CERTIFIED ROBUSTNESS FOR FREE IN DIFFERENTIALLY PRIVATE FEDERATED LEARNING

Chulin Xie
UIUC
chulinx2@illinois.edu

Yunhui Long
UIUC
ylong4@illinois.edu

Pin-Yu Chen
IBM Research
pin-yu.chen@ibm.com

Krishnaram Kenthapadi
Amazon AWS AI/ML
kenthk@amazon.com

Bo Li
UIUC
lbo@illinois.edu

Abstract

Federated Learning (FL) provides an efficient training paradigm to jointly train a global model leveraging data from distributed users. As the local training data comes from different users who may not be trustworthy, several studies have shown that FL is vulnerable to poisoning attacks where adversaries add malicious data during training. On the other hand, to protect the privacy of users, FL is usually trained in a differentially private manner (DPFL). Given these properties of FL, in this paper, we aim to ask: Can we leverage the innate privacy property of DPFL to provide robustness certification against poisoning attacks? Can we further improve the privacy of FL to improve such certification? To this end, we first investigate both user-level and instance-level privacy for FL, and propose novel randomization mechanisms and analysis to achieve improved differential privacy. We then provide two robustness certification criteria: certified prediction and certified attack cost for DPFL on both levels. Theoretically, given different privacy properties of DPFL, we prove their certified robustness under a bounded number of adversarial users or instances. Empirically, we conduct extensive experiments to verify our theories under different attacks on a range of datasets. We show that the global model with a tighter privacy guarantee always provides stronger robustness certification in terms of the certified attack cost, while it may exhibit tradeoffs regarding the certified prediction. We believe our work will inspire future research of developing certifiably robust DPFL based on its inherent properties.

1 Introduction

Federated Learning (FL), which aims to jointly train a global model with distributed local data, has been widely applied in different applications, such as finance [1], medical analysis [2], and user behavior prediction [3,4,5]. However, the fact that the local data and the training process are entirely controlled by the local users who may be adversarial raises great concerns from both security and privacy perspectives. In particular, recent studies show that FL is vulnerable to different types of training-time attacks, such as model poisoning [6], backdoor attacks [7,8,9], and label-flipping attacks [10]. Further, privacy concerns have motivated the need to keep the raw data on local devices without sharing. However, sharing other indirect information such as gradients or model updates as part of the FL training process can also leak sensitive user information [11,12,13,14]. As a result, approaches based on differential privacy (DP) [15], homomorphic encryption [16,17,18], and secure multiparty computation [19,20] have been proposed to protect privacy of users in federated learning. In particular, differentially private federated learning (DPFL) provides strong information theoretic guarantees on user privacy, while causing relatively low performance overhead [21].

Several defenses have been proposed to defend against poisoning attacks in FL. For instance, various robust aggregation methods [10,22,23,24,25,26,27,28] identify and down-weight the malicious updates during aggregation or estimate a true “center” of the received updates rather than taking a weighted average. Other methods include robust federated training protocols (e.g., clipping [29], noisy perturbation [29], and additional evaluation during training [30]) and...
Recent studies suggest that differential privacy (DP) is inherently related with robustness of ML models. Intuitively, DP is designed to protect the privacy of individual data, such that the output of an algorithm remains essentially unchanged when one individual input point is modified. Hence, the prediction of a DP model will be less impacted by a small amount of poisoned training data. Consequently, DP has been used to provide both theoretical and empirical defenses against evasion attacks [34] and data poisoning attacks [35, 36] on centralized ML models. It has also been used as an empirical defense against backdoor attacks [37] in federated learning [7, 29], although no theoretical guarantee is provided. To the best of our knowledge, despite of the wide application of DPFL, there is no work providing certified robustness for DPFL leveraging its privacy property.

In this paper, we aim to leverage the inherent privacy property of DPFL to provide robustness certification for FL against poisoning attacks for free. Our challenges include: (1) performing privacy analysis over training rounds in DPFL algorithms and (2) theoretically guaranteeing certified robustness based on DP properties under a given privacy budget. We propose two robustness certification criteria for FL: \textit{certified prediction} and \textit{certified attack cost} under different attack constraints. We consider both user-level DP [38, 39, 40, 41] which is widely guaranteed in FL, and instance-level DP [43, 44] which is less explored in FL. We prove that a FL model satisfying user-level DP is certifiably robust against a bounded number of adversarial users. In addition, we propose \texttt{InsDP-FedAvg} algorithm to improve instance-level DP in FL, and prove that instance-level DPFL is certifiably robust against a bounded number of adversarial instances. We also study the correlation between privacy guarantee and certified robustness of FL. While stronger privacy guarantees result in greater attack cost, overly strong privacy can hurt the certified prediction by introducing too much noise in the training process. Thus, the optimal certified prediction is often achieved under a proper balance between privacy protection and utility loss.

**Key Contributions.** Our work takes the first step to provide certified robustness in DPFL for free against poisoning attacks. We make contributions on both theoretical and empirical fronts.

- We propose two criteria for certified robustness of FL against poisoning attacks.
- Given a FL model satisfying user-level DP, we prove that it is certifiably robust against arbitrary poisoning attacks with a bounded number of adversarial users.
- We propose \texttt{InsDP-FedAvg} algorithm to improve FL instance-level privacy guarantee. We prove that instance-level DPFL is certifiably robust against the manipulation of a bounded number of instances during training.
- We conduct extensive experiments on MNIST and CIFAR datasets to verify our theoretical results.

## 2 Related work

**Differentially Private Federated Learning.** Different approaches are proposed to guarantee the user-level privacy for FL. [39, 40] clip the norm of each local update, add Gaussian noise on the summed update, and characterize its privacy budget via moment accountant [45]. [40] extends [39] to language models. In CpSGD [38], each user clips and quantizes the model update, and adds noise drawn from Binomial distribution, achieving both communication efficiency and DP. [13] derive DP for FL via Renyi divergence [46] and study its protection against data reconstruction attacks. [42] utilizes Laplacian smoothing for each local update to enhance the model utility. Instead of using moment accountant to track privacy budget over FL rounds as previous work, [41] derives the DP parameters by interpreting each round as a Markov kernel and quantify its impact on privacy parameters. All these works only focus on providing \textit{user-level} privacy, leaving its robustness property unexplored.

In terms of instance-level privacy for FL, there are only a few work [43, 44]. Dopamine [43] provides instance-level privacy guarantee when each user only performs one step of DP-SGD [45] at each FL round. However, it cannot be applied to multi-step SGD for each user, thus it cannot be extended to the general FL setting FedAvg [47]. [44] privately aggregate the labels from users in a voting scheme, and provide DP guarantees on both user level and instance level. However, it is also not applicable to standard FL, since it does not allow aggregating the gradients or updates.

**Differential Privacy and Robustness.** In standard (centralized) learning, Pixel-DP [34] is proposed to certify the model robustness against \textit{evasion} attacks. However, it is unclear how to leverage it to certify against poisoning attacks. To certify the robustness against \textit{poisoning} attacks, [35] show that private learners are resistant to data poisoning and...
analyze the lower bound of attack cost against poisoning attacks for regression models. Here we certify the robustness in DPFL setting with such lower bound as one of our certification criteria and additionally derive its upper bounds. \[36\] show that the off-the-shelf mechanism DP-SGD \[45\], which clips per-sample gradients and add Guassian noises during training, can serve as a defense against poisoning attacks empirically. In \textit{federated learning}, empirical work \[7, 29\] show that DPFL can mitigate backdoor attacks; however, none of these work provides certified robustness guarantees for DPFL against poisoning attacks.

### 3 Preliminaries

We start by providing some background on differential privacy (DP) and federated learning (FL).

**Differential Privacy (DP).** DP is a formal, mathematically rigorous definition (and standard) of privacy that intuitively guarantees that a randomized algorithm behaves similarly on similar inputs and that the output of the algorithm is about the same whether or not an individual’s data is included as part of the input \[15\].

**Definition 1** ((\(\epsilon, \delta\))-DP \[48\]). A randomized mechanism \(\mathcal{M} : \mathcal{D} \rightarrow \Theta\) with domain \(\mathcal{D}\) and range \(\Theta\) satisfies \((\epsilon, \delta\))-DP if for any pair of two adjacent datasets \(d, d' \in \mathcal{D}\), and for any possible (measurable) output set \(E \subseteq \Theta\), it holds that 
\[
\Pr[\mathcal{M}(d) \in E] \leq e^\epsilon \Pr[\mathcal{M}(d') \in E] + \delta.
\]

In Definition 1, when \(\mathcal{M}\) is a training algorithm for ML model, domain \(\mathcal{D}\) and range \(\Theta\) represent all possible training datasets and all possible trained models respectively. Group DP for \((\epsilon, \delta\))-DP mechanisms follows immediately from Definition 1 where the privacy guarantee drops with the size of the group. Formally, it says:

**Lemma 1** (Group DP). For mechanism \(\mathcal{M}\) that satisfies \((\epsilon, \delta\))-DP, it satisfies \((k\epsilon, \frac{1-k\epsilon}{1-\epsilon}\delta\))-DP for groups of size \(k\). That is, for any \(d, d' \in \mathcal{D}\) that differ by \(k\) individuals, and any \(E \subseteq \Theta\) it holds that 
\[
\Pr[\mathcal{M}(d) \in E] \leq e^{k\epsilon} \Pr[\mathcal{M}(d') \in E] + \frac{1-e^{k\epsilon}}{1-\epsilon}\delta.
\]

**Federated Learning.** FedAvg was introduced by \[47\] for FL to train a shared global model without direct access to training data of users. Specifically, given a FL system with \(N\) users, at round \(t\), the server sends the current global model \(w_{t-1}\) to users in the selected user set \(U_t\), where \(|U_t| = m = qN\) and \(q\) is the user sampling probability. Each selected user \(i \in U_t\) locally updates the model for \(t\) local epochs with its dataset \(D_i\) and learning rate \(\eta\) to obtain a new local model. Then, the server sends the local model updates \(\Delta w_i^t\) to the server. Finally, the server aggregates over the updates from all selected users into the new global model \(w_t = w_{t-1} + \frac{1}{m} \sum_{i \in U_t} \Delta w_i^t\).

### 4 User-level Privacy and Certified Robustness for FL

#### 4.1 User-level Privacy

Definition 1 leaves the definition of adjacent datasets flexible, which depends on applications. To protect user-level privacy, adjacent datasets are defined as those differing by data from one user \[40\].

**Definition 2** (User-level \((\epsilon, \delta)\)-DP). Let \(B, B'\) be two user sets with size \(N\). Let \(D\) and \(D'\) be the datasets that are the union of local training examples from all users in \(B\) and \(B'\) respectively. Then, \(D\) and \(D'\) are adjacent if \(B\) and \(B'\) differ by one user. The mechanism \(\mathcal{M}\) satisfies user-level \((\epsilon, \delta)\)-DP if it meets Definition 1 with \(D\) and \(D'\) as adjacent datasets.

