RecUP-FL: Reconciling Utility and Privacy in Federated learning via User-configurable Privacy Defense

Yue Cui
University of Tennessee
Knoxville, TN, US
ycui22@vols.utk.edu

Syed Irfan Ali Meerza
University of Tennessee
Knoxville, TN, US
smeerza@vols.utk.edu

Zhuohang Li
University of Tennessee
Knoxville, TN, US
zli96@vols.utk.edu

Luyang Liu
Rutgers University
New Brunswick, NJ, US
luyang@winlab.rutgers.edu

Jiaxin Zhang
Intuit AI Research
Mountain View, CA, US
jiaxin_zhang@intuit.com

Jian Liu
University of Tennessee
Knoxville, TN, US
jliu@utk.edu

ABSTRACT

Federated learning (FL) provides a variety of privacy advantages by allowing clients to collaboratively train a model without sharing their private data. However, recent studies have shown that private information can still be leaked through shared gradients. To further minimize the risk of privacy leakage, existing defenses usually require clients to locally modify their gradients (e.g., differential privacy) prior to sharing with the server. While these approaches are effective in certain cases, they regard the entire data as a single entity to protect, which usually comes at a large cost in model utility. In this paper, we seek to reconcile utility and privacy in FL by proposing a user-configurable privacy defense, RecUP-FL, that can better focus on the user-specified sensitive attributes while obtaining significant improvements in utility over traditional defenses. Moreover, we observe that existing inference attacks often rely on a machine learning model to extract the private information (e.g., attributes). We thus formulate such a privacy defense as an adversarial learning problem, where RecUP-FL generates slight perturbations that can be added to the gradients before sharing to fool adversary models. To improve the transferability to un-queryable black-box adversary models, inspired by the idea of meta-learning, RecUP-FL forms a model zoo containing a set of substitute models and iteratively alternates between simulations of the white-box and the black-box adversarial attack scenarios to generate perturbations. Extensive experiments on four datasets under various adversarial settings (both attribute inference attack and data reconstruction attack) show that RecUP-FL can meet user-specified privacy constraints over the sensitive attributes while significantly improving the model utility compared with state-of-the-art privacy defenses.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies → Machine learning**;

KEYWORDS

federated learning; privacy defense; user-configurable; meta-learning

ACM Reference Format:
Yue Cui, Syed Irfan Ali Meerza, Zhuohang Li, Luyang Liu, Jiaxin Zhang, and Jian Liu. 2023. RecUP-FL: Reconciling Utility and Privacy in Federated learning via User-configurable Privacy Defense. In *ACM ASIA Conference on Computer and Communications Security (ASIA CCS '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3579856.3582819

1 INTRODUCTION

Over the past few years, deep learning models are being integrated into more and more mobile and edge/IoT applications to bring convenience to the users and help improve the user experience. Successful applications can be found in almost every business sector, including but not limited to personal shopping recommendation [54], speech recognition [18], smart healthcare [29], and fraud prevention in mobile banking [37]. However, to power these intelligent applications, massive data need to be gathered from end users, which would inevitably cause privacy concerns.

Federated learning (FL) [35], an emerging platform for distributed machine learning, has recently received considerable attention for its privacy benefits. In a typical FL system, a central *server* coordinates multiple data providers (i.e., *clients*) to collaboratively train a machine learning model. To protect privacy, clients do not directly share their private data during this learning process. Instead, the server and the clients exchange focused model updates (e.g., gradients) to achieve the learning objective. While offering practical privacy improvements over traditional centralized learning schemes, there is still no formal privacy guarantee in this vanilla form of FL [28]. In fact, prior research has shown that different levels of private information may still be leaked through the shared model updates, ranging from membership information [36], sensitive attributes [13, 34], to even complete reconstruction of private training data samples [15, 55].

