_i|^2}$, the algorithm is (ε, δ)-DP for agent i for*

$$\delta = \exp\left(-\frac{M\tau^2}{2\sigma^2|\mathcal{D}_i|^2}\left(\frac{\varepsilon\sigma^2|\mathcal{D}_i|^2}{M\tau^2} - \frac{1}{2}\right)^2\right).$$

Equivalently, for any $\delta \in [0, 1]$, the algorithm is (ε, δ)-DP for agent i for

$$\varepsilon = 2\sqrt{\frac{M\tau^2}{2\sigma^2|\mathcal{D}_i|^2}\log(\frac{1}{\delta})} + \frac{M\tau^2}{2\sigma^2|\mathcal{D}_i|^2}.$$

Theorem 5.3. *Consider the private `Upcycled-FL` over $2M$ iterations under objective perturbation with noise $n_i^m \sim \exp(-\alpha_i^m||n_i^m||_2)$. Suppose $\hat{F}_i$ is generalized linear model (Iyengar et al., 2019; Bassily et al., 2014)[2] that satisfies $||\nabla\hat{F}_i(\omega; d)|| < u_1,\ |\hat{F}_i''| \leq u_2$. Let feature vectors be normalized such that its norm is no greater than 1, and suppose $u_2 \leq 0.5|\mathcal{D}_i|\mu$ holds. Then the algorithm satisfies $(\varepsilon, 0)$-DP for agent i where $\varepsilon = \sum_{m=0}^{M}\frac{2\alpha_i^m u_1\mu + 2.8u_2}{|\mathcal{D}_i|\mu}$.*

The assumptions on $\hat{F}_i$ are again fairly standard in the literature, see e.g., (Chaudhuri et al., 2011; Zhang & Zhu, 2016; Zhang et al., 2018). Theorem 5.2 and 5.3 show that the total privacy loss experienced by each agent accumulates over iterations and privacy loss only comes from odd iterations. In contrast, if consider differentially private `FedProx`, accumulated privacy loss would come from all iterations. Therefore, to achieve the same privacy guarantee, private `Upcycled-FL` requires much less perturbation per iteration than private `FedProx`. As a result, accuracy can be improved significantly. Experiments in Section 6 show that `Upcycled-FL` significantly improves privacy-accuracy trade-off compared to other methods.

6 EXPERIMENTS

In this section, we empirically evaluate `Upcycled-FL` and compare it with two algorithms: `FedAvg` (McMahan et al., 2017) and `FedProx` (Li et al., 2020). We first consider non-private algorithms to examine the convergence rate and robustness of `Upcycled-FL` against statistical/device heterogeneity. Then, we adopt both output perturbation and objective perturbation to evaluate the performance of `Private Upcycled-FL`.

Experimental setup. We run experiments on both synthetic and real data.

- *Synthetic data:* using the method in Li et al. (2020), we generate `Synthetic(iid)`, `Synthetic(0,0)`, `Synthetic(0.5,0.5)`, `Synthetic(1,1)`, four datasets with increasing statistical heterogeneity.
- *Real data:* we adopt 2 datasets. 1) `Femnist`, a federated version of `Emnist` (Cohen et al., 2017); 2) `Sentiment140`(`Sent140`), a text sentiment analysis task on tweets (Go et al., 2009). Both of the datasets can be obtained from LEAF (Caldas et al., 2018).

[2]In supervised learning, the sample $d = (x, y)$ corresponds to the feature and label pair. Function $\hat{F}_i(\omega, d)$ is generalized linear model if it can be written as a function of $\omega^T x$ and y.

Table 1: Training(testing) accuracy of FedAvg, FedProx, Upcycled-FL under approximate same training time: FedAvg/FedProx/Upcycled-FL are trained over synthetic data for 80/80/160 iterations, and are trained over Femnist,Sent140 for 150/150/300 iterations. For Upcycled-FL, $\lambda = 0.04, 0.12, 0.21, 0.43$ is set based on the value of $\mu = 0.1, 0.3, 0.5, 1$, respectively. Upcycled-FL significantly outperforms FedAvg and FedProx.

Dataset	Proximal term	Accuracy(%)		
		FedAvg	FedProx	Upcycled-FL
Synthetic(iid)	$\mu = 0.1$	97.80%(97.51%)	96.86%(96.34%)	**97.88%(97.51%)**
	$\mu = 0.3$		95.18%(94.88%)	**96.31%(96.05%)**
Synthetic(0,0)	$\mu = 0.1$	78.41%(76.27%)	81.66%(79.50%)	**84.41%(82.44%)**
	$\mu = 0.3$		81.73%(79.35%)	**83.83%(81.58%)**
Synthetic(0.5,0.5)	$\mu = 0.5$	79.53%(78.97%)	79.55%(79.46%)	**82.61%(82.64%)**
	$\mu = 1$		80.11%(80.20%)	**82.74%(82.64%)**
Synthetic(1,1)	$\mu = 0.5$	69.70%(71.12%)	75.36%(76.67%)	**82.78%(83.07%)**
	$\mu = 1$		75.69%(76.58%)	**82.88%(82.69%)**
Femnist	$\mu = 0.1$	20.43%(20.01%)	82.86%(83.73%)	**86.37%(85.91%)**
	$\mu = 0.3$		76.10%(76.66%)	**81.94%(82.12%)**
Sent140	$\mu = 0.1$	73.41%(71.13%)	75.29%(72.57%)	**78.12%(72.38%)**
	$\mu = 0.3$		72.86%(71.38%)	**74.27%(73.02%)**

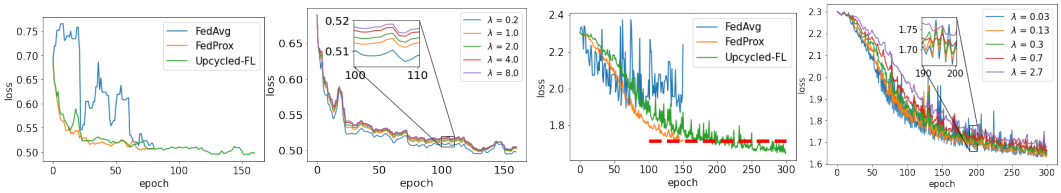
To simulate device heterogeneity, we randomly select a fraction of devices to train at each round, and assume there are stragglers that cannot train for full rounds; both devices and stragglers are selected by random seed to ensure they are the same for all algorithms. We also consider several learning algorithms: we learn *logistic regression* on synthetic data; *multilayer perceptron (MLP)* and deep CNN on Femnist; *Stacked LSTM* on Sent140. For Femnist, we present the results on MLP here and other results in Appendix F. Because even iterations of Upcycled-FL only involve addition/subtraction operations with almost no computational cost, we train Upcycled-FL with **double** iterations compared to FedAvg and FedProx in approximately same training time. Unless explicitly stated, the results we report are averaged outcomes over all devices. More details of experimental setup are in Appendix F.1.

Convergence and Heterogeneity. We evaluate the convergence rate, training time, and accuracy of Upcycled-FL under different parameter settings. We take $\lambda_m = \lambda, \forall m$ while we observe the similar results for time-varying λ_m. Table B illustrates the comparison of training/testing accuracy under high device heterogeneity (90% stragglers), more results are in Appendix F. The results show that Upcycled-FL outperforms both FedProx and FedProx on all datasets. While FedAvg achieves good performance on Synthetic(iid), it is not robust to heterogeneous data. When data is i.i.d., adding the proximal term may hurt the performance (Li et al., 2020). However, the proximal term help stabilize the algorithm and can significantly improve the performance in practical setting when data is heterogeneous. Importantly, Upcycled-FL is more robust to the level of heterogeneity and the choice of hyper-parameter μ than FedProx, that it could attain significant and consistent improvements for all settings. In practice, μ needs to be carefully tuned; with the optimal selections of μ, Upcycled-FL could achieve $2\% \sim 7\%$ improvement on accuracy compared to FedProx.

The reasons why Upcycled-FL outperforms other algorithms are the following: (1) the gradients in odd iterations are reused in the subsequent even iterations, so that more information of data heterogeneity is captured to make better updates; (2) because computations of even updates are highly efficient, it could accommodate more local gradient descent updates within the same training time.

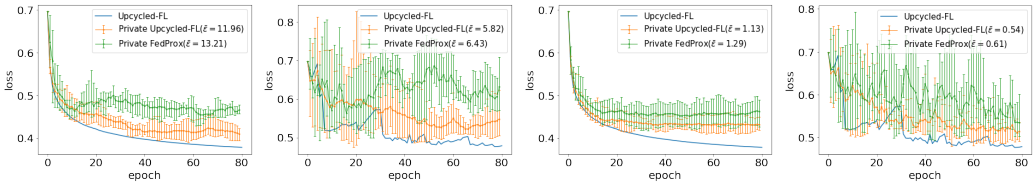
Figure 1(a)1(b) illustrate the convergence of Upcycled-FL on Synthetic(0.5,0.5) and Femnist (in comparison with FedProx and FedAvg), and examine the impact of hyper-parameter λ_m. Loss in y-axes indicates the averaged loss of all devices. In each iteration, 30% of devices are selected with 90% stragglers. It is worth noting that Upcycled-FL could attain a comparable performance as FedProx in the first 80 (resp. 240) iterations on Synthetic(0.5,0.5) (resp. Femnist), with 50% (resp. 20%) of SGD computation reduced. Additional results on other datasets are in Appendix F.4.