Following standard DPFL \[39, 40\], we introduce UserDP-FedAvg (Algorithm 1 in Appendix A.1) to achieve user-level DP. At each round, the server first clips the update from each user with a threshold \(S\) such that its \(\ell_2\)-sensitivity is upper bounded by \(S\). Next, the server sums up the updates, adds Gaussian noise sampled from \(N(0, \sigma^2 S^2)\), and takes the average, i.e., \(w_t \leftarrow w_{t-1} + \frac{1}{m} \left( \sum_{i \in U_t} \text{Clip}(\Delta w_i^t, S) + N(0, \sigma^2 S^2) \right)\). We utilize the moment accountant \[45\] to compute the DP guarantee. Compared to the standard composition theorem \[15\], the moment accountant method provides tighter bounds for the repeated application of the Gaussian mechanism combined with amplification via-sampling. Given the user sampling probability \(q\), noise level \(\sigma\), FL rounds \(T\), and a \(\delta > 0\), UserDP-FedAvg satisfies \((\epsilon, \delta)\)-DP as below, which is a generalization of \[45\]. The proof of proposition 1 is deferred to Appendix A.1. The main challenge is to analyze privacy budget \(\epsilon\), which is accumulated as \(T\) increases due to the continuous access to training data.

**Proposition 1** (UserDP-FedAvg Privacy Guarantee). There exist constants \(c_1\) and \(c_2\) so that given user sampling probability \(q\), and FL rounds \(T\), for any \(\epsilon < c_1 q^2 T\), if \(\sigma \geq c_2 \frac{q \sqrt{T \log(1/\delta)}}{\epsilon}\), the randomized mechanism \(\mathcal{M}\) in Algorithm 1 is \((\epsilon, \delta)\)-DP for any \(\delta > 0\).
4.2 Certified Robustness of User-level DPFL against Poisoning Attacks

**Threat Model.** We consider the poisoning attacks against FL, where \( k \) adversarial users have poisoned instances in local datasets, aiming to fool the trained DPFL global model. Such attacks include backdoor attacks \([57, 49]\) and label flipping attacks \([50, 51]\). The detailed description of these attacks is deferred to Appendix A.2. Note that our robustness certification is attack-agnostic under certain attack constraints (e.g., \( k \)), and we will verify our certification bounds with different poisoning attacks in Section 6. Next, we propose two criteria for the robustness certification in FL: certified prediction and certified attack cost.

**Certified Prediction.** Consider the classification task with \( C \) classes. We define the classification scoring function \( f : (\Theta, \mathbb{R}^d) \to Y^C \) which maps model parameters \( \theta \in \Theta \) and an input data \( x \in \mathbb{R}^d \) to a confidence vector \( f(\theta, x) \), and \( f_c(\theta, x) \in [0, 1] \) represents the confidence of class \( c \). We mainly focus on the confidence after normalization, i.e., \( f(\theta, x) \in Y^C = \{ p \in \mathbb{R}^C_{\geq 0} : \|p\|_1 = 1 \} \) in the probability simplex. Since the DP mechanism \( M \) is randomized and produces a stochastic FL global model \( \theta = M(D) \), it is natural to resort to a probabilistic expression as a bridge for quantitative robustness certifications. Following the convention in \([52, 35]\), we use the expectation of the model’s prediction to provide a quantitative guarantee on the robustness of \( M \). Specifically, we define the expected scoring function \( \bar{F} : (\theta, \mathbb{R}^d) \to Y^C \) where \( f_c(M(D), x) = \mathbb{E}[f_c(M(D), x)] \) is the expected confidence for class \( c \). The expectation is taken over DP training randomness, e.g., random Gaussian noise and random user subsampling. The corresponding prediction \( H : (\theta, \mathbb{R}^d) \to [C] \) is defined by \( H(M(D), x) := \arg \max_{c \in [C]} f_c(M(D), x) \), which is the top-1 class based on the expected prediction confidence. We will prove that such prediction allows robustness certification against poisoning attacks.

Following our threat model above and DPFL training in Algorithm 1, we denote the trained global model exposed to poisoning attacks by \( M(D') \). When \( k = 1 \), \( D \) and \( D' \) are user-level adjacent datasets according to Definition 2. Given that mechanism \( M \) satisfies user-level \((\epsilon, \delta)\)-DP, based on the innate DP property, the distribution of the stochastic model \( M(D') \) is “close” to the distribution of \( M(D) \). Moreover, according to the post-processing property of DP, during testing, given a test sample \( x \), we would expect the values of the expected confidence for each class \( c \), i.e., \( f_c(M(D'), x) \) and \( f_c(M(D), x) \), to be close, and hence the returned most likely class to be the same, i.e., \( H(M(D), x) = H(M(D'), x) \), indicating robust prediction against poisoning attacks.

**Theorem 1** (Condition for Certified Prediction under One Adversarial User). Suppose a randomized mechanism \( M \) satisfies user-level \((\epsilon, \delta)\)-DP. For two user sets \( B \) and \( B' \) that differ by one user, let \( D \) and \( D' \) be the corresponding training datasets. For a test input \( x \), suppose \( A, B \in [C] \) satisfy \( A = \arg \max_{c \in [C]} f_c(M(D), x) \) and \( B = \arg \max_{c \in [C]: c \neq A} f_c(M(D), x) \), then

\[
F_A(M(D), x) > e^{2\epsilon} F_B(M(D), x) + (1 + e^\epsilon)\delta,
\]

it is guaranteed that \( H(M(D'), x) = H(M(D), x) = A \).

When \( k > 1 \), we resort to group DP. According to Lemma 1 given mechanism \( M \) satisfying user-level \((\epsilon, \delta)\)-DP, it also satisfies user-level \((k\epsilon, \frac{1 - e^\epsilon}{1 - e^{2\epsilon}}\delta)\)-DP for groups of size \( k \). When \( k \) is smaller than a certain threshold, leveraging the group DP property, we would expect that the distribution of the stochastic model \( M(D') \) is not too far away from the distribution of \( M(D) \) such that they would make the same prediction for a test sample with probabilistic guarantees. Therefore, the privacy and robustness guarantees are simultaneously met by \( M \).

**Theorem 2** (Upper Bound of \( k \) for Certified Prediction). Suppose a randomized mechanism \( M \) satisfies user-level \((\epsilon, \delta)\)-DP. For two user sets \( B \) and \( B' \) that differ by \( k \) users, let \( D \) and \( D' \) be the corresponding training datasets. For a test input \( x \), suppose \( A, B \in [C] \) satisfy \( A = \arg \max_{c \in [C]} f_c(M(D), x) \) and \( B = \arg \max_{c \in [C]: c \neq A} f_c(M(D), x) \), then \( H(M(D'), x) = H(M(D), x) = A \), \( \forall k < K \) where \( K \) is the certified number of adversarial users:

\[
K = \frac{1}{2\epsilon} \log \frac{F_A(M(D), x)(e^\epsilon - 1) + \delta}{F_B(M(D), x)(e^\epsilon - 1) + \delta}.
\]

The proofs of Theorems 1 and 2 are omitted to Appendix A.4. Theorems 1 and 2 reflect a tradeoff between privacy and certified prediction: (i) In Theorem 1 if \( \epsilon \) is large such that the RHS of Eq (1) > 1, the robustness condition cannot be met since the expected confidence \( F_A(M(D), x) \in [0, 1] \). However, to achieve small \( \epsilon \), i.e., strong privacy, large noise is required during training, which would hurt model utility and thus result in small confidence margin between the top two classes (e.g., \( F_A(M(D), x) \) and \( F_B(M(D), x) \)), making it hard to meet the robustness condition. (ii) In Theorem 2, if we fix \( F_A(M(D), x) \) and \( F_B(M(D), x) \), smaller \( \epsilon \) of FL can certify larger \( K \). However, smaller \( \epsilon \) also induces smaller confidence margin, thus reducing \( K \) instead. As a result, properly choosing \( \epsilon \) would help to certify a large \( K \).
Certified Attack Cost. In addition to the certified prediction, we define the attack cost for attacker $C : \Theta \rightarrow \mathbb{R}$ which quantifies the difference between the poisoned model and the attack goal. In general, attacker aims to minimize the expected attack cost $J(D) := \mathbb{E}[C(M(D))]$, where the expectation is taken over the randomness of DP training. The cost function can be instantiated according to the concrete attack goal in different types of poisoning attacks, and we provide some examples below. Given a global FL model satisfying user-level $(\epsilon, \delta)$-DP, we will prove the lower bound of the attack cost $J(D')$ when manipulating the data of at most $k$ users. Higher lower bound of the attack cost indicates more certifiably robust global model.

Example 1. (Backdoor attack) $C(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(\theta, z_i^*),$ where $z_i^* = (x_i + \delta, y_i).$ $\delta$ is the backdoor pattern, $y_i$ is the target adversarial label. Minimizing $J(D')$ drives the prediction on any test data with the backdoor pattern $\delta$ to the target label $y^*.$

Example 2. (Label Flipping attack) $C(\theta) = \frac{1}{n} \sum_{i=1}^{n} l(\theta, z_i^*),$ where $z_i^* = (x_i, y_i^*)$ and $y_i^*$ is the target adversarial label. Minimizing $J(D')$ drives the prediction on test data $x_i$ to the target label $y_i^*.$

Example 3. (Parameter-Targeting attack) $C(\theta) = \frac{1}{n} \| \theta - \theta^* \|^2,$ where $\theta^*$ is the target model. Minimizing $J(D')$ drives the poisoned model to be close to the target model.

Theorem 3 (Attack Cost with $k$ Attackers). Suppose a randomized mechanism $\mathcal{M}$ satisfies user-level $(\epsilon, \delta)$-DP. For two user sets $B$ and $B'$ that differ $k$ users, $D$ and $D'$ are the corresponding training datasets. Let $J(D)$ be the expected attack cost where $|\mathcal{C}(\cdot)| \leq \tilde{C}.$ Then,

$$
\min \{ e^{kC} J(D) + \frac{e^{kC} - 1}{e^\epsilon - 1} \delta \tilde{C}, \tilde{C} \} \geq J(D') \geq \max \{ e^{-kC} J(D) - \frac{1 - e^{-kC}}{e^\epsilon - 1} \delta \tilde{C}, 0 \}, \text{ if } C(\cdot) \geq 0
$$

$$
\min \{ e^{-kC} J(D) + \frac{1 - e^{-kC}}{e^\epsilon - 1} \delta \tilde{C}, 0 \} \geq J(D') \geq \max \{ e^{kC} J(D) - \frac{e^{kC} - 1}{e^\epsilon - 1} \delta \tilde{C}, -\tilde{C} \}, \text{ if } C(\cdot) \leq 0
$$

The proof is omitted to Appendix A.4. Theorem 3 provides the upper bounds and lower bounds for attack cost $J(D').$ The lower bounds show that to what extent the attack can reduce $J(D')$ by manipulating up to $k$ users, i.e., how successful the attack can be. The lower bounds depend on the attack cost on clean model $J(D),$ $k$ and $\epsilon.$ When $J(D)$ is higher, the DPFL model under poisoning attacks is more robust because the lower bounds are accordingly higher; a tighter privacy guarantee, i.e., smaller $\epsilon,$ can also lead to higher robustness certification as it increases the lower bounds; with larger $k,$ the attacker ability grows and thus lead to lower possible $J(D').$ The upper bounds show the least adversarial effect brought by $k$ attackers, i.e., how vulnerable the DPFL model is under the optimistic case (e.g., the backdoor pattern is less distinguishable).

Leveraging the lower bounds in Theorem 3, we can lower-bound the minimum number of attackers required to reduce the attack cost to certain level associated with hyperparameter $\tau$ in Corollary 4.