In response to these threats, several privacy-preserving techniques have been proposed. For instance, secure multiparty computation (MPC) [9, 11] seeks to leverage cryptographic solutions to secure "*how it is computed*" so that only the results of the computation are revealed to the intended parties. In addition, given the potential threats from other clients or malicious eavesdroppers on the client's communication channel, a formal privacy guarantee

ASIA CCS '23, July 10–14, 2023, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0098-9/23/07…$15.00
https://doi.org/10.1145/3579856.3582819

is needed on such client-level basis to protect "*what is computed*" (i.e., model updates shared by participating clients). For instance, to prevent gradient leakage without requiring trust in a centralized server, existing strategies require clients to apply a *"local transformation"* to their gradients before sharing with the server, such as applying local differential privacy (local-DP) [16, 46], gradient compression [53] and representation perturbation [44], etc.

Limitations of Existing Efforts. Despite offering remarkable privacy improvement, existing client-level approaches usually require adding a significant amount of noise (e.g., local-DP [16, 46]) or largely modifying the gradients [44, 53], which will inevitably depreciate the utility and usability of the resulting model. Moreover, it is usually more challenging to maintain a reasonable utility-privacy trade-off with these client-level approaches. For instance, local-DP requires adding much higher noise than what is required by central-DP[1]. On the other hand, existing solutions consider each client's private data as a single entity and attempt to protect all attributes, even including the ones that are helpful for the target learning task, and therefore they often come at a large cost in model utility.

Key Insights. To improve the utility-privacy trade-offs of privacy defenses in FL, we draw inspirations from the following key insights: (1) Users may value different aspects of privacy differently, which may result in different privacy requirements [41]. For example, people may have different comfort level sharing certain attributes, such as their political view, sexual orientation or religion. Thus, the actual privacy requirements can be relaxed by allowing user-specific privacy configurations. (2) Recent data protection regulations, such as General Data Protection Regulation (GDPR) [1], and California Consumer Privacy Act (CCPA) [2], require giving data providers (e.g., clients in FL) their explicit consent to collect and process their sensitive personal data. GDPR also makes a clear distinction between sensitive personal data and non-sensitive personal data, and sensitive data have more stringent requirements in terms of data collection and processing. While the non-sensitive data do not need to be treated with extra security, existing privacy defenses in FL regard all data equally sensitive and aim to sanitize the gradients to protect the whole data. Thus, we believe there is still much room for further reconciling utility and privacy preservation in FL. (3) Most privacy leakages through gradients are caused by inference attacks, which usually rely on a machine learning model to learn the mapping between the exchanged model updates and private attributes. On the other hand, learning models have been proven to be naturally vulnerable to adversarial examples [17], which can potentially be leveraged to mitigate such privacy leakage.

Our Solution. Based on these insights, in this paper, we propose RecUP-FL, the first user-configurable local privacy defense framework that seeks to reconcile the utility and privacy in FL. Unlike existing solutions that attempt to protect clients' entire training data, our objective is to focus on protecting *a subset of* sensitive attributes specified by each user (i.e., client) according to their privacy preferences. By relaxing the privacy constraints on the non-sensitive attributes, RecUP-FL can achieve a relaxed notion of privacy that better focuses on the identified sensitive attributes and at the same time obtain significant improvements in model utility over traditional defenses. For instance, voice data carry a set of sensitive information besides speech contents, such as identity, personality, geographical origin, emotions, gender and age, etc [13]. Users would value the privacy of these attributes differently given where and how their voice-controllable devices are used. Similarly, images also contain different types of sensitive information, such as visited location, nationality, and fingerprint, etc. A wide user study conducted by Orekondy *et al.* [41] shows that most people think the leakage of fingerprint extremely violates their privacy while nationality is the least private information. RecUP-FL provides a means for users to select any sensitive attributes that they would like to protect from all carried information before participating in the training, which can enhance privacy and raise their willingness to get involved in the training.