The effect of $\lambda_m = \lambda$ on convergence rate of Upcycled-FL is also evaluated in Figure 1(a)1(b) (right). Although the sufficient conditions in Theorem 4.6 and Corollary **??** suggest that Upcycled-FL converges when λ is sufficiently large, the experiments indicate that Upcycled-FL



(a) `Synthetic(0.5,0.5)` (b) `Femnist`

Figure 1: Convergence of `Upcycled-FL` compared to `FedProx`, `FedAvg` under approximate same training time (left) and impact of λ_m on convergence rate of `Upcycled-FL` (right) for two datasets: (a) μ=0.5, λ=0.21, the final training(testing) accuracy is `Upcycled-FL` **82.61%(82.64%)**, `FedProx` 79.55%(79.46%), `FedAvg` 79.53%(78.97%); (b) μ=0.3, λ=0.13, the final training(testing) accuracy is `Upcycled-FL` **81.94%(82.12%)**, `FedProx` 76.10%(76.66%), `FedAvg` 20.43%(20.01%).



(a) `Synthetic(iid)` (b) `Synthetic(0,0)` (c) `Synthetic(iid)` (d) `Synthetic(0,0)`

Figure 2: Comparison of `Private Upcycled-FL` and `Private FedProx` using objective perturbation ((a)(b)) and output perturbation ((c)(d)) with $\delta = 0.001$. We take $\mu = 0.1, \lambda = 0.04$ for all datasets. $\bar{\epsilon}$ denotes the average ϵ among all devices. In both experiments, the privacy of `Private Upcycled-FL` is strictly stronger than `Private FedProx`.

indeed easily converges in practice for small λ. Furthermore, we observe that models with smaller λ tend to converge faster. This is because that λ controls step size $\frac{\mu}{\mu+\lambda}$ (Eqn. equation 4): a larger λ results in smaller step size and thus slower convergence.

Privacy-Accuracy Tradeoff. We next inspect accuracy-privacy tradeoff of `Private Upcycled-FL` and compare it with `Private FedProx`. We adopt both objective perturbation and output perturbation to preserve privacy; these are just two examples while other privacy-preserving techniques can also be used. For each parameter setting, we perform 10 independent runs of experiment and record both the mean and range of their losses. To precisely quantify privacy, we focus on settings without device heterogeneity (no straggler) and low statistical heterogeneity (`Synthetic(iid)`, `Synthetic(0,0)`). More results can be found in F.5.

Figure 2(a)2(b) shows the performance (average loss and the respective privacy loss ϵ) of `Private Upcylced-FL` and `Private FedProx` compared to non-private `Upcylced-FL` on synthetic data using objective perturbation. As expected, `Private Upcycled-FL` is much more stable and significantly improves both privacy and accuracy over `Private FedProx`; this is because information leakage only happens during half of updates in `Private Upcycled-FL` so that it requires much less perturbation to attain the same privacy guarantee. Specifically, **`Private Upcycled-FL` experiences 9.5% less privacy loss than `Private FedProx` while having 11.0%(11.5%) higher accuracy in average and reducing 48% of training time.** We also conduct the similar experiments using output perturbation (Figure 2(d)2(c)). Results show that **`Private Upcycled-FL` experiences 12.0% less privacy loss than `Private FedProx` in average while having 6.1%(6.2%) higher accuracy in average and reducing 48% of training time.**

We also conduct experiments with different α to examine accuracy-privacy tradeoff (the larger α corresponds to the less perturbation and thus higher accuracy). We see that `Private Upcycled-FL` consistently outperforms `Private FedProx`. These results are presented in Appendix F.5.1.

7 Discussion

In Appendix A we discuss an equivalent interpretation of `Upcycled-FL`, other FL framework, other privacy analysis tools and amplification techniques, as well as the limitation of our work.

REFERENCES

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*, pp. 308–318, 2016.

Shahab Asoodeh, Wei-Ning Chen, Flavio P Calmon, and Ayfer Özgür. Differentially private federated learning: An information-theoretic perspective. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 344–349. IEEE, 2021.

Brendan Avent, Aleksandra Korolova, David Zeber, Torgeir Hovden, and Benjamin Livshits. {BLENDER}: Enabling local search with a hybrid differential privacy model. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pp. 747–764, 2017.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: tight analyses via couplings and divergences. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6280–6290, 2018.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473. IEEE, 2014.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94(3):401–437, 2014.

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

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12. Springer, 2006.

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

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Jingchen Hu, Joerg Drechsler, and Hang J Kim. Accuracy gains from privacy amplification through sampling for differential privacy. *arXiv preprint arXiv:2103.09705*, 2021.

Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 299–316. IEEE, 2019.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1. JMLR Workshop and Conference Proceedings, 2012.

Muah Kim, Onur Günlü, and Rafael F Schaefer. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2650–2654. IEEE, 2021.

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.

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*, pp. 1273–1282. PMLR, 2017.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

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

Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2587–2596. IEEE, 2019.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020a.

Yansheng Wang, Yongxin Tong, and Dingyuan Shi. Federated latent dirichlet allocation: A local differential privacy based framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6283–6290, 2020b.

Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pp. 2493–2502. PMLR, 2015.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1226–1235. PMLR, 2019.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3922–3928, 2017.

Lefeng Zhang, Tianqing Zhu, Ping Xiong, Wanlei Zhou, and Philip Yu. A robust game-theoretical federated learning framework with joint differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Tao Zhang and Quanyan Zhu. Dynamic differential privacy for admm-based distributed classification learning. *IEEE Transactions on Information Forensics and Security*, 12(1):172–187, 2016.

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. Recycled admm: Improve privacy and accuracy with less computation in distributed algorithms. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 959–965. IEEE, 2018.

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

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 *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2019. URL https://arxiv.org/abs/1811.12469.

A DISCUSSION

An equivalent model and additional interpretation. Recall the even update $\overline{\omega}^{2m} = \sum_{i\in\mathcal{I}} p_i \omega_i^{2m} = \sum_{i\in\mathcal{I}} p_i \omega_i^{2m-1} + \frac{\mu}{\mu+\lambda_m}(\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2})$. If we view two consecutive steps (odd followed by even) as a single step in the iteration, we can obtain an equivalent model where selected devices train and submit local weights ω_i^m; the central server computes the aggregated weights $\overline{\omega}^m = \sum_{i\in\mathcal{I}} p_i \omega_i^m$ and updates the global model using $\omega_{global}^m = \omega_{global}^{m-1} + \frac{2\mu+\lambda_m}{\mu+\lambda_m}(\overline{\omega}^m - \omega_{global}^{m-1})$. It can be seen that this method effectively allows the global model to not only average but also to move more toward the changing direction. It's worth pointing out that this updating rule is not the same as increasing local learning rate, which does not reuse any existing information, nor will it result in the same updating direction. The illustration of the difference and more discussion are in Appendix F.6.

Other FL framework. We have developed `Upcycled-FL` based on `FedProx`. But the essence of our proposed updating rule can be applied to other FL frameworks, such as `FedNova` (Wang et al., 2020a), `pFedMe` (T Dinh et al., 2020).

Other perturbation methods & privacy analysis tools. In this work, we primarily adopt output/objective perturbation to make `Upcycled-FL` differentially private. This is just an example we use to illustrate how `Upcycled-FL` can improve privacy-accuracy tradeoff. Other perturbation methods can also be used and our conclusion would still hold. Similarly, in privacy analysis we have adopted (ε, δ)-differential privacy and moments accountant method (Abadi et al., 2016) to compose privacy loss. Other privacy notion and composition theorems can also be used. This is because our key idea for improving privacy-accuracy trade-off (i.e., revealing less information) is subject to algorithm itself and is orthogonal to the choice of the perturbation method and the privacy definition/analysis tools.

Other privacy amplification techniques. Our approach improves privacy-accuracy trade-off by directly modifying FL algorithms. It can also be combined with other techniques to further strengthen the privacy protection, such as *privacy amplification by sampling* (Balle et al., 2018; Beimel et al., 2014; Hu et al., 2021; Wang et al., 2019; Kasiviswanathan et al., 2011; Wang et al., 2015; Abadi et al., 2016), *leveraging non-private public data* (Avent et al., 2017; Papernot et al., 2016), *amplification by shuffling* (Úlfar Erlingsson et al., 2019), etc.

Limitations & Negative Societal Impacts. (1) Although `Upcycled-FL` outperforms existing FL framework on heterogeneous data, there's still space to improve accuracy on data that is highly heterogeneous. (2) The hyper-parameter λ could be analysed more precisely. One interesting direction is to take hessian matrix into consideration, and provide more accurate approximation. (3) This work has solely focused on accuracy and privacy, while unfairness issue is not considered. When data is non-i.i.d. across different devices, it is possible that our method may have disparate impact on devices with less data. Incorporating fairness into our method is a potential future direction.