Corollary 1 (Lower Bound of $k$ Given $\tau$). Suppose a randomized mechanism $\mathcal{M}$ satisfies user-level $(\epsilon, \delta)$-DP. Let attack cost function be $C,$ the expected attack cost be $J(\cdot).$ In order to achieve $J(D') \leq \frac{\tau}{\gamma} J(D)$ for $\tau \geq 1$ when $0 \leq C(\cdot) \leq \tilde{C},$ or achieve $J(D') \leq \tau J(D)$ for $1 \leq \tau \leq -\frac{\tilde{C}}{J(D)}$ when $-\tilde{C} \leq C(\cdot) \leq 0,$ the number of adversarial users should satisfy:

$$
k \geq \frac{1}{\epsilon} \log \frac{e^{\epsilon} - 1}{e^{\epsilon} - 1} J(D) \tau + \tilde{C} \delta \tau \quad \text{or} \quad k \geq \frac{1}{\epsilon} \log \frac{e^{\epsilon} - 1}{e^{\epsilon} - 1} J(D) \tau - \tilde{C} \delta$$

The proof is omitted to Appendix A.4. Corollary 4 shows that stronger privacy guarantee (i.e., smaller $\epsilon$) requires more attackers to achieve the same effectiveness of attack, indicating higher robustness.

5 Instance-level Privacy and Certified Robustness for FL

5.1 Instance-level Privacy

In this section, we introduce the instance-level DP definition, the corresponding algorithm, and the privacy analysis for FL. When DP is used to protect the privacy of individual instance, the trained stochastic FL model should not differ much if one instance is modified. Hence, the adjacent datasets in instance-level DP are defined as those differing by one instance.

Definition 3 (Instance-level $(\epsilon, \delta)$-DP). Let $D$ be the dataset that is the union of local training examples from all users. Then, $D$ and $D'$ are adjacent if they differ by one instance. The mechanism $\mathcal{M}$ is instance-level $(\epsilon, \delta)$-DP if it meets Definition 4 with $D$ and $D'$ as adjacent datasets.
Dopamine [43] provides the first instance-level privacy guarantee under FedSGD [47]. However, it has two limitations. First, its privacy bound is loose. Although FedSGD performs both user and batch sampling during training, Dopamine ignores the privacy gain provided by random user sampling. In this section, we improve the privacy guarantee under FedSGD with privacy amplification via user sampling [52, 45]. This improvement leads to algorithm InsDP-FedSGD, to achieve tighter privacy analysis. We defer the algorithm (Algorithm 2) as well as its privacy guarantee to Appendix A.1.

Besides the loose privacy bound, Dopamine [43] only allows users to perform one step of DP-SGD [45] during each FL round. This restriction limits the efficiency of the algorithm and increases the communication overhead. In practice, users in FL are typically allowed to update their local models for many steps before submitting updates to reduce the communication cost. To solve this problem, we further improve InsDP-FedSGD to support multiple local steps during each round. Specifically, we propose a novel instance-level DPFL algorithm InsDP-FedAvg (Algorithm 3) in Appendix A.1 allowing users to train multiple local SGD steps before submitting the updates. In InsDP-FedAvg, each user i performs local DP-SGD so that the local training mechanism \( M^i \) satisfies instance-level DP. Then, the server aggregates the updates. We prove that the global mechanism \( M \) preserves instance-level DP using DP parallel composition theorem [54] and moment accountant [45].

Algorithm 3 formally presents the InsDP-FedAvg algorithm and the calculation of its privacy budget \( \epsilon \). Specifically, at first, local privacy cost \( \epsilon_0^i \) is initialized as 0 before FL training. At round \( t \), if user \( i \) is not selected, its local privacy cost is kept unchanged \( \epsilon_t^i \leftarrow \epsilon_{t-1}^i \). Otherwise user \( i \) updates local model by running DP-SGD for \( V \) local steps with batch sampling probability \( p \), noise level \( \sigma \) and clipping threshold \( S \), and \( \epsilon_t^i \) is accumulated upon \( \epsilon_{t-1}^i \) via local model accountant. Next, the server aggregates the updates from selected users, and leverages \( \{ \epsilon_t^i \}_{i \in [N]} \) and the parallel composition in Theorem 4 to calculate the global privacy cost \( \epsilon \). After \( T \) rounds, the mechanism \( M \) that outputs the FL global model in Algorithm 3 is instance-level \( (\epsilon, \delta) \)-DP.

**Theorem 4 (InsDP-FedAvg Privacy Guarantee).** In Algorithm 3 during round \( t \), if the local mechanism \( M^i \) satisfies \( (\epsilon_t^i, \delta) \)-DP, then the global mechanism \( M \) satisfies \( \left( \max_{i \in [N]} \epsilon_t^i, \delta \right) \)-DP.

The idea behind Theorem 4 is that when \( D' \) and \( D \) differ in one instance, the modified instance only falls into one local dataset, and thus parallel composition theorem [54] can be applied. Then the privacy guarantee corresponds to the worst-case, and is obtained by taking the maximum local privacy cost across all the users. The detailed proof is given in Appendix A.1.

### 5.2 Certified Robustness of Instance-level DPFL against Poisoning Attacks

**Threat Model.** We consider poisoning attacks under the presence of \( k \) poisoned instances. These instances could be controlled by the same or multiple adversarial users. Our robustness certification is agnostic to the attack methods as long as the number of poisoned instances is constrained.

According to the group DP property (Lemma 1) and the post-processing property for FL model with instance-level \( (\epsilon, \delta) \)-DP, we prove that our robust certification results proposed for user-level DP are also applicable to instance-level DP. Below is the formal theorem (proof is given in Appendix A.4).

**Theorem 5.** Suppose \( D \) and \( D' \) differ by \( k \) instances, and the randomized mechanism \( M \) satisfies instance-level \( (\epsilon, \delta) \)-DP. The results in Theorems 1, 2 and 3 and Corollary 1 hold for \( M, \) \( D, \) and \( D' \).

**Comparison with existing certified prediction methods in centralized setting.** The form of Theorem 5 is similar with the robustness condition against test-time attack in Proposition 1 of [54]. This is because the derived robustness conditions are both rooted in the DP properties, but ours focus on the robustness against training-time attacks in FL, which is more challenging considering the distributed nature and the model training dynamics, i.e., the analysis of the privacy budget over training rounds. Our Theorem 5 is also different from previous randomized smoothing-based certifiably robust centralized learning against backdoor [55] and label flipping [56]. First, our randomness comes from the inherent training randomness of user/instance-level \( (\epsilon, \delta) \)-DP, e.g., user subsampling and Gaussian noise. Thus, the certified robustness for free in FedFL means that the DPFL learning algorithm \( M \) itself is randomized, and such randomness can lead to the robustness certification with non-trivial quantitative measurement of the randomness. On the contrary, robustness in randomized smoothing-based methods comes from explicitly making the classification process randomized via adding noise in training datasets [55, 56], or test samples [54, 57] which is easier to measure. Second, our Theorem 1, 2 hold no matter how \( \epsilon \) is achieved, which means that we can add different types of noise, leverage different subsampling strategies or even different FL training protocols to achieve user/instance-level \( \epsilon \). However, in [55, 56] different certifications require different types of noise (Laplacian, Gaussian, etc.). Additionally, DP is suitable to characterize the robustness against poisoning since DP composition theorems can be leveraged to track privacy cost.
6 Experiments

We present evaluations for robustness certifications, especially Thm. 2 and Cor. 1. We find that 1) there is a tradeoff between certified prediction and privacy on certain datasets; 2) a tighter privacy guarantee always provides stronger certified robustness in terms of the certified attack cost; 3) our lower bounds of certified attack cost are generally tight when \( k \) is small. When \( k \) is large, they are tight under strong attacks (e.g., large local poisoning ratio \( \alpha \)). Stronger attacks or tighter certification are required to further tighten the gap between the empirical robustness and theoretical bounds.

Data and Model. We evaluate our robustness certification results with two datasets: MNIST and CIFAR-10. For each dataset, we use corresponding standard CNN architectures in the differential privacy library \( \text{DPLLib} \) of PyTorch. Following previous work on DP ML [59, 55] and backdoor attacks [60, 55] which evaluate with two classes, we focus on binary classification for MNIST (digit 0 and 1) and CIFAR-10 (airplane and bird), and defer the 10-class results to Appendix A.3. We train FL model following Algorithm 1 for user-level privacy and Algorithm 2 for instance-level privacy. We refer the readers to Appendix A.3 for details about the datasets, networks, parameter setups.

Poisoning Attacks. We evaluate several state-of-the-art poisoning attacks against the proposed UserDP-FedAvg and InsDP-FedAvg. We first consider backdoor attacks (BKD) and label flipping attacks (LF). For InsDP-FedAvg, we consider the worst case where \( k \) backdoored or label-flipped instances are fallen into the dataset of one user. For UserDP-FedAvg, we additionally evaluate distributed backdoor attack (DBA), which is claimed to be a more stealthy backdoor attack against FL. Moreover, we consider BKD, LF and DBA via model replacement approach [7] where \( k \) attackers train the local models using local datasets with a fraction of poisoned instances, and scale the malicious updates with hyperparameter \( \gamma \), i.e., \( \Delta w^i_\gamma \leftarrow \gamma \Delta w^i_\gamma \) before sending them to the sever. This way, the malicious updates would have a stronger impact on the FL model. Note that even when attackers perform scaling, after server clipping, the sensitivity of updates is still upper-bounded by the clipping threshold \( S \). So the privacy guarantee in Proposition 1 still holds under poisoning attacks via model replacement. Detailed attack setups are presented in Appendix A.3.

Evaluation Metrics and Setup. We consider two evaluation metrics based on our robustness certification criteria. The first metric is certified accuracy, which is the fraction of the test set for which the poisoned FL model makes correct and consistent predictions compared with the clean FL model. Given a test set of size \( n \), for \( i \)-th test sample, the ground truth label is \( y_i \), the output prediction is \( c_i \), and the certified number of adversarial users/instances is \( K_i \). We calculate the certified accuracy at \( k \) as
\[
\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{c_i = y_i \text{ and } K_i \geq k\}.
\]

The second metric is the lower bound of attack cost in Theorem 3:
\[
\frac{1}{n} \sum_{i=1}^{n} \max \left\{ e^{-\epsilon \text{CertAcc}}(D) -\frac{1-e^{-\epsilon}k}{e^\epsilon}, 0 \right\}.
\]

We evaluate the tightness of \( \frac{1}{n} \sum_{i=1}^{n} \max \left\{ e^{-\epsilon \text{CertAcc}}(D) -\frac{1-e^{-\epsilon}k}{e^\epsilon}, 0 \right\} \) by comparing it with empirical attack cost \( \frac{1}{n} \sum_{i=1}^{n} \max \left\{ e^{-\epsilon \text{CertAcc}}(D) -\frac{1-e^{-\epsilon}k}{e^\epsilon}, 0 \right\} \).

6.1 Robustness Evaluation of User-level DPFL

Certified Prediction. Figure 4(a) and 4(b) present the user-level certified accuracy under different \( \epsilon \) by training DPFL models with different noise scale \( \sigma \). We observe that the largest \( k \) can be certified when \( \epsilon \) is around 0.6298 in MNIST and 0.1451 in CIFAR-10, which verifies the tradeoff between \( \epsilon \) and certified accuracy as we discussed in Section 4.2.