To achieve maximized privacy (i.e., reduce the attack success rate of attackers who leverage learning models to launch inference attacks as much as possible), RecUP-FL is designed to be a local defense solution: at each communication round in FL, besides computing the model updates, each client also locally computes a perturbation based on the specified sensitive attributes and only shares the perturbed model updates to the server. In this way, the clients can ensure their targeted data privacy without trusting any other parties. To protect specified sensitive attributes while maintaining a good level of utility, for each user-specified sensitive attribute, we formulate the defense as an optimization problem where the goal is to find the minimal perturbation that can prevent the adversary from making the correct prediction. Such formulation is equivalent to launching an adversarial attack against the adversary model that aims to classify the victim's sensitive attributes from the shared model updates. However, computing such perturbations is not trivial, since the clients (1) possess no information about the configuration of the adversary model, including model parameters, model architecture, and even model type (e.g., Random Forest or Neural Network); and (2) do not have any query access to the adversary model, which makes existing query-based black-box adversarial attack methods [25] inapplicable.

To tackle the aforementioned challenges, RecUP-FL adopts a two-stage method for calculating the defensive perturbation. In the first stage, RecUP-FL forms a model zoo by loading a set of pre-trained substitute models (referred to as *defender models*). For each selected sensitive attribute, the defender models mimics the behavior of the adversary by attempting to infer the attribute from the clients' model updates. In the second stage, RecUP-FL obtains the perturbations by launching an adversarial attack against the defender models. To enable the defense to be generalizable to unseen adversary models, RecUP-FL exploits the meta gradient adversarial attack [50] to improve the transferability of the calculated adversarial perturbations. Note that RecUP-FL targets ex-post empirical privacy instead of providing a formal differential privacy guarantee. We compare RecUP-FL with four state-of-the-art baseline privacy defenses, including applying local differential privacy with both Gaussian noise [46] and Laplace noise [31], gradient sparsification [30], and Soteria [44], under different settings with various types of threat models. We consider two settings of threat models: (1) *a third party eavesdropping on the communication channel* and (2) *an honest but curious central server*. Both of them are able to

[1] Local-DP requires a lower bound of noise magnitude $\Omega(\sqrt{n}/\epsilon)$, while the central-DP only requires $O(1/\epsilon)$, where n is the number of participating clients and ϵ is the privacy loss [47].

infer the potential sensitive attribute information without affecting the training process. The results show that the proposed RecUP-FL can maximize the model utility and satisfy user-specified privacy constraints against various privacy attacks.

Contributions. We summarize our main contributions as follows:

- To the best of our knowledge, RecUP-FL is the first framework that seeks to reconcile utility and privacy via user-configurable local privacy defenses in FL .
- To improve utility and privacy trade-off, RecUP-FL finds the minimal perturbation for protecting user-specified attributes by generating adversarial examples against a set of substitute defender models.
- In order to improve the generalizability of RecUP-FL over unseen and un-queryable adversary models, we exploit the meta gradient adversarial attack method to iteratively improve the transferability of the defense by leveraging a collection of carefully-configured defender models.
- We evaluate the proposed RecUP-FL on four datasets, including AudioMNIST, Adult Income, LFW and CelebA, under various adversary settings. We show that RecUP-FL is able to resist both attribute inference and data reconstruction attacks while achieving better utility-privacy trade-offs.

2 RELATED WORK

2.1 Privacy Leakage in Federated Learning

Attribute Inference Attack. Attribute inference attack aims to infer certain input attributes of the client's private training data through analyzing shared gradient information. This type of attack was first formulated in centralized learning against Hidden Markov Models (HMMs) and Support Vector Machine (SVM) classifiers [6] and then was extended to work on fully connected neural networks (FCNNs) [14] to determine whether the training data has a certain set of properties. In FL settings, Hitaj *et al.* [20] considered the adversary works as a client inside the privacy-preserving collaborative protocol and aims to infer class-representative information about a label that the adversary does not own. Further, Melis *et al.* [36] showed that an adversarial client can infer certain attributes of another client that are independent of its training task based on the exchanged model gradients (e.g., whether people in the training data wear glasses in a gender classification task). More recently, Lyu *et al.* [34] considered a more practical scenario where clients share their epoch-averaged gradients instead of small batch-averaged gradients, and an honest but curious server will infer the sensitive attributes of local training data via a gradient-matching-based method. Feng *et al.* [13] proposed an attribute inference attack that can infer sensitive attributes (e.g., the client's gender information) from shared gradients while training a speech emotion recognition classifier via shadow training.