B NOTATION TABLE

$\mathcal{I}$	set of agents
$\mathcal{D}_i$	dataset of agent i
p_i	size of agent i's data as a fraction of total data samples
ω_i^t	agent i's local model parameter at time t
$\overline{\omega}^t$	aggregated model at central server at t
$\widehat{\omega}_i^t$	differentially private version of ω_i^t
F_i	local objective function of agent i
f	overall objective function
n_i^t	random noise added to agent i at time t
μ	hyper-parameter for proximal term in `FedProx` and `Upcycled-FL`
λ_m	hyper-parameter for first-order approximation at even iteration $2m$ in `Upcycled-FL`

C LEMMAS

Lemma C.1. *Define $\widetilde{\omega}^t = \mathbb{E}_i(\omega_i^t)$. Suppose conditions in Theorem 4.6 hold. Then the following holds for all m:*

$$\begin{aligned} f(\widetilde{\omega}^{2m+1}) \quad \leq \quad & f(\overline{\omega}^{2m-1}) - \widehat{C}_1\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}\Big)||\nabla f(\overline{\omega}^{2m-1})||^2 \\ & + \widehat{C}_2\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big)||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}|| \\ & + \widehat{C}_3\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big)||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}||^2 \end{aligned}$$

where coefficients satisfy

$$\begin{aligned} \widehat{C}_1\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}\Big) \quad &= \quad \frac{1}{\mu} - \frac{LB}{\mu^2\rho} - \frac{LB^2}{2\rho^2} \\ \widehat{C}_2\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) \quad &= \quad \Big(\frac{L^2}{\mu^2\rho} + \frac{L+\mu}{\mu^2} + \frac{L(L+\rho)B}{\rho^2}\Big)\frac{\mu}{\mu+\lambda_m} \\ \widehat{C}_3\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) \quad &= \quad \frac{L(L+\rho)^2}{2\rho^2}\frac{\mu^2}{(\mu+\lambda_m)^2} \end{aligned}$$

Lemma C.2. *Let $\mathcal{S}_m$ be the set of K randomly selected local devices got updated at iterations $2m-1$ and $2m$, and $\mathbb{E}_{\mathcal{S}_m}[\cdot]$ be expectation with respect to the choice of devices. Then we have*

$$\begin{aligned} \mathbb{E}_{\mathcal{S}_m}[f(\overline{\omega}^{2m+1})] \quad \leq \quad & f(\widetilde{\omega}^{2m+1}) + \widetilde{C}_1\Big(B, L, \frac{1}{K}, \frac{1}{\rho}\Big)||\nabla f(\overline{\omega}^{2m-1})||^2 \\ & + \widetilde{C}_2\Big(B, L, \frac{1}{K}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big)||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}|| \\ & + \widetilde{C}_3\Big(B, L, \frac{1}{K}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big)||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}||^2 \end{aligned}$$

where coefficients satisfy

$$\begin{aligned} \widetilde{C}_1\Big(B, L, \frac{1}{K}, \frac{1}{\rho}\Big) \quad &= \quad \frac{2B^2}{K\rho^2} + \frac{2LB+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{B}{\rho} \\ \widetilde{C}_2\Big(B, L, \frac{1}{K}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) \quad &= \quad \Big(\frac{4LB}{K\rho^2} + \frac{2LB+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{L}{\rho} + 2L\frac{L+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{B}{\rho}\Big)\cdot\frac{\mu}{\mu+\lambda_m} \\ \widetilde{C}_3\Big(B, L, \frac{1}{K}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) \quad &= \quad \Big(\frac{2}{K}\frac{L^2}{\rho^2} + 2L\frac{L+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{L}{\rho}\Big)\cdot\Big(\frac{\mu}{\mu+\lambda_m}\Big)^2 \end{aligned}$$

D PROOFS

Proof of Lemma C.1

Proof. Since local functions F_i are L-Lipschitz smooth, at iteration $2m-1$, we have

$$\begin{aligned} & f(\widetilde{\omega}^{2m+1}) \\ & \leq f(\overline{\omega}^{2m-1}) + \langle\nabla f(\overline{\omega}^{2m-1}), \widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}\rangle + \frac{L}{2}||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}||^2 \\ & = f(\overline{\omega}^{2m-1}) + \langle\nabla f(\overline{\omega}^{2m-1}), -\frac{1}{\mu}\nabla f(\overline{\omega}^{2m-1}) + \Phi^{2m+1}\rangle + \frac{L}{2}||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}||^2 \\ & \leq f(\overline{\omega}^{2m-1}) - \frac{1}{\mu}||\nabla f(\overline{\omega}^{2m-1})||^2 + \frac{1}{\mu}||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\Phi^{2m+1}|| + \frac{L}{2}||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}||^2 \end{aligned}$$

where

$$\Phi^{2m+1} \quad = \quad \frac{1}{\mu}\nabla f(\overline{\omega}^{2m-1}) + \widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1} = \mathbb{E}_i\Big[\frac{1}{\mu}\nabla F_i(\overline{\omega}^{2m-1}) + \omega_i^{2m+1} - \overline{\omega}^{2m-1}\Big] \quad (6)$$

By first-order condition, the following holds at $(2m+1)$-th iteration:

$$\omega_i^{2m+1} - \overline{\omega}^{2m-1} = -\frac{1}{\mu}\nabla F_i(\omega_i^{2m+1}) + \overline{\omega}^{2m} - \overline{\omega}^{2m-1}$$

Plug into Eqn. equation 6, we have

$$\Phi^{2m+1} = \mathbb{E}_i\Big[\frac{1}{\mu}\Big(\nabla F_i(\overline{\omega}^{2m-1}) - \nabla F_i(\omega_i^{2m+1})\Big) + \overline{\omega}^{2m} - \overline{\omega}^{2m-1}\Big]$$

By L-Lipschitz smoothness and Jensen's inequality, we have

$$\begin{aligned} ||\Phi^{2m+1}|| &\leq \mathbb{E}_i\Big[\frac{1}{\mu}||\nabla F_i(\overline{\omega}^{2m-1}) - \nabla F_i(\omega_i^{2m+1})||\Big] + ||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \\ &\leq \mathbb{E}_i\Big[\frac{L}{\mu}||\overline{\omega}^{2m-1} - \omega_i^{2m+1}||\Big] + ||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \\ &\leq \mathbb{E}_i\Big[\frac{L}{\mu}||\omega_i^{2m+1} - \overline{\omega}^{2m}||\Big] + \frac{L+\mu}{\mu}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \end{aligned} \tag{7}$$

Since h_i are ρ-strongly convex, F_i is L-Lipschitz smooth, and $\omega_i^{2m+1} = \arg\min_\omega h_i(\omega; \overline{\omega}^{2m})$ we have

$$\begin{aligned} ||\omega_i^{2m+1} - \overline{\omega}^{2m}|| &\leq \frac{1}{\rho}||\nabla h_i(\omega_i^{2m+1}; \overline{\omega}^{2m}) - \nabla h_i(\overline{\omega}^{2m}; \overline{\omega}^{2m})|| = \frac{1}{\rho}||0 - \nabla F_i(\overline{\omega}^{2m})|| \\ &\leq \frac{L}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{1}{\rho}||\nabla F_i(\overline{\omega}^{2m-1})|| \end{aligned} \tag{8}$$

Plug in Eqn. equation 7,

$$||\Phi^{2m+1}|| \leq \frac{L}{\mu\rho}\mathbb{E}_i[||\nabla F_i(\overline{\omega}^{2m-1})||] + \Big(\frac{L^2}{\mu\rho} + \frac{L+\mu}{\mu}\Big)||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}||$$

Consider the following term

$$\begin{aligned} ||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}|| &= ||\mathbb{E}_i[\omega_i^{2m+1}] - \overline{\omega}^{2m-1}|| \leq \mathbb{E}_i[||\omega_i^{2m+1} - \overline{\omega}^{2m-1}||] \\ &\leq \mathbb{E}_i[||\omega_i^{2m+1} - \overline{\omega}^{2m}|| + ||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}||] \\ &\leq \frac{L+\rho}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{1}{\rho}\mathbb{E}_i\Big[||\nabla F_i(\overline{\omega}^{2m-1})||\Big] \end{aligned} \tag{9}$$

Because F_i is B-dissimilar, we have

$$\mathbb{E}_i\Big[||\nabla F_i(\overline{\omega}^{2m-1})||\Big] \leq B||\nabla f(\overline{\omega}^{2m-1})||$$

Therefore,

$$\begin{aligned} &f(\widetilde{\omega}^{2m+1}) \\ &\leq f(\overline{\omega}^{2m-1}) - \frac{1}{\mu}||\nabla f(\overline{\omega}^{2m-1})||^2 + \frac{1}{\mu}||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\Phi^{2m+1}|| + \frac{L}{2}||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}||^2 \\ &\leq f(\overline{\omega}^{2m-1}) - \Big(\frac{1}{\mu} - \frac{LB}{\mu^2\rho} - \frac{LB^2}{2\rho^2}\Big)||\nabla f(\overline{\omega}^{2m-1})||^2 \\ &+ \Big(\frac{L^2}{\mu^2\rho} + \frac{L+\mu}{\mu^2} + \frac{L(L+\rho)B}{\rho^2}\Big)||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \\ &+ \frac{L(L+\rho)^2}{2\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}||^2 \end{aligned}$$

After applying first-order approximation in even iterations, we have

$$\overline{\omega}^{2m} - \overline{\omega}^{2m-1} = \frac{\mu}{\mu + \lambda_m}(\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2})$$

Therefore,

$$\begin{aligned} f(\widetilde{\omega}^{2m+1}) &\leq f(\overline{\omega}^{2m-1}) - \Big(\frac{1}{\mu} - \frac{LB}{\mu^2\rho} - \frac{LB^2}{2\rho^2}\Big)||\nabla f(\overline{\omega}^{2m-1})||^2 \\ &+\Big(\frac{L^2}{\mu^2\rho} + \frac{L+\mu}{\mu^2} + \frac{L(L+\rho)B}{\rho^2}\Big)\frac{\mu}{\mu+\lambda_m}||\nabla f(\overline{\omega}^{2m-1})|| \cdot ||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}|| \\ &+\frac{L(L+\rho)^2}{2\rho^2}\frac{\mu^2}{(\mu+\lambda_m)^2}||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}||^2 \end{aligned}$$

The Lemma C.1 is proved. □

Proof of Lemma C.2

Proof. Because local function F_i is L-Lipschitz smooth, f is local Lipschitz continuous.