Advanced DP protocols that require less noise while achieving similar level of privacy are favored to improve the privacy, utility, and certified accuracy simultaneously.

Certified Attack Cost. In order to evaluate Theorem 3 and characterize the tightness of our theoretical lower bound \( \frac{1}{n} \sum_{i=1}^{n} \max \left\{ e^{-\epsilon \text{CertAcc}}(D) -\frac{1-e^{-\epsilon}k}{e^\epsilon}, 0 \right\} \) under different local poison fraction \( \alpha \), attack methods and scale factor \( \gamma \) in Figure 2. Note that when \( k = 0 \), the model is benign so the empirical cost equals to the certified one. We find that 1) when \( k \) increases, the attack ability grows, and both the empirical attack cost and theoretical lower bound decreases. 2) In Figure 2 row 1, given the same \( k \), higher \( \alpha \), i.e., poisoning more local instances for each attacker, achieves a stronger attack, under which lower empirical \( \frac{1}{n} \sum_{i=1}^{n} \max \left\{ e^{-\epsilon \text{CertAcc}}(D) -\frac{1-e^{-\epsilon}k}{e^\epsilon}, 0 \right\} \) can be achieved and is more close to the certified lower
Figure 1: Certified accuracy of FL satisfying user-level DP (a,b), and instance-level DP (c,d).

bound. This indicates that the lower bound appears tighter when the poisoning attack is stronger. 3) In Figure 2 row 2, we fix $\alpha = 100\%$ and evaluate $\text{UserDP-FedAvg}$ under different $\gamma$ and attack methods. It turns out that DP serves as a strong defense empirically for FL, given that $J(D)$ did not vary much under different $\gamma (1, 50, 100)$ and different attack methods (BKD, DBA, LF). This is because the clipping operation restricts the magnitude of malicious updates, rendering the model replacement ineffective; the Gaussian noise perturbs the malicious updates and makes the DPFL model stable, and thus the FL model is less likely to memorize the poisoning instances. 4) In both rows, the lower bounds are tight when $k$ is small. When $k$ is large, there remains a gap between our theoretical lower bounds and empirical attack costs under different attacks, which will inspire more effective poisoning attacks or tighter robustness certification.

Figure 2: Certified attack cost of user-level DPFL given different $k$, under attacks with different $\alpha$ or $\gamma$.

**Certified Attack Cost under Different $\epsilon$.** Here we further explore the impacts of different factors on the certified attack cost. Figure 2 presents the empirical attack cost and the certified attack cost lower bound given different $\epsilon$ on user-level DP. It is shown that as the privacy guarantee becomes stronger, i.e., smaller $\epsilon$, the model is more robust achieving higher $J(D')$ and $J(D')$. In Figure 5(a)(b), we train user-level $(\epsilon, \delta)$ DPFL models, calculate corresponding $J(D)$, and plot the lower bound of $k$ given different attack effectiveness hyperparameter $\tau$ according to Corollary 1. It shows that 1) when the required attack effectiveness is higher, i.e., $\tau$ is larger, more number of attackers is required. 2) To achieve the same effectiveness of attack, fewer number of attackers is needed for larger $\epsilon$, which means that DPFL model with weaker privacy is more vulnerable to poisoning attacks.

Figure 3: Certified attack cost of user-level DPFL with different $\epsilon$ under different attacks.

**6.2 Robustness Evaluation of Instance-level DPFL**

**Certified Prediction.** Figure 1(c)(d) show the instance-level certified accuracy under different $\epsilon$. The optimal $\epsilon$ for $K$ is around 0.3593 for MNIST and 0.6546 for CIFAR-10, which is aligned with our observation of the tradeoff between certified accuracy and privacy on user-level DPFL (Section 6.1).
Certified Attack Cost. Figure 4 show the certified attack cost on CIFAR-10. From Figure 4 (a)(b), poisoning more instances (i.e., larger $k$) induces lower theoretical and empirical attack cost. From Figure 4 (c)(d), it is clear that instance-level DPFL with stronger privacy guarantee provides higher attack cost both empirically and theoretically, meaning that it is more robust against poisoning attacks. Results on MNIST are deferred to Appendix A.3. Figure 5 (c)(d) show the lower bound of $k$ under different instance-level $\epsilon$ given different $\tau$. Fewer poisoned instances are required to reduce the $\mathbb{J}(D')$ to the similar level for a less private DPFL model, indicating that the model is easier to be attacked.

Figure 4: Certified attack cost of instance-level DPFL under different attacks given different number of malicious instances $k$ (a)(b) and different $\epsilon$ (c)(d).

Figure 5: Lower bound of $k$ under user-level $\epsilon$ (a,b) and instance-level $\epsilon$ (c,d) given attack effectiveness $\tau$.

7 Conclusion

In this paper, we present the first work on deriving certified robustness in DPFL for free against poisoning attacks. We propose two robustness certification criteria, based on which we prove that a FL model satisfying user-level (instance-level) DP is certifiably robust against a bounded number of adversarial users (instances). Our theoretical analysis characterizes the inherent relation between certified robustness and differential privacy of FL on both user and instance levels, which are empirically verified with extensive experiments. Our results can be used to improve the trustworthiness of DPFL.

References


Appendix

The Appendix is organized as follows:

- Appendix A.1 provides the DPFL algorithms on both user and instance levels, and the proofs for corresponding privacy guarantees.
- Appendix A.2 specifies our threat models.
- Appendix A.3 provides more details on experimental setups for training and evaluation, the addition experimental results on confidence level, robustness evaluation of instance-level DPFL on MNIST, robustness evaluation on 10-class classification, and DP bound comparison between InsDP-FedSGD and Dopamine.
- Appendix A.4 provides the proofs for the certified robustness related analysis, including Lemma 1, Theorem 1, 2, 3, 5 and Corollary 1.

A.1 Differentially Private Federated Learning

A.1.1 UserDP-FedAvg

**Algorithm 1: UserDP-FedAvg.**

| Input: | Initial model $w_0$, user sampling probability $q$, privacy parameter $\delta$, clipping threshold $S$, local datasets $D_1, ..., D_N$, local epochs $E$, learning rate $\eta$. |
| Output: | FL model $w_T$ and privacy cost $\epsilon$. |

Server executes:

for each round $t = 1$ to $T$

- $m \leftarrow \max(q \cdot N, 1)$;
- $U_t \leftarrow$ (random subset of $m$ users);
- for each user $i \in U_t$ in parallel do
  - $\Delta w_i^t \leftarrow \text{UserUpdate}(i, w_{t-1})$;
  - $w_i \leftarrow w_{t-1} + \frac{1}{m} \left( \sum_{i \in U_t} \text{Clip}(\Delta w_i^t, S) + N(0, \sigma^2 S^2) \right)$;
- $M.\text{accum_priv_spending}(\sigma, q, \delta)$;

$\epsilon = M.\text{get_privacy_spent}();$
return $w_T, \epsilon$

Procedure **UserUpdate** $(i, w_{t-1})$

- $w \leftarrow w_{t-1}$;
- for local epoch $e = 1$ to $E$
  - for batch $b$ in local dataset $D_i$ do
    - $w \leftarrow w - \eta \nabla l(w; b)$
  - $\Delta w_i^t \leftarrow w - w_{t-1}$;
return $\Delta w_i^t$

Procedure **Clip** $(\Delta, S)$

return $\Delta / \max(1, |\Delta|^2)$

In Algorithm 1, $M.\text{accum_priv_spending}()$ and $M.\text{get_privacy_spent}()$ are the calls on the moments accountant $M$ refer to the API of [45].

The privacy guarantee for Algorithm 1 is a generalization of [45]. We recall Proposition 1.

**Proposition 1 (UserDP-FedAvg Privacy Guarantee).** There exist constants $c_1$ and $c_2$ so that given user sampling probability $q$, and FL rounds $T$, for any $\epsilon < c_1 q^2 T$, if $\sigma \geq c_2 \frac{q \sqrt{T \log(1/\delta)}}{\epsilon}$, the randomized mechanism $M$ in Algorithm 1 is $(\epsilon, \delta)$-DP for any $\delta > 0$.

**Proof.** The proof follows the proof of Theorem 1 in [45], while the notations have slightly different meanings under FL settings. In Proposition 1, we use $q$ to represent user-level sampling probability and $T$ to represent FL training rounds.

**Discussion.** [21] divide the user-level privacy into global privacy [39, 40] and local privacy [38]. In both local and global privacy, the norm of each update is clipped. The difference lies in that the noise is added on the aggregated model updates in global privacy because a trusted server is assumed, while the noise is added on each local update in local privacy because it assumes that the central server might be malicious. Algorithm 1 belongs to global privacy.

A.1.2 InsDP-FedSGD

Under FedSGD, when each local model performs one step of DP-SGD [45], the randomized mechanism $M$ that outputs the global model preserves the instance-level DP. We can regard the one-step update for the global model in Algorithm 2 as:

$$w_t \leftarrow w_{t-1} - \frac{1}{m} \sum_{i \in U_t} \eta \frac{\bar{g}}{L} \left( \sum_{x_j \in b^i_t} \tilde{g}(x_j) + N(0, \sigma^2 S^2) \right)$$ (5)
We defer the DP bound evaluation comparison between Algorithm 2: InsDP-FedSGD.

**Algorithm 2: InsDP-FedSGD.**

**Input:** Initial model $w_0$, user sampling probability $q$, parameter $\delta$, local clipping threshold $S$, local noise level $\sigma$, local datasets $D_1, \ldots, D_N$, learning rate $\eta$, batch sampling probability $p$.

**Output:** FL model $w_T$ and privacy cost $\epsilon$

**Server executes:**

for each round $t = 1$ to $T$ do

\[ m \leftarrow \max (q \cdot N, 1); \]

\[ U_t \leftarrow \text{(random subset of} m \text{ clients)}; \]

for each user $i \in U_t$ in parallel do

\[ \Delta w_i^t \leftarrow \text{UserUpdate}(i, w_{t-1}); \]

\[ w_t \leftarrow w_{t-1} + \frac{1}{m} \sum_{i \in U_t} \Delta w_i^t; \]

\[ \epsilon = M.get\_privacy\_spent(); \]

return $w_T, \epsilon$

**Procedure UserUpdate($i, w_{t-1}$)**

\[ w \leftarrow w_{t-1}; \]

\[ b_i \leftarrow \text{uniformly sample a batch from} D_i \text{ with probability} p = L/|D_i|; \]

for each $x_j \in b_i$ do

\[ g(x_j) \leftarrow \nabla l(w; x_j); \]

\[ \tilde{g} \leftarrow \frac{1}{t} \left( \sum_{i} g(x_j) + \mathcal{N}(0, \sigma^2 S^2) \right); \]

\[ w \leftarrow w - \eta \tilde{g}; \]

\[ \Delta w_i^t \leftarrow w - w_{t-1}; \]

return $\Delta w_i^t$

**Procedure Clip($\Lambda, S$)**

\[ \text{return} \Delta / \max \left(1, \frac{\| \Delta \|_2}{S} \right) \]

**Proposition 2 (InsDP-FedSGD Privacy Guarantee).** There exist constants $c_1$ and $c_2$ so that given batch sampling probability $p$, and user sampling probability $q$, the number of selected users each round $m$, and FL rounds $T$, for any $\epsilon < c_1(pq)^2 T$, if $\sigma \geq c_2 \frac{D \sqrt{\log(1/\delta)}}{\epsilon \sqrt{m}}$, the randomized mechanism $\mathcal{M}$ in Algorithm 2 is $(\epsilon, \delta)$-DP for any $\delta > 0$.