Data Reconstruction Attack. Prior studies showed the possibility of recovering class-level [20] or even client-level [45] data representatives through generative models. More recent studies [15, 49, 52, 55] showed that an adversary could even fully restore the training data from its shared gradient information. Specifically, Zhu *et al.* recently [55] proposed to solve this gradient inversion problem by solving for the optimal pair of input and label that best matches the exchanged gradients. As a follow-up study, Zhao *et al.* [52] provided an analytical computation method to precisely infer the label information by performing binary classification to the direction of the last layer's gradient. It can effectively involve label information in the reconstruction process and thus improve the attack performance. However, they can only work on shallow network architectures with low-resolution images. To launch such attacks in more realistic scenarios, Geiping *et al.* [15] proposed to use a magnitude-invariant design along with various optimization strategies to restore ImageNet-level high-resolution data in large batch size from deeper networks (e.g, ResNet [19]). Yin *et al.* [49] also achieved image batch reconstruction by utilizing batch normalization statistics and image fidelity regularization.

2.2 Privacy Defenses in Federated Learning

Crypto-based Defenses. One line of defense strategy is to protect the aggregation of model updates through secure multi-party computation (MPC) [9, 11, 39], where a set of parties jointly compute a common function of interest without revealing their private inputs to other parties. For instance, Danner *et al.* [11] proposed a secure sum protocol using a tree topology and homomorphic encryption. SecureML [39] adopt a two-server model for preserving privacy, in which clients process, encrypt, and/or secret-share their data among two non-colluding servers to train a global model. Additionally, Bonawitz *et al.* [9] require the aggregation of model updates in FL to be logically performed by the virtual, incorruptible third party so that the server can only receive the aggregated model update. However, these crypto-based methods would inevitably cause high computational overhead, and recent inference attacks [36] showed that the adversary can still reveal private information even though they can only access the aggregated model update. Therefore, to ensure rigorous privacy guarantees in FL, secure computation techniques is usually deployed in parallel with the techniques for privacy-preserving disclosure, such as gradient-degradation-based defenses [28] to be mentioned next.

Gradient-degradation-based Defenses. To prevent privacy leakage via shared gradient information in FL, a very straightforward way is to intentionally "degrade" the fidelity of gradients on the client prior to sharing them with the server. As a standard and common method, differential privacy (DP) can be either applied at the client side (i.e., local DP) or the server side (i.e., central DP) to perturb the client's shared gradients and the aggregated gradients [16, 46], respectively, and thereby mitigate privacy risks. Compared with central DP, local DP usually provides a better notion of privacy since it does not require trust in a centralized server. However, local DP requires injecting random noises to the gradients at a large number of clients, making this local approach often come at a large cost in utility. Zhao *et al.* [53] theoretically and empirically proved that DP makes data private by adding a significantly large amount of noise, but it simultaneously filters out much useful information. In addition to DP, Zhu *et al.* [55] demonstrated that performing gradient sparsification (i.e., gradients with small magnitudes are pruned to zero) can also help prevent privacy leakage from the gradient. A more recent work, Soteria [44], proposed to compute the gradients based on perturbed data representations to maintain a good level of model utility while achieving a certified robustness guarantee to FL. While these gradient-degradation-based methods can mitigate privacy risks in certain cases, they can only achieve a sub-optimal utility-privacy trade-off because they treat the entire training data (including non-sensitive information) as a single entity, thereby redundantly degrading the gradients' fidelity. Different from the above methods, RecUP-FL only protects the sensitive attributes identified by users instead of the whole data, thus obtaining an improved utility-privacy trade-off. The general framework of our approach is under the umbrella of context-aware privacy defenses (e.g., [22, 23, 42]), which can leverage the context (e.g., dataset statistics, dataset's utility) to achieve better utility-privacy trade-offs. However, to the best of our knowledge, this is the first work that leverages the knowledge of user-specific sensitive attributes to improve utility-privacy trade-offs in FL.