$$f(\overline{\omega}^{2m+1}) \leq f(\widetilde{\omega}^{2m+1}) + L_0||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||$$

where L_0 is the local Lipschitz continuouty constant. Moreover, we have

$$L_0 \leq ||\nabla f(\overline{\omega}^{2m-1})|| + L(||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}|| + ||\overline{\omega}^{2m+1} - \overline{\omega}^{2m-1}||)$$

Therefore,

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}_m}[f(\overline{\omega}^{2m+1})] \\ &\leq f(\widetilde{\omega}^{2m+1}) \\ &+ \mathbb{E}_{\mathcal{S}_m}\Big[\Big(||\nabla f(\overline{\omega}^{2m-1})|| + L(||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}|| + ||\overline{\omega}^{2m+1} - \overline{\omega}^{2m-1}||)\Big)||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||\Big] \\ &= f(\widetilde{\omega}^{2m+1}) + \Big(||\nabla f(\overline{\omega}^{2m-1})|| + L||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}||\Big) \cdot \mathbb{E}_{\mathcal{S}_m}[||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||] \\ &+ L\mathbb{E}_{\mathcal{S}_m}\Big[||\overline{\omega}^{2m+1} - \overline{\omega}^{2m-1}|| \cdot ||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||\Big] \\ &\leq f(\widetilde{\omega}^{2m+1}) + \Big(||\nabla f(\overline{\omega}^{2m-1})|| + 2L||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}||\Big) \cdot \mathbb{E}_{\mathcal{S}_m}[||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||] \\ &+ L\mathbb{E}_{\mathcal{S}_m}\Big[||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||^2\Big] \end{aligned}$$

When K devices are randomly selected, by Eqn. equation 8, we have

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}_m}\Big[||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||^2\Big] \\ &\leq \frac{1}{K}\mathbb{E}_i\Big[||\omega_i^{2m+1} - \widetilde{\omega}^{2m+1}||^2\Big] \leq \frac{2}{K}\mathbb{E}_i\Big[||\omega_i^{2m+1} - \overline{\omega}^{2m}||^2\Big] \\ &\leq \frac{2}{K}\mathbb{E}_i\Big[\frac{L^2}{\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}||^2 + \frac{1}{\rho^2}||\nabla F_i(\overline{\omega}^{2m-1})||^2 + \frac{2L}{\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \cdot ||\nabla F_i(\overline{\omega}^{2m-1})||\Big] \\ &\leq \frac{2}{K}\frac{L^2}{\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}||^2 + \frac{2B^2}{K\rho^2}||\nabla f(\overline{\omega}^{2m-1})||^2 + \frac{4LB}{K\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \cdot ||\nabla f(\overline{\omega}^{2m-1})|| \\ &= \frac{2}{K}\Big(\frac{L}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{B}{\rho}||\nabla f(\overline{\omega}^{2m-1})||\Big)^2 \end{aligned}$$

By Jensen's inequality,

$$\begin{aligned} \mathbb{E}_{\mathcal{S}_m}\Big[||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||\Big] &\leq \sqrt{\mathbb{E}_{\mathcal{S}_m}\Big[||\overline{\omega}^{2m+1} - \widetilde{\omega}^{2m+1}||^2\Big]} \\ &= \sqrt{\frac{2}{K}}\Big(\frac{L}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{B}{\rho}||\nabla f(\overline{\omega}^{2m-1})||\Big) \end{aligned}$$

By Eqn. equation 9,

$$||\nabla f(\overline{\omega}^{2m-1})|| + 2L||\widetilde{\omega}^{2m+1} - \overline{\omega}^{2m-1}|| \leq 2L\frac{L+\rho}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{2LB+\rho}{\rho}||\nabla f(\overline{\omega}^{2m-1})||$$

Re-organize, we have

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{S}_m}[f(\overline{\omega}^{2m+1})] \\
&\leq f(\widetilde{\omega}^{2m+1}) + \frac{2}{K}\frac{L^2}{\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}||^2 + \frac{2B^2}{K\rho^2}||\nabla f(\overline{\omega}^{2m-1})||^2 \\
&+ \frac{4LB}{K\rho^2}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \cdot ||\nabla f(\overline{\omega}^{2m-1})|| \\
&+ \Big(2L\frac{L+\rho}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{2LB+\rho}{\rho}||\nabla f(\overline{\omega}^{2m-1})||\Big)\sqrt{\frac{2}{K}}\frac{L}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| \\
&+ \Big(2L\frac{L+\rho}{\rho}||\overline{\omega}^{2m} - \overline{\omega}^{2m-1}|| + \frac{2LB+\rho}{\rho}||\nabla f(\overline{\omega}^{2m-1})||\Big)\sqrt{\frac{2}{K}}\frac{B}{\rho}||\nabla f(\overline{\omega}^{2m-1})|| \\
&= f(\widetilde{\omega}^{2m+1}) + \Big(\frac{2}{K}\frac{L^2}{\rho^2} + 2L\frac{L+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{L}{\rho}\Big)\cdot\Big(\frac{\mu}{\mu+\lambda_m}\Big)^2||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}||^2 \\
&+ \Big(\frac{4LB}{K\rho^2} + \frac{2LB+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{L}{\rho} + 2L\frac{L+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{B}{\rho}\Big)\cdot\frac{\mu}{\mu+\lambda_m}||\overline{\omega}^{2m-1} - \overline{\omega}^{2m-2}|| \cdot ||\nabla f(\overline{\omega}^{2m-1})|| \\
&+ \Big(\frac{2B^2}{K\rho^2} + \frac{2LB+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{B}{\rho}\Big)||\nabla f(\overline{\omega}^{2m-1})||^2
\end{aligned}
$$

Lemma C.2 is proved. □

Proof of Lemma 4.5

Proof. Lemma 4.5 can be proved by combing Lemmas C.1 and C.2, where

$$
\begin{aligned}
\boldsymbol{C_1} &:= C_1\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}\Big) = \widehat{C}_1\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}\Big) - \widetilde{C}_1\Big(L, B, \frac{1}{K}, \frac{1}{\rho}\Big) \\
&= \frac{1}{\mu} - \frac{LB}{\mu^2\rho} - \frac{LB^2}{2\rho^2} - \frac{2B^2}{K\rho^2} - \frac{2LB+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{B}{\rho} \qquad (10) \\
\boldsymbol{C_2} &:= C_2\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}, \frac{\mu}{\mu+\lambda_m}\Big) = \widehat{C}_2\Big(L, B, \frac{1}{\mu}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) + \widetilde{C}_2\Big(L, B, \frac{1}{K}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) \\
&= \Big(\frac{L^2}{\mu^2\rho} + \frac{L+\mu}{\mu^2} + \frac{L(L+\rho)B}{\rho^2} + \frac{4LB}{K\rho^2} + \frac{(4L^2B+\rho L(1+2B)}{\rho^2}\sqrt{\frac{2}{K}}\Big)\cdot\frac{\mu}{\mu+\lambda_m} \qquad (11) \\
\boldsymbol{C_3} &:= C_3\Big(L, \frac{1}{\mu}, \frac{1}{\rho}, \frac{1}{K}, \frac{\mu}{\mu+\lambda_m}\Big) = \widehat{C}_3\Big(L, \frac{1}{\mu}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) + \widetilde{C}_3\Big(L, \frac{1}{K}, \frac{1}{\rho}, \frac{\mu}{\mu+\lambda_m}\Big) \\
&= \Big(\frac{L(L+\rho)^2}{2\rho^2} + \frac{2}{K}\frac{L^2}{\rho^2} + 2L\frac{L+\rho}{\rho}\sqrt{\frac{2}{K}}\frac{L}{\rho}\Big)\cdot\Big(\frac{\mu}{\mu+\lambda_m}\Big)^2 \qquad (12)
\end{aligned}
$$

□

E PROOF OF THEOREM 5.2

WLOG, consider the case when local device got updated in every iteration and the algorithm runs over $2M$ iterations in total. We will use the uppercase letters X and lowercase letters x to denote random variables and the corresponding realizations, and use $P_X(\cdot)$ to denote its probability distribution. To simplify the notations, we will drop the index i as we are only concerned with one agent.