**Proof.** i) In instance-level DP, we consider the sampling probability of each instance under the combination of user-level sampling and batch-level sampling. Since the user-level sampling probability is $q$ and the batch-level sampling probability is $p$, each instance is sampled with probability $pq$. ii) Additionally, since the sensitivity of instance-wise gradient w.r.t one instance is $S$, after local gradient descent and server FL aggregation, the equivalent sensitivity of global model w.r.t one instance is $S' = \frac{qS}{\sqrt{m}}$ according to Eq (5). iii) Moreover, since the local noise is $n_i \sim \mathcal{N}(0, \sigma^2 S^2)$, then the “virtual” global noise is $n = \frac{1}{mT} \sum_{i \in U_t} n_i$ according to Eq (5), so $n \sim \mathcal{N}(0, \frac{\sigma^2 S^2}{mT})$. Let $\frac{\sigma^2 S^2}{mT} = \sigma^2 S'^2$ such that $n \sim \mathcal{N}(0, \sigma^2 S'^2)$. Because $S' = \frac{qS}{\sqrt{m}}$, the equivalent global noise level is $\sigma'^2 = \sigma^2 m$, i.e., $\sigma' = \sigma \sqrt{m}$.

In Proposition 2, we use $pq$ to represent instance-level sampling probability, $T$ to represent FL training rounds, $\sigma \sqrt{m}$ to represent the equivalent global noise level. The rest of the proof follows the proof of Theorem 1 in [45].

We defer the DP bound evaluation comparison between InsDP-FedSGD and Dopamine to Appendix A.3.7.

**A.1.3 InsDP-FedAvg**

**Lemma 2 (InsDP-FedAvg Privacy Guarantee when $T = 1$).** In Algorithm 2 when $T = 1$, suppose local mechanism $\mathcal{M}'$ satisfies $(\epsilon', \delta')$-DP, then global mechanism $\mathcal{M}$ satisfies $(\max_{i \in [N]} \epsilon', \delta')$-DP.

**Proof.** We can regard federated learning as partitioning a dataset $D$ into $N$ disjoint subsets $\{D_1, D_2, \ldots, D_N\}$. $N$ mechanisms $\{\mathcal{M}'_1, \ldots, \mathcal{M}'_N\}$ are operated on these $N$ parts separately and each $\mathcal{M}'_i$ satisfies its own $\epsilon'_i$-DP for $i \in [1, N]$. Note that if $i$-th user is not selected, $\epsilon'_i = 0$ because local dataset $D_i$ is not accessed and there is no privacy cost. Without loss of generality, we assume the modified data sample $x'$ (x → x' causes $D \rightarrow D'$) is in the local dataset of $k$-th client $D_k$. Let $D, D'$ be two neighboring datasets ($D_k, D'_k$ are also two neighboring datasets). $\mathcal{M}$ is randomized mechanism that outputs the global model, and $\mathcal{M}'$ is the randomized mechanism that outputs the local model update $\Delta w_i'$. Suppose $w_0$ is the initialized and deterministic global model, and $\{z_1, \ldots, z_N\}$ are randomized local updates. We have a sequence of computations $\{z_1 = \mathcal{M}'_1(D_1), z_2 = \mathcal{M}'_2(D_2; z_1), z_3 = \mathcal{M}'_3(D_3; z_1, z_2) \ldots\}$ and $z = \mathcal{M}(D) = w_0 + \sum_{i=1}^{N} z_i$. Note that if $i$-th user is not selected, $z_i = 0$. According to the parallel composition[62],
we have
\[
\Pr[\mathcal{M}(D) = z] = \Pr[\mathcal{M}_1(D_1; z_1)] \Pr[\mathcal{M}_2(D_2; z_2)] \ldots \Pr[\mathcal{M}_N(D_N; z_1, \ldots, z_{N-1}) = z_N]
\leq \exp(e^k) \Pr[\mathcal{M}^k(D'_i; z_1, \ldots, z_{i-1}) = z_i] = \exp(e^k) \Pr[\mathcal{M}(D') = z]
\]
So $\mathcal{M}$ satisfies $e^k$-DP when the modified data sample lies in the subset $D_k$. Consider the worst case of where the modified data sample could fall in, we know that $\mathcal{M}$ satisfies $(\max_{i \in [N]} e^i)$-DP.

We recall Theorem 4.

**Theorem 4 (InsDP-FedAvg Privacy Guarantee).** In Algorithm 3 during round $t$, if the local mechanism $\mathcal{M}_i$ satisfies $(e^i, \delta)$-DP, then the global mechanism $\mathcal{M}$ satisfies $(\max_{i \in [N]} e^i, \delta)$-DP.

**Proof.** Again, without loss of generality, we assume the modified data sample $x'$ ($x \rightarrow x'$ causes $D \rightarrow D'$) is in the local dataset of $k$-th user $D_k$. We first consider the case when all users are selected. At each round $t$, $N$ mechanisms are operated on $N$ disjoint sets and each $\mathcal{M}_i$ satisfies own $e^i$-DP where $e^i$ is the privacy cost for accessing the local dataset $D_i$ for one round (not accumulating over previous rounds). Let $D, D'$ be two neighboring datasets ($D_k, D'_k$ are also two neighboring datasets). Suppose $z_0 = \mathcal{M}_{t-1}(D)$ is the aggregated randomized global model at round $t-1$, and $\{z_1, \ldots, z_N\}$ are the randomized local updates at round $t$, we have a sequence of computations $\{z_1 = \mathcal{M}_1(D_1; z_0), z_2 = \mathcal{M}_2(D_2; z_0, z_1), \ldots, z_N = \mathcal{M}_N(D_N; z_0, z_1, \ldots, z_{N-1})\}$ and $z = \mathcal{M}_t(D) = z_0 + \sum_{i=1}^{N} z_i$. We first consider the sequential composition [15] to accumulate the privacy cost over FL rounds. According to parallel composition, we have
\[
\Pr[\mathcal{M}_t(D) = z] = \Pr[\mathcal{M}_{t-1}(D) = z_0] \prod_{i=1}^{N} \Pr[\mathcal{M}_i(D_i; z_0, z_1, \ldots, z_{i-1}) = z_i]
\]
\[
= \Pr[\mathcal{M}_{t-1}(D) = z_0] \Pr[\mathcal{M}_t^k(D_i; z_0, z_1, \ldots, z_{i-1}) = z_i] \prod_{i \neq k} \Pr[\mathcal{M}_i(D_i; z_0, z_1, \ldots, z_{i-1}) = z_i]
\leq \exp(e_{t-1}) \Pr[\mathcal{M}_{t-1}(D') = z_0] \exp(e^k) \Pr[\mathcal{M}_t^k(D_i; z_0, z_1, \ldots, z_{i-1}) = z_i] \prod_{i \neq k} \Pr[\mathcal{M}_i(D_i; z_0, z_1, \ldots, z_{i-1}) = z_i]
\]
\[
= \exp(e_{t-1} + e^k) \Pr[\mathcal{M}_t(D') = z]
\]
We evaluate our robustness certification results with two datasets: MNIST \cite{lecun1998gradient} and CIFAR-10 \cite{krizhevsky2009learning}. For each dataset, we use corresponding standard CNN architectures in the differential privacy library \cite{abadi2016deep} of PyTorch \cite{paszke2017automatic}.

Moreover, moment accountant \cite{zhang2018efficient} is known to reduce the privacy cost from $O(t)$ to $O(\sqrt{t})$. We can use the more advanced composition, i.e., moment accountant, instead of the sequential composition, to accumulate the privacy cost for local mechanism $\mathcal{M}^k$ over $t$ FL rounds. In addition, we consider user subsampling. As described in Algorithm \cite{abadi2016deep} if the user $i$ is not selected at round $t$, then its local privacy cost is kept unchanged at this round.

Take the worst case of where $x'$ could lie in, at round $t$, $\mathcal{M}$ satisfies $\epsilon_i$-DP, where $\epsilon_i = \max_{i \in [N]} \epsilon'_i$, local mechanism $\mathcal{M}'$ satisfies $\epsilon_i$-DP, and the local privacy cost $\epsilon'_i$ is accumulated via local moment accountant in $i$-th user over $t$ rounds.

\section{A.2 Threat Models}

We consider targeted poisoning attacks of two types. In backdoor attacks \cite{huang2016badnets, li2019backdoor}, the goal is to embed a backdoor pattern (i.e., a trigger) during training such that any test input with such pattern will be mis-classified as the target. In label flipping attacks \cite{tian2020understanding, xie2020understanding}, the labels of clean training examples from one source class are flipped to the target class while the features of the data are kept unchanged. In FL, the purpose of backdoor attacks is to manipulate local models with backdoored local data, so that the global model would behave normally on untampered data samples while achieving high attack success rate on clean data \cite{tian2020understanding}. Given the same purpose, \textit{distributed backdoor} attack (DBA) \cite{lin2020towards} decomposes the same backdoor pattern to several smaller ones and embeds them to different local training sets for different adversarial users. The goal of label flipping attack against FL is to manipulate local datasets with flipped labels such that the global model will mis-classify the test data in the source class as the target class. The model replacement \cite{wong2020adversarial} is a more powerful approach to perform the above attacks, where the attackers first train the local models using the poisoned datasets and then scale the malicious updates before sending them to the server. This way, the attacker’s updates would have a stronger impact on the FL model. We use the model replacement method to perform poisoning attacks and study the effectiveness of DPFL.

For UserDP-FedAvg, we consider backdoor, distributed backdoor, and label flipping attacks via the model replacement approach. Next, we formalize the attack process and introduce the notations. Suppose the attacker controls $k$ adversarial users, i.e., there are $k$ attackers out of $N$ users. Let $B$ be the original user set of $N$ benign users, and $B'$ be the user set that contains $k$ attackers. Let $D := \{D_1, D_2, \ldots, D_N\}$ be the union of original benign local datasets across all users. For a data sample $z^i_j := \{x^i_j, y^i_j\}$ in $D_i$, we denote its backdoored version as $z'^{i}_{\text{backdoor}} := \{x^i_j + \delta_{x}^i, y^*\}$, where $\delta_{x}^i$ is the backdoor pattern, $y^*$ is the targeted label; the distributed backdoor attack (DBA) version as $z'^{i}_{\text{DBA}} := \{x^i_j + \delta_{x}^i, y^*\}$, where $\delta_{x}^i$ is the distributed backdoor pattern for attacker $i$; the label-flipped version as $z'^{i}_{\text{flip}} := \{x^i_j, y^*\}$. Note that the composition of all DBA patterns is equivalent to the backdoor pattern, i.e., $\sum_{i=1}^{k} \delta_{x}^i = \delta_{x}$. We assume attacker $i$ has $\xi_i$ fraction of poisoned samples in its local dataset $D'_i$. Let $D' := \{D'_1, \ldots, D'_{k-1}, D'_k, D_{k+1}, \ldots, D_N\}$ be the union of local datasets when $k$ attackers are present. The adversarial user $i$ performs model replacement by scaling the model update with hyperparameter $\gamma$ before submitting it to the server, i.e., $\Delta w_i^t \leftarrow \gamma \Delta w_i^t$.