According to Abadi et al. (2016), for a mechanism $\mathscr{M}$ outputs o, with inputs d and $\hat{d}$, let a random variable $c(o; \mathscr{M}, d, \hat{d}) = \log\frac{\Pr(\mathscr{M}(d)=o)}{\Pr(\mathscr{M}(\hat{d})=o)}$ denote the privacy loss at o, and

$$\alpha_{\mathscr{M}}(\lambda) = \max_{d,\hat{d}} \log \mathbb{E}_{o\sim\mathscr{M}(d)}\{\exp(\lambda c(o; \mathscr{M}, d, \hat{d}))\}$$

For two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ of agent i, by Lemma 5.1, the total privacy loss is only contributed by odd iterations. Thus, for any sequence of private (clipped) models $\widehat{\omega}^t$ generated by

mechanisms $\{\mathscr{M}^m\}_{m=1}^M$ over $2M$ iterations, there is:

$$\begin{aligned} c(\widehat{\omega}^{0:2M}; \{\mathscr{M}^m\}_{m=1}^M, \mathcal{D}, \mathcal{D}') &= \log \frac{P_{\widehat{\Omega}^{0:2M}}(\widehat{\omega}^{0:2M}|\mathcal{D})}{P_{\widehat{\Omega}^{0:2M}}(\widehat{\omega}^{0:2M}|\mathcal{D}')} \\ &= \sum_{m=0}^{M} \log \frac{P_{\widehat{\Omega}^{2m+1}}(\widehat{\omega}^{2m+1}|\mathcal{D}, \widehat{\omega}^{0:2m})}{P_{\widehat{\Omega}^{2m+1}}(\widehat{\omega}^{2m+1}|\mathcal{D}', \widehat{\omega}^{0:2m})} + \log \frac{P_{\widehat{\Omega}^0}(\widehat{\omega}^0|\mathcal{D})}{P_{\widehat{\Omega}^0}(\widehat{\omega}^0|\mathcal{D}')} \\ &= \sum_{m=0}^{M} c(\widehat{\omega}^{2m+1}; \mathscr{M}^m, \widehat{\omega}^{0:2m}, \mathcal{D}, \mathcal{D}') \end{aligned}$$

where $\widehat{\omega}^{0:t} = \{\widehat{\omega}^\tau\}_{\tau=0}^t$ and $\widehat{\Omega}^t$ is random variable whose realization is $\widehat{\omega}^t$. Since $\widehat{\omega}^0$ is randomly generated, which is independent of dataset, we have $P_{\widehat{\Omega}^0}(\widehat{\omega}^0|\mathcal{D}) = P_{\widehat{\Omega}^0}(\widehat{\omega}^0|\mathcal{D}')$. Moreover, the following holds for any λ:

$$\begin{aligned} & \log \mathbb{E}_{\widehat{\omega}^{0:2M}}\{\exp(\lambda c(\widehat{\omega}^{0:2M}; \{\mathscr{M}^m\}_{m=1}^M, \mathcal{D}, \mathcal{D}'))\} \\ = \; & \log \mathbb{E}_{\widehat{\omega}^{0:2M}}\{\exp(\lambda \sum_{m=0}^{M} c(\widehat{\omega}^{2m+1}; \mathscr{M}^m, \widehat{\omega}^{0:2m}, \mathcal{D}, \mathcal{D}')\} \\ = \; & \sum_{m=0}^{M} \log \mathbb{E}_{\widehat{\omega}^{2m+1}}\{\exp(\lambda c(\widehat{\omega}^{2m+1}; \mathscr{M}^m, \widehat{\omega}^{0:2m}, \mathcal{D}, \mathcal{D}')\} \end{aligned} \tag{13}$$

Therefore, $\alpha_{\{\mathscr{M}^m\}_{m=1}^M}(\lambda) \leq \sum_{m=1}^M \alpha_{\mathscr{M}^m}(\lambda)$ also holds. First bound each individual $\alpha_{\mathscr{M}^m}(\lambda)$.

Consider two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. Private (clipped) model $\widehat{\omega}^{2m+1}$ is generated by mechanism $\mathscr{M}^m(D) = \xi(\omega^{2m+1}) + N = \frac{1}{|D|}\sum_{d\in D}\eta(d) + N$ with function $||\eta(\cdot)||_2 \leq \tau$ and Gaussian noise $N \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. Without loss of generality, let $\mathcal{D}' = \mathcal{D} \cup \{d_n\}$, $f(d_n) = \pm\tau\mathbf{e}_1$ and $\sum_{d\in\mathcal{D}}\eta(d) = \mathbf{0}$. Then $\mathscr{M}^m(\mathcal{D})$ and $\mathscr{M}^m(\mathcal{D}')$ are distributed identically except for the first coordinate and the problem can be reduced to one-dimensional problem.

$$\begin{aligned} c(\widehat{\omega}^{2m+1}; \mathscr{M}^m, \widehat{\omega}^{0:2m}, \mathcal{D}, \mathcal{D}') &= \log \frac{P_{\widehat{\Omega}^{2m+1}}(\widehat{\omega}^{2m+1}|\mathcal{D}, \widehat{\omega}^{0:2m})}{P_{\widehat{\Omega}^{2m+1}}(\widehat{\omega}^{2m+1}|\mathcal{D}', \widehat{\omega}^{0:2m})} \\ &= \log \frac{P_N(n)}{P_N(n \pm \tau)} \\ &\leq \frac{\tau}{2|D|\sigma^2}(2|n| + \tau)\,. \end{aligned}$$

where $n + \frac{1}{|D|}\sum_{d\in\mathcal{D}}\eta(d) = \widehat{\omega}^{2m+1}$. Therefore,

$$\begin{aligned} \alpha_{\mathscr{M}^m}(\lambda) &= \log \mathbb{E}_{N\sim\mathcal{N}(0,\sigma^2)}\{\exp(\lambda \frac{\tau}{2|D|\sigma^2}(2N + \tau))\} \\ &= \log \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(n - \lambda\frac{\tau}{|D|})^2) \cdot \exp(\frac{\tau^2}{2|D|^2\sigma^2}(\lambda^2 + \lambda))dn \\ &= \frac{\tau^2\lambda(\lambda+1)}{2|D|^2\sigma^2}\,. \\ \alpha_{\{\mathscr{M}^m\}_{m=1}^M}(\lambda) &\leq \sum_{m=1}^{M} \alpha_{\mathscr{M}^m}(\lambda) = \frac{M\tau^2\lambda(\lambda+1)}{2|D|^2\sigma^2} \end{aligned}$$

Use the tail bound [Theorem 2, Abadi et al. (2016)], for any $\varepsilon \geq \frac{M\tau^2}{2|D|^2\sigma^2}$, the algorithm is (ε, δ)-differentially private for

$$\delta = \min_{\lambda:\lambda\geq 0} h(\lambda) = \min_{\lambda:\lambda\geq 0} \exp\Big(\frac{M\tau^2\lambda(\lambda+1)}{2|D|^2\sigma^2} - \lambda\varepsilon\Big)$$

To find $\lambda^* = \underset{\lambda:\lambda\geq 0}{\text{argmin}}\, h(\lambda)$, take derivative of $h(\lambda)$ and assign 0 gives the solution $\bar{\lambda} = \frac{\varepsilon|D|^2\sigma^2}{M\tau^2} - \frac{1}{2} \geq 0$, and $h''(\bar{\lambda}) > 0$, implies $\lambda^* = \bar{\lambda}$. Plug into equation 14 gives:

$$\delta = \exp\left(\left(\frac{M\tau^2}{4|D|^2\sigma^2} - \frac{\varepsilon}{2}\right)\left(\frac{\varepsilon|D|^2\sigma^2}{M\tau^2} - \frac{1}{2}\right)\right) \tag{14}$$

Similarly, for any $\delta \in [0,1]$, the algorithm is (ε, δ)-differentially private for

$$\varepsilon \;=\; \min_{\lambda:\lambda\geq 0} h_1(\lambda) = \min_{\lambda:\lambda\geq 0} \frac{M\tau^2(\lambda+1)}{2|D|^2\sigma^2} + \frac{1}{\lambda}\log\left(\frac{1}{\delta}\right) = 2\sqrt{\frac{M\tau^2}{2|D|^2\sigma^2}\log(\frac{1}{\delta})} + \frac{M\tau^2}{2|D|^2\sigma^2}$$

Proof of Theorem 5.3

Proof. WLOG, consider the case when local device got updated in every iteration and the algorithm runs over $2M$ iteration in total.

We will use the uppercase letters X and lowercase letters x to denote random variables and the corresponding realizations, and use $P_X(\cdot)$ to denote its probability distribution. To simplify the notations, we will drop the index i as we are only concerned with one agent, and use ω^t to denote private output $\widehat{\omega}^t$.

For two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ of agent i, by Lemma 5.1, the total privacy loss is only contributed by odd iterations. Thus, the ratio of joint probabilities (privacy loss) is given by:

$$\frac{P_{\Omega^{0:2M}}(\omega^{0:2M}|\mathcal{D})}{P_{\Omega^{0:2M}}(\omega^{0:2M}|\mathcal{D}')} = \frac{P_{\Omega^0}(\omega^0|\mathcal{D})}{P_{\Omega^0}(\omega^0|\mathcal{D}')} \cdot \prod_{m=0}^{M} \frac{P_{\Omega^{2m+1}}(\omega^{2m+1}|\omega^{0:2m},\mathcal{D})}{P_{\Omega^{2m+1}}(\omega^{2m+1}|\omega^{0:2m},\mathcal{D}')} \tag{15}$$

where $\omega^{0:t} := \{\omega^s\}_{s=1}^t$ and Ω^t denotes random variable of ω^t. Since ω^0 is randomly generated, which is independent of dataset. We have $P_{\Omega^0}(\omega^0|\mathcal{D}) = P_{\Omega^0}(\omega^0|\mathcal{D}')$.

Consider the $(2m+1)$-th iteration, by first-order condition, we have:

$$n^m = -\nabla F_i(\omega^{2m+1};\mathcal{D}) - \mu(\omega^{2m+1} - \overline{\omega}^{2m}) := g(\omega^{2m+1};\mathcal{D})$$

Given $\omega^{0:2m}$, n^m and ω^{2m+1} will be bijective and the relation is captured by a one-to-one mapping $g:\mathbb{R}^d \to \mathbb{R}^d$ defined above. By Jacobian transformation, we have

$$P_{\Omega^{2m+1}}(\omega^{2m+1}|\omega^{0:2m},\mathcal{D}) = P_{N^m}(g(\omega^{2m+1};\mathcal{D}))\cdot|\det(\mathbf{J}(g(\omega^{2m+1};\mathcal{D})))|$$

Therefore,

$$\frac{P_{\Omega^{2m+1}}(\omega^{2m+1}|\omega^{0:2m},\mathcal{D})}{P_{\Omega^{2m+1}}(\omega^{2m+1}|\omega^{0:2m},\mathcal{D}')} = \frac{P_{N^m}(g(\omega^{2m+1};\mathcal{D}))}{P_{N^m}(g(\omega^{2m+1};\mathcal{D}'))}\cdot\frac{|\det(\mathbf{J}(g(\omega^{2m+1};\mathcal{D})))|}{|\det(\mathbf{J}(g(\omega^{2m+1};\mathcal{D}')))|}$$

Let $n^m := g(\omega^{2m+1};\mathcal{D})$, $n^{m'} := g(\omega^{2m+1};\mathcal{D}')$ be noise vectors that result in output ω^{2m+1} under neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ respectively. WLOG, let $d_1 \in \mathcal{D}$ and $d_1' \in \mathcal{D}'$ be the data pints in two datasets that are different, and $\mathcal{D}\setminus d_1 = \mathcal{D}'\setminus d_1'$. Because noise vector $N^m \sim \exp(-\alpha^m||n^m||)$, we have,

$$\begin{aligned}\frac{P_{N^m}(g(\omega^{2m+1};\mathcal{D}))}{P_{N^m}(g(\omega^{2m+1};\mathcal{D}'))} &\leq \exp(\alpha^m||n^m - n^{m'}||) = \exp(\alpha^m||\nabla F_i(\omega^{2m+1};\mathcal{D}') - \nabla F_i(\omega^{2m+1};\mathcal{D})||) \\ &= \exp\left(\frac{\alpha^m}{|\mathcal{D}|}||\nabla F_i(\omega^{2m+1};d_1') - \nabla F_i(\omega^{2m+1};d_1)||\right) \leq \exp\left(\frac{2\alpha^m u_1}{|\mathcal{D}|}\right)\end{aligned} \tag{16}$$

Jacobian matrix

$$\mathbf{J}(g(\omega^{2m+1};\mathcal{D}))) = -\nabla^2 F_i(\omega^{2m+1};\mathcal{D}) - \mu\mathbf{I}_d := A \tag{17}$$

Further define matrix

$$A_\Delta = \mathbf{J}(g(\omega^{2m+1}; \mathcal{D}'))) - A = \frac{1}{|\mathcal{D}|}\Big(\nabla^2 F_i(\omega^{2m+1}; d_1) - \nabla^2 F_i(\omega^{2m+1}; d_1')\Big)$$

Then

$$\frac{|\det(\mathbf{J}(g(\omega^{2m+1}; \mathcal{D})))|}{|\det(\mathbf{J}(g(\omega^{2m+1}; \mathcal{D}')))|} = \frac{|\det(A)|}{|\det(A_\Delta + A)|} = \frac{1}{|\det(I + A^{-1}A_\Delta)|} = \frac{1}{|\prod_{k=1}^{r}(1 + \lambda_k(A^{-1}A_\Delta))|}$$

where $\lambda_k(A^{-1}A_\Delta)$ denotes the k-th largest eigenvalue of matrix $A^{-1}A_\Delta$. Under generalized linear models, A_Δ has rank at most 2. Because $-\frac{u_2}{|\mathcal{D}|\mu} \leq \lambda_k(A^{-1}A_\Delta) \leq \frac{u_2}{|\mathcal{D}|\mu}$ and $\mu, u_2, |\mathcal{D}|$ satisfy $\frac{u_2}{|\mathcal{D}|\mu} \leq 0.5$, we have,

$$\frac{|\det(\mathbf{J}(g(\omega^{2m+1}; \mathcal{D})))|}{|\det(\mathbf{J}(g(\omega^{2m+1}; \mathcal{D}')))|} \leq \frac{1}{|1 - \frac{u_2}{|\mathcal{D}|\mu}|^2} = \exp(-2\ln(1 - \frac{u_2}{|\mathcal{D}|\mu})) \leq \exp\Big(\frac{2.8u_2}{|\mathcal{D}|\mu}\Big) \tag{18}$$

where the last inequality holds because $-\ln(1-x) < 1.4x, \forall x \in [0, 0.5]$.

Combine Eqn. equation 15, equation 18 and equation 16, we have

$$\begin{aligned}\frac{P_{\Omega^{0:2M}}(\omega^{0:2M}|\mathcal{D})}{P_{\Omega^{0:2M}}(\omega^{0:2M}|\mathcal{D}')} &\leq \prod_{m=0}^{M} \exp\Big(\frac{2\alpha^m u_1}{|\mathcal{D}|}\Big) \cdot \exp\Big(\frac{2.8u_2}{|\mathcal{D}|\mu}\Big) \\ &= \exp\Big(\sum_{m=0}^{M} \frac{2\alpha^m u_1 \mu + 2.8u_2}{|\mathcal{D}|\mu}\Big)\end{aligned}$$

Theorem 5.2 is proved. □

F EXPERIMENTS

F.1 DETAILS OF DATASETS

`Synthetic`. The synthetic data is generated using the same method in Li et al. (2020). We briefly describe the generating steps here. For each device k, y_k is computed from a softmax function $y_k = \text{argmax}(\text{softmax}(W_k x_k + b_k))$. W_k and b_k are drawn from the same Gaussian distribution with mean u_k and variance 1, where $u_k \in N(0; \beta)$. $x_k \in N(v_k; \Sigma)$. v_k is drawn from a Gaussian distribution with mean $B_k \in \mathcal{N}(0, \gamma)$ and variance 1. Σ is diagonal with $\sum j, j = j^{-1.2}$. In such a setting, β controls how many local models differ from each other and γ controls how much local data at each device differs from that of other devices.

In our experiment, we take $k = 30$, $x \in \mathcal{R}^{20}$, $W \in \mathcal{R}^{10*20}$, $b \in \mathcal{R}^{10}$. We generate 4 datasets in total. They're `Synthetic(iid)` `Synthetic(0,0)` with $\beta = 0$ and $\gamma = 0$, `Synthetic(0.5,0.5)` with $\beta = 0.5$ and $\gamma = 0.5$ and `Synthetic(1,1)` with $\beta = 1$ and $\gamma = 1$.

`Femnist`: Similar with Li et al. (2020), we subsample 10 lower case characters ('a'-'j') from EMNIST Cohen et al. (2017) and distribute 5 classes to each device. There are 50 devices in total. The input is 28*28 image.

`Sent140`: A text sentiment analysis task on tweets Go et al. (2009). The input is sequence of length 25 and output is the probabilities of 2 classes.

A brief summary of dataset can be found here.

F.2 DETAILS OF THE MODEL

Classifier and Loss function. In our experiment, we use logistic regression for `Synthetic`, Multilayer perceptron and CNN for `Femnist`, Stacked LSTM for `Sent140`. Note that without privacy concern, any classifier and loss function can be plugged in to `Upcycled-FL`. However, if we adopt objective perturbation as privacy protection, the loss function should also satisfy assumptions

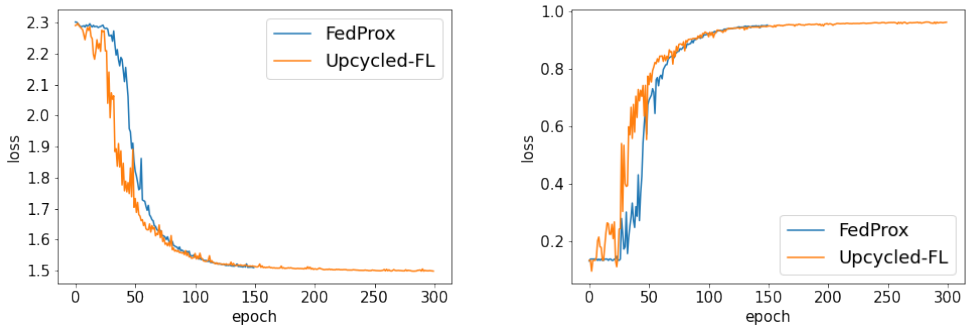
Table 2: Details of datasets. Numbers in parentheses represent the amount of test data. All of the numbers round to integer.

Dataset		Samples	# of device	Samples per device	
				mean	stdev
Synthetic	iid	6726(683)	30	224	166
	0,0	13791(1395)	30	460	841
	0.5,0.5	8036(818)	30	268	410
	1,1	10493(1063)	30	350	586
Femnist		16421(1924)	50	328	273
Sent140		32299(8484)	52	621	105

in Theorem 5.3. Thus log loss $\mathscr{L}(z) = log(1 + \exp(-z))$ is used. It's not hard to verify that this loss function satisfies assumptions in Theorem 5.3 by taking first and second order derivatives($|\mathscr{L}'| \leq u_1 = 1$ and $\mathscr{L}'' \leq u_2 = \frac{1}{4}$). For Multilayer perceptron, we use a two layer network with hidden dimension 14*14. For CNN, a 3*3 convolutional layer with 3 channels is adopted, following by two linear layers(followed by activation and dropout) with dimension 1000 and 100. For Stacked LTSM, we use 2 layers with 128 dimension `Sent140`. The word embedding dimension of `Sent140` is 300.

Implementation and Environment. Our code is implemented in Pytorch 1.11 Paszke et al. (2019). We employ SGD as local optimizer, with momentum 0.5, and set number of epochs E to 20 at each iteration m. Learning rate is tuned to 0.05. We run the model on a single GTX 1080 Ti.