For InSaDP-FedAvg, we consider both backdoor and label flipping attacks. Since distributed backdoor and model replacement attacks are proposed for adversarial users rather than adversarial instances, we do not consider them for instance-level DPFL. There are $k$ backdoored or label-flipped instances $\{z'_1, z'_2, \ldots, z'_k\}$, which could be controlled by same or multiple users.

\section{A.3 Experimental Details and Additional Results}

\subsection{A.3.1 Datasets and Models}

We evaluate our robustness certification results with two datasets: MNIST \cite{lecun1998gradient} and CIFAR-10 \cite{krizhevsky2009learning}. For each dataset, we use corresponding standard CNN architectures in the differential privacy library \cite{abadi2016deep} of PyTorch \cite{paszke2017automatic}.

**MNIST**: We study an image classification problem of handwritten digits in MNIST. It is a dataset of 70000 28x28 pixel images of digits in 10 classes, split into a train set of 60000 images and a test set of 10000 images. Except Section \ref{subsec:eval}, we consider binary classification on classes 0 and 1, making our train set contain 12665 samples, and the test set 21TS samples. The model consists of two Conv-ReLu-MaxPooling layers and two linear layers.
Table 1: Dataset description and parameters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Dataset</th>
<th>#training samples</th>
<th>N</th>
<th>m</th>
<th>E</th>
<th>V</th>
<th>batch size</th>
<th>η</th>
<th>S</th>
<th>δ</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserDP-FedAvg</td>
<td>MNIST</td>
<td>12665</td>
<td>200</td>
<td>20</td>
<td>10</td>
<td>/</td>
<td>60</td>
<td>0.02</td>
<td>0.7</td>
<td>0.0029</td>
<td>0.5</td>
</tr>
<tr>
<td>UserDP-FedAvg</td>
<td>CIFAR-10</td>
<td>10000</td>
<td>200</td>
<td>40</td>
<td>5</td>
<td>/</td>
<td>50</td>
<td>0.05</td>
<td>1</td>
<td>0.0029</td>
<td>0.2</td>
</tr>
<tr>
<td>InsDP-FedAvg</td>
<td>MNIST</td>
<td>12665</td>
<td>10</td>
<td>10</td>
<td>/</td>
<td>25</td>
<td>60</td>
<td>0.02</td>
<td>0.7</td>
<td>0.00001</td>
<td>0.5</td>
</tr>
<tr>
<td>InsDP-FedAvg</td>
<td>CIFAR-10</td>
<td>10000</td>
<td>10</td>
<td>10</td>
<td>/</td>
<td>100</td>
<td>50</td>
<td>0.05</td>
<td>1</td>
<td>0.00001</td>
<td>2</td>
</tr>
</tbody>
</table>

CIFAR-10: We study image classification of vehicles and animals in CIFAR-10. This is a harder dataset than MNIST, consisting of 60000 32x32x3 images, split into a train set of 50000 and a test set of 10000. Except Section A.3.6 we consider binary classification on class airplane and bird, making our train set contain 10000 samples, and the test set 2000 samples. The model consists of four Conv-ReLu-AveragePooling layers and one linear layer. When training on CIFAR-10, we follow the standard practice for differential privacy [45,59] and fine-tune a whole model pre-trained non-privately on the more complex CIFAR100, a similarly sized but more complex benchmark dataset.

A.3.2 Training Details

We simulate the federated learning setup by splitting the training datasets for N FL users in an i.i.d manner. FL users run SGD with learning rate η, momentum 0.9, weight decay 0.0005 to update the local models. The training parameter setups are summarized in Table 1. Following [40] that use δ ≈ 1/N^2 as privacy parameter, for UserDP-FedAvg we set δ = 0.0029 according to the total number of users, and for InsDP-FedAvg we set δ = 0.00001 according to the total number of training samples. Next we summarize the privacy guarantees and clean accuracy offered when we study the certified prediction and certified attack cost, which are also the training parameters setups when k = 0 in Figure 1(a)(b).

User-level DPFL In order to study the user-level certified prediction under different privacy guarantee, for MNIST, we set η to be 0.2808, 0.4187, 0.6298, 0.8694, 1.8504, 4.8913, 6.9269, which are obtained by training UserDP-FedAvg FL model for 3 rounds with noise level σ = 3.0, 2.3, 1.8, 1.5, 1.0, 0.8, 0.6, 0.5, respectively (Figure 1(a)). CIFAR-10, we set η to be 0.1083, 0.1179, 0.1451, 0.2444, 0.3663, 0.4527, 0.5460, 0.8781, which are obtained by training UserDP-FedAvg FL model for one round with noise level σ = 10.0, 8.0, 6.0, 4.0, 3.0, 2.6, 2.3, 1.7, respectively (Figure 1(b)).

To certify the attack cost under different number of adversarial users k (Figure 2), for MNIST, we set the noise level σ to be 2.5. When k = 0, after training UserDP-FedAvg for T = 3, 4, 5 rounds, we obtain FL models with privacy guarantee ε = 0.3672, 0.4025, 0.4344 and clean accuracy (average over M runs) 86.69%, 88.76%, 88.99%. For CIFAR-10, we set the noise level σ to be 3.0. After training UserDP-FedAvg for T = 3, 4 rounds under k = 0, we obtain FL models with privacy guarantee ε = 0.5346, 0.5978 and clean accuracy 78.63%, 78.46%.

With the interest of certifying attack cost under different user-level DP guarantee (Figure 3, Figure 5), we explore the empirical attack cost and the certified attack cost lower bound given different ε. For MNIST, we set the privacy guarantee ε to be 1.2716, 0.8794, 0.6608, 0.5249, 0.4344, which are obtained by training UserDP-FedAvg FL models for 5 rounds under noise level σ = 1.3, 1.6, 1.9, 2.2, 2.5, respectively, and the clean accuracy for the corresponding models are 99.50%, 99.06%, 96.52%, 93.39%, 88.99%. For CIFAR-10, we set the privacy guarantee ε to be 1.600, 1.2172, 1.0395, 0.8530, 0.6616, 0.6543, 0.5978, which are obtained by training UserDP-FedAvg FL models for 4 rounds under noise level σ = 1.5, 1.8, 2.0, 2.3, 2.5, 2.8, 3.0, respectively, and the clean accuracy for the corresponding models are 85.59%, 84.52%, 83.23%, 81.90%, 81.27%, 79.23%, 78.46%.

Instance-level DPFL To certify the prediction of instance-level DPFL under different privacy guarantee, for MNIST, we set privacy cost ε to be 0.2029, 0.2251, 0.2484, 0.3593, 0.4589, 0.6373, 1.0587, 3.5691, which are obtained by training InsDP-FedAvg FL models for 3 rounds with noise level σ = 15, 10, 8, 5, 4, 3, 2, 1, respectively (Figure 1(c)). For CIFAR-10, we set privacy cost ε to be 0.3158, 0.3587, 0.4221, 0.5130, 0.6546, 0.9067, 1.4949, 4.6978, which are obtained by training InsDP-FedAvg FL models for one round with noise level σ = 8, 7, 6, 5, 4, 3, 2, 1, respectively (Figure 1(d)).

With the aim to study certified attack cost under different number of adversarial instances k, for MNIST, we set the noise level σ to be 10. When k = 0, after training InsDP-FedAvg for T = 4, 9 rounds, we obtain FL models with privacy guarantee ε = 0.2383, 0.304 and clean accuracy (average over M runs) 96.40%, 96.93% (Figure 8(a)(b)). For CIFAR-10, we set the noise level σ to be 8.0. After training InsDP-FedAvg for one round under k = 0, we obtain FL models with privacy guarantee ε = 0.3158 and clean accuracy 61.78% (Figure 4(a)(b)).
In order to study the empirical attack cost and certified attack cost lower bound under different instance-level DP guarantee, we set the privacy guarantee $\epsilon$ to be 0.5016, 0.311, 0.2646, 0.2318, 0.2202, 0.2096, 0.205 for MNIST, which are obtained by training InsDP-FedAvg FL models for 6 rounds under noise level $\sigma = 5, 8, 10, 13, 15, 18, 20$, respectively, and the clean accuracy for the corresponding models are 99.60%, 98.81%, 97.34%, 92.92%, 88.01%, 80.94%, 79.60% (Figure 8(c)(d)). For CIFAR-10, we set the privacy guarantee $\epsilon$ to be 1.261, 0.9146, 0.7187, 0.5923, 0.5038, 0.4385, which are obtained by training InsDP-FedAvg FL models for 2 rounds under noise level $\sigma = 3, 4, 5, 6, 7, 8$, respectively, and the clean accuracy for the corresponding models are 84.47%, 80.99%, 76.01%, 68.65%, 63.07%, 60.65% (Figure 4(c)(d)).

With the intention of exploring the upper bound for $k$ given $\tau$ under different instance-level DP guarantee, for MNIST, we set noise level $\sigma$ to be 5, 8, 10, 13, 20, respectively, to obtain instance-DP FL models after 10 rounds with privacy guarantee $\epsilon = 0.6439, 0.3937, 0.3172, 0.2626, 0.2179$ and clean accuracy 99.58%, 98.83%, 97.58%, 95.23%, 85.72% (Figure 5(c)). For CIFAR-10, we set noise level $\sigma$ to be 3, 4, 5, 6, 7, 8 and train InsDP-FedAvg for $T = 3$ rounds, to obtain FL models with privacy guarantee $\epsilon = 1.5365, 1.1162, 0.8777, 0.7238, 0.6159, 0.5361$ and clean accuracy 84.34%, 80.27%, 74.62%, 66.94%, 62.14%, 59.75% (Figure 5(d)).

### A.3.3 Additional Implementation Details

**Threat Models** For the attacks against UserDP-FedAvg, by default, the local poison fraction $\alpha = 100\%$, and the scale factor $\gamma = 50$. We use same parameters setups for all $k$ attackers. In terms of label flipping attacks, the attackers swap the label of images in source class (digit 1 for MNIST; bird for CIFAR-10) into the target label (digit 0 for MNIST; airplane for CIFAR-10). In terms of backdoor attacks in MNIST and CIFAR-10, the attackers add a backdoor pattern, as shown in Figure 6(left), in images and swap the label of any sample with such pattern into the target label (digit 0 for MNIST; airplane for CIFAR-10). In terms of distributed backdoor attacks, Figure 6(right) shows an example when the triangle pattern is evenly decomposed into $k = 4$ parts, and they are used as the distributed patterns for $k = 4$ attackers respectively. For the cases where there are more or fewer distributed attackers, the similar decomposition strategy is adopted.

For the attacks against InsDP-FedAvg, the same target classes and backdoor patterns are used as UserDP-FedAvg. The parameters setups are the same for all $k$ poisoned instances.

**Robustness Certification** We certified 2115/2000 test samples from the MNIST/CIFAR-10 test sets. In Theorem 2 and Corollary 1 that are related to certified attack cost, $\bar{C}$ specifies the range of $C(\cdot)$. In the implementation, $\bar{C}$ is set to be larger than the maximum empirical attack cost evaluated on the test sets (see Table 1 for details). For each dataset, we use the same $\bar{C}$ for cost function $C$ defined in Example 1 and Example 2. When using Monte-Carlo sampling, we run $M = 1000$ times for certified accuracy, and $M = 100$ times for certified attack cost in all experiments.

**Machines** We simulate the federated learning setup (1 server and N users) on a Linux machine with Intel® Xeon® Gold 6132 CPUs and 8 NVidia® 1080Ti GPUs.