F.3 CNN ON FEMNIST



(a) (b)

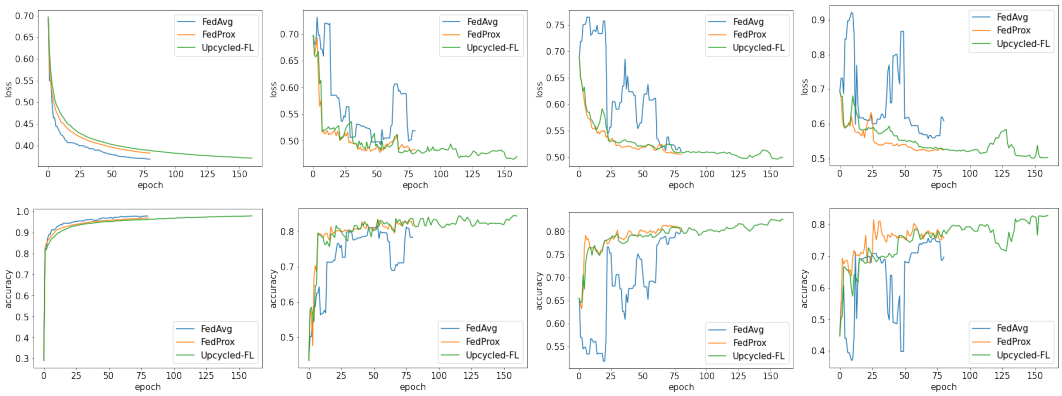
Figure 3: Loss and accuracy on `Femnist` using CNN. Trained with fraction=0.3, straggler=90%. $\mu = 0.1, \lambda = 0.04$. Training(testing) accuracy is `Upcycled-FL` **96.36%(96.21%)**, `FedProx` 95.63%(95.74%).

Table 3: Training(testing) accuracy of FedAvg, FedProx, Upcycled-FL with 90% straggler: FedAvg/FedProx/Upcycled-FL are trained over synthetic data for 80/80/160 iterations, and are trained over Femnist for 150/150/300 iterations. For Upcycled-FL, $\lambda = 0.04, 0.12, 0.21, 0.43$ is set based on the value of $\mu = 0.1, 0.3, 0.5, 1$, respectively. Upcycled-FL significantly outperforms FedAvg and FedProx.

Dataset	Proximal term	Accuracy(%)		
		FedAvg	FedProx	Upcycled-FL
Synthetic(iid)	$\mu = 0.1$	97.80%(97.51%)	96.86%(96.34%)	**97.88%(97.51%)**
	$\mu = 0.3$		95.18%(94.88%)	**96.31%(96.05%)**
	$\mu = 0.5$		94.03%(93.70%)	**95.40%(95.02%)**
	$\mu = 1$		92.14%(91.80%)	**93.76%(93.70%)**
Synthetic(0,0)	$\mu = 0.1$	78.41%(76.27%)	81.66%(79.50%)	**84.41%(82.44%)**
	$\mu = 0.3$		81.73%(79.35%)	**83.83%(81.58%)**
	$\mu = 0.5$		81.55%(79.35%)	**82.84%(81.29%)**
	$\mu = 1$		81.22%(78.85%)	**82.41%(80.57%)**
Synthetic(0.5,0.5)	$\mu = 0.1$	79.53%(78.97%)	79.63%(79.58%)	**81.19%(81.54%)**
	$\mu = 0.3$		80.31%(80.56%)	**82.14%(83.74%)**
	$\mu = 0.5$		79.55%(79.46%)	**82.61%(82.64%)**
	$\mu = 1$		80.11%(80.20%)	**82.74%(82.64%)**
Synthetic(1,1)	$\mu = 0.1$	69.70%(71.12%)	**79.20%(78.83%)**	77.38%(78.64%)
	$\mu = 0.3$		74.77%(75.91%)	**80.43%(81.47%)**
	$\mu = 0.5$		75.36%(76.67%)	**82.78%(83.07%)**
	$\mu = 1$		75.69%(76.58%)	**82.88%(82.69%)**
Femnist	$\mu = 0.1$	20.43%(20.01%)	82.86%(83.73%)	**86.37%(85.91%)**
	$\mu = 0.3$		76.10%(76.66%)	**81.94%(82.12%)**
	$\mu = 0.5$		68.28%(68.87%)	**77.77%(77.60%)**
	$\mu = 1$		28.82%(29.05%)	**54.81%(54.00%)**
Sent140	$\mu = 0.1$	73.41%(71.13%)	75.29%(72.57%)	**78.12%(72.38%)**
	$\mu = 0.3$		72.86%(71.38%)	**74.27%(73.02%)**
	$\mu = 0.5$		70.18%(69.07%)	72.92%(71.44%)
	$\mu = 1$		62.10%(62.13%)	63.91%(64.11%)

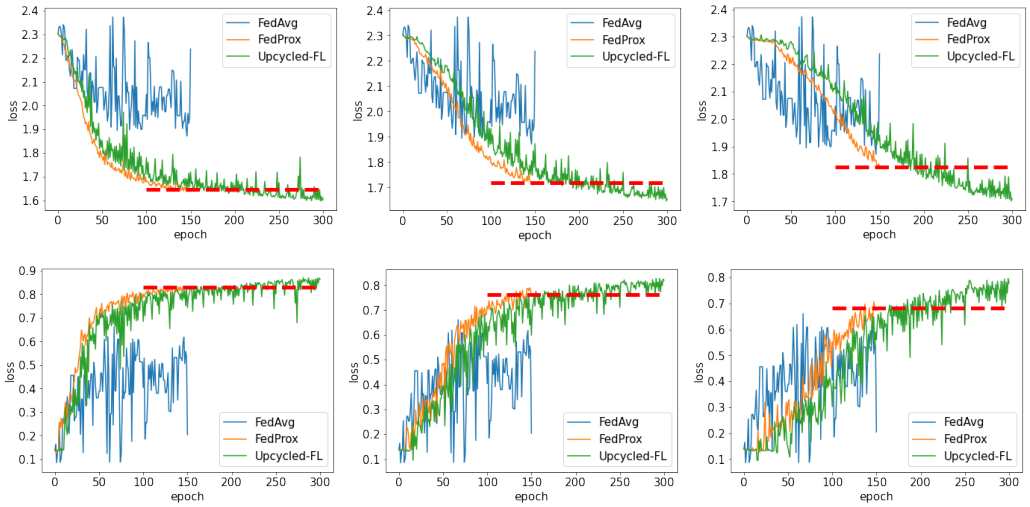
F.4 CONVERGENCE ON ALL DATASETS

F.4.1 90% STRAGGLER



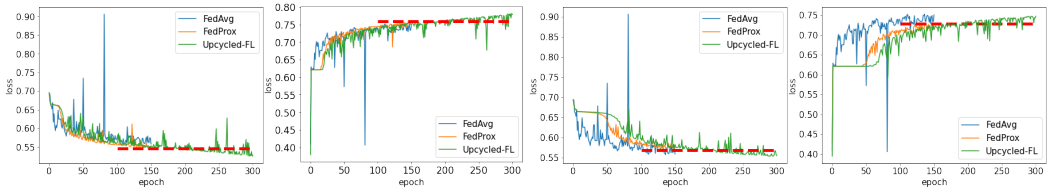
(a) (b) (c) (d)

Figure 4: Loss and accuracy on synthetic datasets. Trained with fraction=0.3, straggler=90%. (a) `Synthetic(iid)`; (b) `Synthetic(0,0)`; (c) `Synthetic(0.5,0.5)`; (d) `Synthetic(1,1)`. $\mu = 0.1, \lambda = 0.04$ for `Synthetic(iid)`, `Synthetic(0,0)`, $\mu = 1, \lambda = 0.42$ for `Synthetic(0.5,0.5)`, `Synthetic(1,1)`.



(a) $\mu = 0.1$ (b) $\mu = 0.3$ (c) $\mu = 0.5$

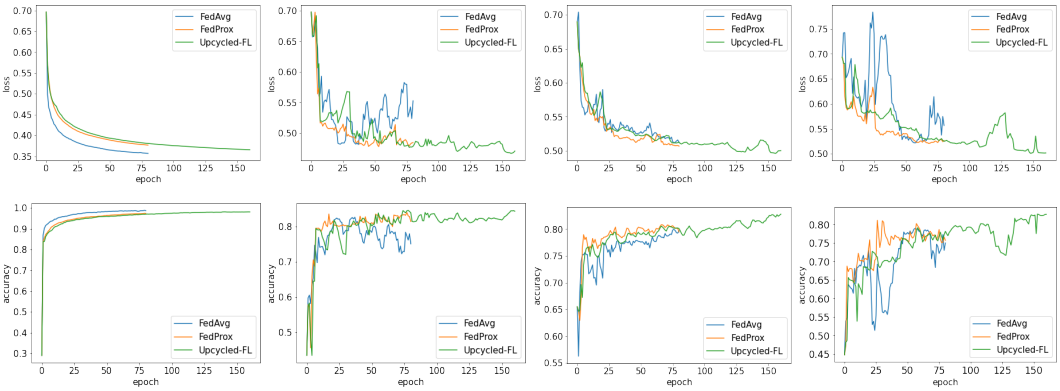
Figure 5: Loss and accuracy on `Femnist`. Trained with fraction=0.3, straggler=90%. $\lambda = 0.04, 0.12, 0.21$ is set based on the value of $\mu = 0.1, 0.3, 0.5$, respectively.



(a) $\mu = 0.1$ (b) $\mu = 0.3$

Figure 6: Loss and accuracy on `Sent140`. Trained with fraction=0.3, straggler=90%. $\mu = 0.1, \lambda = 0.04$ for (a), $\mu = 0.3, \lambda = 0.12$ for (b).

F.4.2 30% STRAGGLER



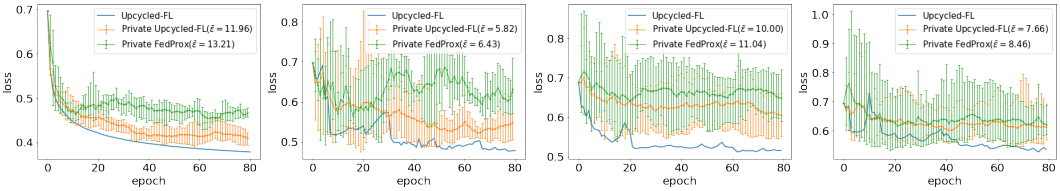
(a) (b) (c) (d)

Figure 7: Loss and accuracy on synthetic datasets. Trained with fraction=0.3, straggler=30%. (a) `Synthetic(iid)`; (b) `Synthetic(0,0)`; (c) `Synthetic(0.5,0.5)`; (d) `Synthetic(1,1)`. $\mu = 0.1, \lambda = 0.04$ for `Synthetic(iid)`, `Synthetic(0,0)`, $\mu = 1, \lambda = 0.42$ for `Synthetic(0.5,0.5)`, `Synthetic(1,1)`.

F.5 PRIVACY ON ALL DATASETS

F.5.1 OBJECTIVE PERTURBATION

- **Convergence**



(a) (b) (c) (d)

Figure 8: Objective perturbation on synthetic datasets. Trained with fraction=1, straggler=0%. $\mu = 0.1, \lambda = 0.04$. (a) `Synthetic(iid)`; (b) `Synthetic(0,0)`; (c) `Synthetic(0.5,0.5)`; (d) `Synthetic(1,1)`.

- **Impact of α**

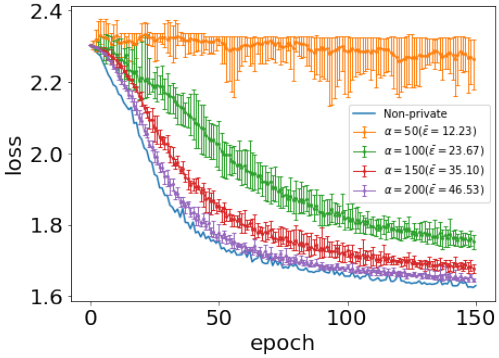
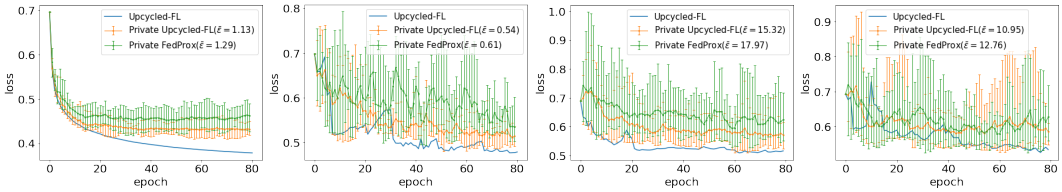


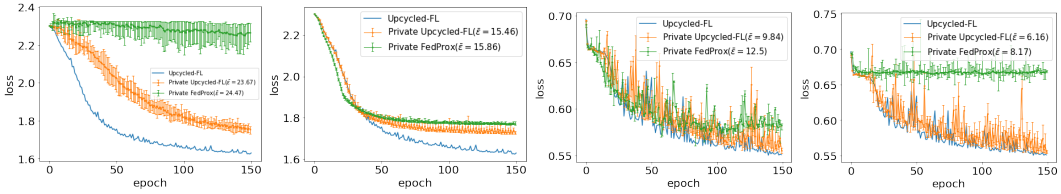
Figure 9: Impact of α on `Femnist`. Trained with fraction=1, straggler=0%. $\mu = 0.1,\ \lambda = 0.04$.

F.5.2 OUTPUT PERTURBATION



(a) (b) (c) (d)

Figure 10: Output perturbation on synthetic datasets. Trained with fraction=1, straggler=0%. $\mu = 0.1, \lambda = 0.04$. (a) `Synthetic(iid)`; (b) `Synthetic(0,0)`; (c) `Synthetic(0.5,0.5)`; (d) `Synthetic(1,1)`.



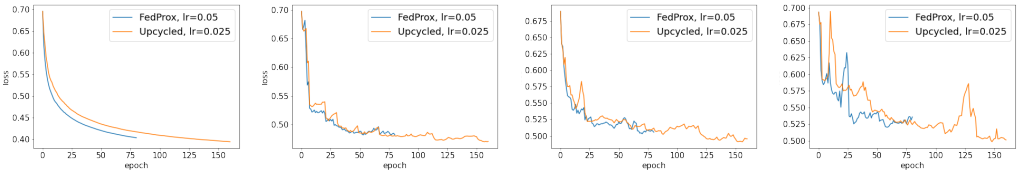
(a) (b) (c) (d)

Figure 11: Objective and output perturbation on `Femnist` and `Sent140`. (a) shows the result using objective perturbation on `Femnist`. The average accuracy of `Private Upcycled-FL` is 71.04%(70.71%), of `Private FedProx` is 19.72%(18.82%). (b) shows the result using output perturbation on `Femnist`. The average accuracy of `Private Upcycled-FL` is 79.43%(79.63%), of `Private FedProx` is 78.43%(78.36%). (c) shows the result using objective perturbation on `Sent140`. The average accuracy of `Private Upcycled-FL` is 74.36%(72.38%), of `Private FedProx` is 69.94%(68.02%). (d) shows the result using output perturbation on `Sent140`. The average accuracy of `Private Upcycled-FL` is 74.42%(72.52%), of `Private FedProx` is 62.10%(62.13%).All trained with fraction=1, straggler=0%, $\mu = 0.1, \lambda = 0.04$. For objective perturbation, $\delta = 0.001$.

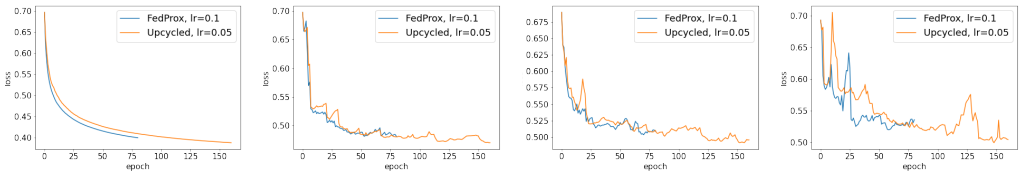
Table 4: Comparison of `Upcycled-FL` and `FedProx` using different learning rates. Trained with fraction=0.3, straggler=90%. For synthetic datasets, $\mu = 0.3, \lambda = 0.12$. For `Femnist`, $\mu = 0.1, \lambda = 0.04$.

Setting	`Synthetic(iid)`	`Synthetic(0,0)`	`Synthetic(0.5,0.5)`	`Synthetic(1,1)`	`Femnist`
`Upcycled-FL`, lr 0.025	**95.86%(95.90%)**	**83.88%(81.51%)**	**82.04%(83.99%)**	**81.02%(81.47)**	**85.56%(85.45%)**
`FedProx`, lr 0.05	95.18%(94.88%)	81.73%(79.35%)	80.31%(80.56%)	74.77%(75.91%)	82.86%(83.73%)
`Upcycled-FL`, lr 0.05	**96.31%(96.05%)**	**83.82%(81.57%)**	**82.14%(83.74%)**	**80.43%(81.47)**	**86.37%(85.91%)**
`FedProx`, lr 0.1	95.43%(95.17%)	81.92%(79.50%)	74.71%(75.82%)	74.77%(75.91%)	78.45%(78.79%)

F.6 DOUBLE LEARNING RATE OF FEDPROX

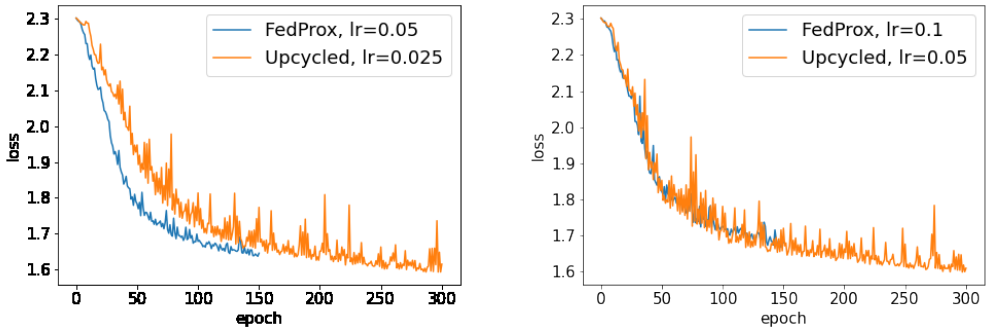


(a) learning rate 0.025 for `Upcycled-FL` and 0.05 for `FedProx`.



(b) learning rate 0.05 for `Upcycled-FL` and 0.1 for `FedProx`.

Figure 12: Comparison of `Upcycled-FL` and `FedProx` using different learning rates on synthetic datasets. Trained with fraction=0.3, straggler=90%. $\mu = 0.1, \lambda = 0.04$. (a) `Synthetic(iid)`; (b) `Synthetic(0,0)`; (c) `Synthetic(0.5,0.5)`; (d) `Synthetic(1,1)`. See Table 4 for accuracy reported.



(a) (b)

Figure 13: Comparison of `Upcycled-FL` and `FedProx` using different learning rates on `Femnist`. Trained with fraction=0.3, straggler=90%. $\mu = 0.1, \lambda = 0.04$. See Table 4 for accuracy reported.