**Libraries** All code is implemented in Pytorch [65]. Please see the submitted code for full details.

### A.3.4 Certified Accuracy with Confidence Level

Here we present the certified accuracy with confidence level. We use Hoeffding’s inequality [61] to calibrate the empirical estimation with one-sided error tolerance $\psi$, i.e., one-sided confidence level $1 - \psi$. We first use Monte-Carlo sampling by running the private FL algorithms for $M$ times, with class confidence $f^c_x = f^c_x(\mathcal{M}(D),x)$ for class $c$ each time. We denote the empirical estimation as $\bar{f}^c_x(\mathcal{M}(D),x) = \frac{1}{M} \sum_{i=1}^{M} f^c_x$. For a test input $x$, suppose $A, B \in [C]$ satisfy $A = \arg\max_{c \in [C]} \bar{f}^c_x(\mathcal{M}(D),x)$ and $B = \arg\max_{c \in [C] x \neq A} \bar{f}^c_x(\mathcal{M}(D),x)$. For a given error tolerance $\psi$, we use Hoeffding’s inequality to compute a lower bound $F_A(\mathcal{M}(D),x)$ on the class confidence $F_A(\mathcal{M}(D),x)$ and a upper bound $F_B(\mathcal{M}(D),x)$ on the class confidence $F_B(\mathcal{M}(D),x)$ according to

$$F_A(\mathcal{M}(D),x) = \bar{f}_A(\mathcal{M}(D),x) - \sqrt{\log(1/\psi) / 2M}, \quad F_B(\mathcal{M}(D),x) = \bar{f}_B(\mathcal{M}(D),x) + \sqrt{\log(1/\psi) / 2M}.$$  

$F_A(\mathcal{M}(D),x)$ and $F_B(\mathcal{M}(D),x)$ are used as the expected class confidences for the evaluation of Theorem 2. We use $\psi = 0.01$ and $M = 1000$ for all experiments.
As shown in Figure 7, we can observe the same tradeoff between $\epsilon$ and certified accuracy as we discussed in Figure 1. In general, the $K$ in Figure 7 is smaller than the $K$ in Figure 1 because we calibrate the empirical estimation according to Eq. 6, and the class confidence gap between top-1 and top-2 class is narrowed.

A.3.5 Additional Robustness Evaluation of Instance-level DPFL

Here we report the robustness evaluation of instance-level DPFL on MNIST. As shown in Figure 8, the results on MNIST are similar to the results on CIFAR-10 in Figure 4.

As shown in Figure 7, we can observe the same tradeoff between $\epsilon$ and certified accuracy as we discussed in Figure 1. In general, the $K$ in Figure 7 is smaller than the $K$ in Figure 1 because we calibrate the empirical estimation according to Eq. 6, and the class confidence gap between top-1 and top-2 class is narrowed.

A.3.6 Robustness Evaluation on 10-class Classification

Here we report the robustness evaluation of user-level DPFL under backdoor attacks on 10-class classification problem. Figure 10 presents the certified accuracy under different $\epsilon$. We can observe the tradeoff between $\epsilon$ and certified accuracy on MNIST. On CIFAR-10, larger $k$ can be certified with smaller $\epsilon$. The certified $K$ is relatively small because we set large $\epsilon$ to preserve a reasonable accuracy for 10-class classification. Our results can inspire advanced DP mechanisms that provide tighter privacy guarantee (i.e., smaller $\epsilon$) while achieving similar level of accuracy. In terms of certified attack cost, as shown in Figure 9 and 11 the trends are similar to the 2-class results in Figure 2, 3 and 5.

A.3.7 DP bound comparison between InsDP-FedSGD and Dopamine

Here we compare Dopamine to our InsDP-FedSGD, both of which are proposed for FedSGD. Under the same noise level ($\sigma = 3.0$), clipping threshold ($S = 1.5$), user sampling probability ($m/N = 20/30$), and batch sampling
Figure 10: Certified accuracy of FL satisfying user-level DP on 10-class classification.

Figure 11: Lower bound of $k$ on 10-class classification under user-level $\epsilon$ given attack effectiveness $\tau$.

Figure 12: Comparison of DP bound $\epsilon$ under FedSGD on MNIST dataset. Our InsDP-FedSGD achieves a tighter DP bound.

Probability (0.4) settings, both algorithms achieve about 92% accuracy on MNIST (10 classes). The Figure 12 shows the results of privacy guarantee estimation over training rounds, which demonstrates that our method achieves tighter privacy certification. For instance, at round 200, our method ($\epsilon = 1.4029$) achieves a much tighter privacy guarantee than Dopamine ($\epsilon = 2.1303$).

A.4 Proofs of Certified Robustness Analysis

We restate our Lemma 1 here.

**Lemma 1** (Group DP). For mechanism $\mathcal{M}$ that satisfies $(\epsilon, \delta)$-DP, it satisfies $(ke, \frac{1 - e^k \epsilon}{1 - e^\epsilon})$-DP for groups of size $k$. That is, for any $d, d' \in D$ that differ by $k$ individuals, and any $E \subseteq \Theta$ it holds that $\Pr[\mathcal{M}(d) \in E] \leq e^{ke} \Pr[\mathcal{M}(d') \in E] + \frac{1 - e^k \epsilon}{1 - e^\epsilon} \delta$.

**Proof.** We denote $d$ as $d_0$, $d'$ as $d_k$. $d_i$ differ $i$ individuals with $d_0$. For any $i \in [1, k]$, $d_i$ and $d_{i-1}$ differ by one individual, thus

$$\Pr[M(d_{i-1})] \leq e^\epsilon \Pr[M(d_i)] + \delta. \quad (7)$$

By iteratively applying Eq. (7) $k$ times, we have

$$\Pr[M(d_0)] \leq e^{ke} \Pr[M(d_k)] + (1 + e^\epsilon + e^{2\epsilon} + \ldots + e^{(k-1)\epsilon}) \delta$$

$$= e^{ke} \Pr[M(d_k)] + \frac{1 - e^{ke}}{1 - e^\epsilon} \delta.$$

Before we prove Theorem 1, we introduce the following lemma:

**Lemma 3.** Suppose a randomized mechanism $\mathcal{M}$ satisfies user-level $(\epsilon, \delta)$-DP. For two user sets $B$ and $B'$ that differ by one user, $D$ and $D'$ are the corresponding training datasets. For a test input $x$, for any $c \in [C]$, $f_c(\mathcal{M}(D), x) \in [0, 1]$ is the class confidence, then the expected class confidence $F_c(\mathcal{M}(D), x) := \mathbb{E}[f_c(\mathcal{M}(D), x)]$ meets the following property:

$$F_c(\mathcal{M}(D), x) \leq e^\epsilon F_c(\mathcal{M}(D'), x) + \delta \quad (8)$$
We recall Theorem 1.

\[ F_c(\mathcal{M}(D), x) = \mathbb{E}[f_c(\mathcal{M}(D), x)] = \int_0^1 \mathbb{P}[f_c(\mathcal{M}(D), x) > a] \, da \]
\[ = \int_0^1 \mathbb{P}[\mathcal{M}(D) \in \Theta(a)] \, da \]
\[ \leq \int_0^1 (e^\varepsilon \mathbb{P}[\mathcal{M}(D') \in \Theta(a)] + \delta) \, da \]
\[ = \int_0^1 e^\varepsilon \mathbb{P}[f_c(\mathcal{M}(D'), x) > a] \, da + \int_0^1 \delta \, da \]
\[ = e^\varepsilon F_c(\mathcal{M}(D'), x) + \delta \]

\[ \square \]

We recall Theorem 1

**Theorem 1 (Condition for Certified Prediction under One Adversarial User).** Suppose a randomized mechanism \( \mathcal{M} \) satisfies user-level \((\varepsilon, \delta)\)-DP. For two user sets \( B \) and \( B' \) that differ by one user, let \( D \) and \( D' \) be the corresponding training datasets. For a test input \( x \), suppose \( \mathcal{A}, \mathcal{B} \in \mathcal{C} \) satisfy \( \mathcal{A} = \arg \max_{c \in [C]} F_c(\mathcal{M}(D), x) \) and \( \mathcal{B} = \arg \max_{c \in [C], c \neq A} F_c(\mathcal{M}(D), x) \), then

\[ F_{\mathcal{A}}(\mathcal{M}(D), x) > e^{2\varepsilon} F_{\mathcal{B}}(\mathcal{M}(D), x) + (1 + e^\varepsilon)\delta, \] (1)

it is guaranteed that \( H(\mathcal{M}(D'), x) = H(\mathcal{M}(D), x) = \mathcal{A} \).

**Proof.** According to Lemma 3

\[ F_{\mathcal{A}}(\mathcal{M}(D), x) \leq e^\varepsilon F_{\mathcal{A}}(\mathcal{M}(D'), x) + \delta \] (9)

\[ F_{\mathcal{B}}(\mathcal{M}(D'), x) \leq e^\varepsilon F_{\mathcal{B}}(\mathcal{M}(D), x) + \delta. \] (10)

Then

\[ F_{\mathcal{A}}(\mathcal{M}(D'), x) \geq \frac{F_{\mathcal{A}}(\mathcal{M}(D), x) - \delta}{e^\varepsilon} \] (Because of Eq. 9)
\[ \geq \frac{e^{2\varepsilon} F_{\mathcal{B}}(\mathcal{M}(D), x) + (1 + e^\varepsilon)\delta - \delta}{e^\varepsilon} \] (Because of the given condition Eq. 1)
\[ = \frac{e^\varepsilon F_{\mathcal{B}}(\mathcal{M}(D), x) + \delta}{e^\varepsilon} \]
\[ \geq e^\varepsilon \left( \frac{F_{\mathcal{B}}(\mathcal{M}(D'), x) - \delta}{e^\varepsilon} \right) + \delta \] (Because of Eq. 10)
\[ = F_{\mathcal{B}}(\mathcal{M}(D'), x), \]

which indicates that the prediction of \( \mathcal{M}(D') \) at \( x \) is \( \mathcal{A} \) by definition.

\[ \square \]

Before we prove Theorem 2, we introduce the following lemma:

**Lemma 4.** Suppose a randomized mechanism \( \mathcal{M} \) satisfies user-level \((\varepsilon, \delta)\)-DP. For two user sets \( B \) and \( B' \) that differ \( k \) users, \( D \) and \( D' \) are the corresponding training datasets. For a test input \( x \), for any \( c \in [C], f_c(\mathcal{M}(D), x) \in [0, 1] \) is the class confidence, then the expected class confidence \( F_c(\mathcal{M}(D), x) := \mathbb{E}[f_c(\mathcal{M}(D), x)] \) meets the following property:

\[ F_c(\mathcal{M}(D), x) \leq e^{\varepsilon k} F_c(\mathcal{M}(D'), x) + \frac{1 - e^{\varepsilon k}}{1 - e^\varepsilon} \delta \] (11)
Proof. Define $\Theta(a) := \{\theta : f_\theta(x) > a\}$. Then

$$F_c(\mathcal{M}(D), x) = \int_0^1 P[f_c(\mathcal{M}(D), x) > a] \, da$$

$$= \int_0^1 P[\mathcal{M}(D) \in \Theta(a)] \, da$$

$$\leq \int_0^1 \left( e^{ke} P[\mathcal{M}(D') \in \Theta(a)] + \frac{1 - e^{ke}}{1 - e^\delta} \right) \, da \quad \text{(Because of Group DP property in Lemma[1])}$$

$$= \int_0^1 e^{ke} P[f_c(\mathcal{M}(D'), x) > a] \, da + \int_0^1 \frac{1 - e^{ke}}{1 - e^\delta} \, da$$

$$= e^{ke} F_c(\mathcal{M}(D'), x) + \frac{1 - e^{ke}}{1 - e^\delta}$$

We recall Theorem[2].

**Theorem 2** (Upper Bound of $k$ for Certified Prediction). Suppose a randomized mechanism $\mathcal{M}$ satisfies user-level $(\epsilon, \delta)$-DP. For two user sets $B$ and $B'$ that differ by $k$ users, let $D$ and $D'$ be the corresponding training datasets. For a test input $x$, suppose $A, B \in [C]$ satisfy $A = \text{arg max}_{c \in [C]} F_c(\mathcal{M}(D), x)$ and $B = \text{arg max}_{c \in [C], c \neq A} F_c(\mathcal{M}(D), x)$, then $H(\mathcal{M}(D'), x) = H(\mathcal{M}(D), x) = A$, $\forall k < K$ where $K$ is the certified number of adversarial users:

$$K = \frac{1}{2\epsilon} \log \frac{F_A(\mathcal{M}(D), x)(e^\epsilon - 1) + \delta}{\int F_B(\mathcal{M}(D), x)(e^\epsilon - 1) + \delta} \quad (2)$$

Proof. According to Lemma[4] we have

$$F_A(\mathcal{M}(D), x) \leq e^{ke} F_A(\mathcal{M}(D'), x) + \frac{1 - e^{ke}}{1 - e^\delta} \quad (12)$$

$$F_B(\mathcal{M}(D'), x) \leq e^{ke} F_B(\mathcal{M}(D), x) + \frac{1 - e^{ke}}{1 - e^\delta} \quad (13)$$

We can re-write the given condition $k < K$ according to Eq. (2) as

$$e^{2ke} F_B(\mathcal{M}(D), x) + (1 + e^{ke}) \frac{1 - e^{ke}}{1 - e^\delta} < F_A(\mathcal{M}(D), x). \quad (14)$$

Then

$$F_A(\mathcal{M}(D'), x) \geq F_A(\mathcal{M}(D), x) - \frac{1 - e^{ke}}{e^{ke}} \quad \text{(Because of Eq. [12])}$$

$$> \frac{e^{2ke} F_B(\mathcal{M}(D), x) + (1 + e^{ke}) \frac{1 - e^{ke}}{1 - e^\delta} - \frac{1 - e^{ke}}{1 - e^\delta}}{e^{ke}} \quad \text{(Because of the given condition Eq [14])}$$

$$= e^{ke} F_B(\mathcal{M}(D), x) + \frac{1 - e^{ke}}{1 - e^\delta}$$

$$\geq e^{ke} \left( \frac{F_B(\mathcal{M}(D'), x) - \frac{1 - e^{ke}}{1 - e^\delta}}{e^{ke}} \right) + \frac{1 - e^{ke}}{1 - e^\delta} \quad \text{(Because of Eq. [13])}$$

$$= F_B(\mathcal{M}(D'), x),$$

which indicates that the prediction of $\mathcal{M}(D')$ at $x$ is $A$ by definition. \qed
**Theorem 3** (Attack Cost with $k$ Attackers). Suppose a randomized mechanism $M$ satisfies user-level $(\varepsilon, \delta)$-DP. For two user sets $B$ and $B'$ that differ $k$ users, $D$ and $D'$ are the corresponding training datasets. Let $J(D)$ be the expected attack cost where $|C(\cdot)| \leq \tilde{C}$. Then,

$$
\min \{ e^{k\varepsilon} J(D) + \frac{e^{k\varepsilon} - 1}{e^{\varepsilon} - 1} \delta \tilde{C}, \tilde{C} \} \geq J(D') \geq \max \{ e^{-k\varepsilon} J(D) - \frac{1}{e^{\varepsilon} - 1} \delta \tilde{C}, 0 \}, \quad \text{if } C(\cdot) \geq 0
$$

$$
\min \{ e^{-k\varepsilon} J(D) + \frac{1 - e^{-k\varepsilon}}{e^{\varepsilon} - 1} \delta \tilde{C}, 0 \} \geq J(D') \geq \max \{ e^{k\varepsilon} J(D) - \frac{e^{k\varepsilon} - 1}{e^{\varepsilon} - 1} \delta \tilde{C}, -\tilde{C} \}, \quad \text{if } C(\cdot) \leq 0
$$

(3)

**Proof.** We first consider $C(\cdot) \geq 0$. Define $\Theta(a) = \{ \theta : C(\theta) > a \}$.

$$
J(D) = \int_{\tilde{C}}^{\tilde{C}} P[C(M(D)) > a] \, da
= \int_0^{\tilde{C}} P[M(D)] \in \Theta(a)] \, da
\leq \int_0^{\tilde{C}} \left( e^{k\varepsilon} P[M(D')] \in \Theta(a)] + \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C} \right) \, da
$$

(Because of Group DP property in Lemma 1)

$$
= \int_0^{\tilde{C}} e^{k\varepsilon} P[M(D')] \in \Theta(a)] \, da + \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}
= \int_0^{\tilde{C}} e^{k\varepsilon} P[C(M(D')) > a] \, da + \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}
= e^{k\varepsilon} J(D') + \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}
$$

i.e.,

$$
J(D') \geq e^{-k\varepsilon} J(D) - \frac{1}{e^{\varepsilon} - 1} \delta \tilde{C}.
$$

Switch the role of $D$ and $D'$, we have

$$
J(D') \leq e^{k\varepsilon} J(D) + \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}.
$$

Also note that $0 \leq J(D') \leq \tilde{C}$ trivially holds due to $0 \leq C(\cdot) \leq \tilde{C}$, thus

$$
\min \{ e^{k\varepsilon} J(D) + \frac{e^{k\varepsilon} - 1}{e^{\varepsilon} - 1} \delta \tilde{C}, \tilde{C} \} \geq J(D') \geq \max \{ e^{-k\varepsilon} J(D) - \frac{1}{e^{\varepsilon} - 1} \delta \tilde{C}, 0 \}.
$$

Next we consider $C(\cdot) \leq 0$. Define $\Theta(a) = \{ \theta : C(\theta) < a \}$.

$$
J(D) = -\int_{-\tilde{C}}^{0} P[C(M(D)) < a] \, da
= -\int_{-\tilde{C}}^{0} P[M(D)] \in \Theta(a)] \, da
\geq -\int_{-\tilde{C}}^{0} \left( e^{k\varepsilon} P[M(D')] \in \Theta(a)] + \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C} \right) \, da
$$

(Because of Group DP property in Lemma 1)

$$
= -\int_{-\tilde{C}}^{0} e^{k\varepsilon} P[M(D')] \in \Theta(a)] \, da - \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}
= -\int_{-\tilde{C}}^{0} e^{k\varepsilon} P[C(M(D')) < a] \, da - \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}
= e^{k\varepsilon} J(D') - \frac{1 - e^{k\varepsilon}}{1 - e^{\varepsilon}} \delta \tilde{C}
$$

i.e.,

$$
J(D') \leq e^{-k\varepsilon} J(D) + \frac{1 - e^{-k\varepsilon}}{e^{\varepsilon} - 1} \delta \tilde{C}.
$$
We note that all the above robustness certification related proofs are built upon the user-level DP property. According to Definition 1 and Definition 2, the definition of user-level DP and instance-level DP are both induced from DP (Definition 1) despite the different definitions of adjacent datasets. By applying the definition of instance-level (ε, δ)-DP and following the proof steps of Theorem 1, 2, 3 and Corollary 1, we can derive the similar theoretical conclusions for instance-level DP, leading to Theorem 5 to achieve the certifiably robust FL for free given the DP property.

Switch the role of D and D', we have

\[ J(D') \geq e^{\epsilon k} J(D) - \frac{1 - e^{\epsilon k}}{1 - e^\epsilon} \delta \hat{C}. \]

Also note that \(-\hat{C} \leq J(D') \leq 0\) trivially holds due to \(-\hat{C} \leq C(\cdot) \leq 0\), thus

\[ \min \{e^{-\epsilon k} J(D) + \frac{1 - e^{-\epsilon k}}{e^\epsilon - 1} \delta \hat{C}, 0\} \geq J(D') \geq \max \{e^{\epsilon k} J(D) - \frac{e^{\epsilon k} - 1}{e^\epsilon - 1} \delta \hat{C}, -\hat{C}\} \]

We recall Corollary 1.

**Corollary 1 (Lower Bound of k Given τ).** Suppose a randomized mechanism \( \mathcal{M} \) satisfies user-level (ε, δ)-DP. Let attack cost function be \( C \), the expected attack cost be \( J(J) \). In order to achieve \( J(D') \leq \frac{1}{\tau} J(D) \) for \( \tau \geq 1 \) when \( 0 \leq C(\cdot) \leq \hat{C} \), or achieve \( J(D') \leq \tau J(D) \) for \( 1 \leq \tau \leq -\frac{C}{\hat{C}} \) when \(-\hat{C} \leq C(\cdot) \leq 0\), the number of adversarial users should satisfy:

\[ k \geq \frac{1}{e} \log \frac{(e^\epsilon - 1) J(D) \tau + \hat{C} \delta \tau}{(e^\epsilon - 1) J(D) + \hat{C} \delta} \quad \text{or} \quad k \geq \frac{1}{e} \log \frac{(e^\epsilon - 1) J(D) \tau - \hat{C} \delta}{(e^\epsilon - 1) J(D) - \hat{C} \delta} \]

respectively. (4)

**Proof.** We first consider \( C(\cdot) \geq 0 \). According to the lower bound in Theorem 3 when \( B' \) and \( B \) differ \( k \) users, \( J(D') \geq e^{-\epsilon k} J(D) - \frac{1 - e^{-\epsilon k}}{e^\epsilon - 1} \delta \hat{C} \). Since we require \( J(D') \leq \frac{1}{\tau} J(D) \), then \( e^{-\epsilon k} J(D) - \frac{1 - e^{-\epsilon k}}{e^\epsilon - 1} \delta \hat{C} \leq \frac{1}{\tau} J(D) \). Rearranging gives the result.

Next, we consider \( C(\cdot) \leq 0 \). According to the lower bound in Theorem 3 when \( B' \) and \( B \) differ \( k \) users, \( J(D') \geq e^{\epsilon k} J(D) - \frac{\delta \hat{C}}{e^\epsilon - 1} \). Since we require \( J(D') \leq \tau J(D) \), then \( e^{\epsilon k} J(D) - \frac{\delta \hat{C}}{e^\epsilon - 1} \leq \tau J(D) \). Rearranging gives the result.

We note that all the above robustness certification related proofs are built upon the user-level (ε, δ)-DP property and the Group DP property. According to Definition 2 and Definition 3, the definition of user-level DP and instance-level DP are both induced from DP (Definition 1) despite the different definitions of adjacent datasets. By applying the definition of instance-level (ε, δ)-DP and following the proof steps of Theorem 1, 2, 3, 5 and Corollary 1, we can derive the similar theoretical conclusions for instance-level DP, leading to Theorem 5 to achieve the certifiably robust FL for free given the DP property.