3 PROBLEM FORMULATION & RECUP-FL DESIGN OBJECTIVES

3.1 Problem Formulation

Federated Learning. Without loss of generality, we assume there are K participating clients $\mathbf{C} = \{C_1, ..., C_K\}$ and one central server in our FL setting. The clients $\mathbf{C}$ will collaboratively train a global model $\mathcal{G}$ under the organization of the central server. The client C_i holds its local data D_i which is composed by N individual data records $(\mathbf{X}_i, \mathbf{Y}_i) = \{(x_{i,1}, y_{i,1}), ..., (x_{i,n}, y_{i,n})\}, n \in [1, N]$, where $x_{i,n}$ denotes the n-th data sample in the i-th client, and $y_{i,n}$ denotes its corresponding label of the training FL task.

At the beginning of the FL process, the central server first initializes the global model $\mathcal{G}$ with random initial weights w_0. After initialization, the central server repeatedly interacts with clients for T communication rounds until the global model converges. Each communication round $t \in [1, T]$ contains the following steps:

- **Step 1: Synchronization.** The central model sends the current model $\mathcal{G}$ with weights w_t to all K clients.
- **Step 2: Local Training.** Each client C_i performs one or more training steps on the received model $\mathcal{G}$ using its local data D_i. After training, each client sends its model update (i.e., gradients) $\nabla w_{t,i}$ back to the central server.
- **Step 3: Aggregation.** The central server averages all participating clients' model updates to update the global model [35]: $w_{t+1} = w_t - \alpha \cdot \sum_{i=1}^{K} \frac{\nabla w_{t,i}}{K}$ via gradient descent, where α is the learning rate.

Unlike the centralized training scheme that requires clients to send their local data D_i to the central server, FL only requires the model updates $\nabla w_{t,i}$ to be shared with the central server, and thereby the privacy concerns can be mitigated.

Threat Models. Despite the fact that training data can be kept locally in FL, the shared model updates still carry much sensitive information about the local data, which can be leveraged by an adversary to gain knowledge of the client's private data. In order to evaluate our defense under the worst-case scenario, we assume a very powerful adversary with the ability to access each client's update. In practice, the adversary can be either (1) *a third party* outside the training process eavesdropping on the communication channel [51], gathering the model updates, and launching attacks, or (2) *an honest but curious central server*, who executes the regular training procedure but also attempts to infer the client's private information from the received model updates [34]. The goal of the adversary is to reveal as much sensitive information about the client's private data as possible.

To investigate the worst-case scenario, we consider two critical privacy leakage attacks in FL: attribute inference attack and data reconstruction attack. In the attribute inference attack, the adversary aims to infer the sensitive attributes of the client's private training data from the shared model updates. As stated in prior studies [14, 43], it usually builds an adversary model $\mathcal{I}$, intercepts the model updates $\nabla w_{t,i}$, and infers the sensitive attribute value a_i utilizing the pre-trained adversary model (i.e., $a_i^{'} = \mathcal{I}(\nabla w_{t,i})$). In the data reconstruction attack, the adversary aims to completely recover the client's private training data X_i from the shared model updates by solving a gradient-matching optimization problem [15, 52, 55] as $X_i^{'} = \mathcal{R}(\nabla w_{t,i}))$ where $\mathcal{R}$ denotes the reconstructor.

Privacy Defense. A common way to defeat inference attack in FL is to apply a defensive local transformation function $\varphi(\cdot)$ on the model update before sharing as follows:

$$\nabla w'_{t,i} = \varphi(\nabla w_{t,i}), \tag{1}$$

where $\nabla w'_{t,i}$ is the perturbed model update after applying the transformation. We consider the following four state-of-the-art defensive transformation functions:

(1) **DP (Gaussian)** [46]: One of the most common solutions to defend against privacy leakage attacks in FL is DP (Gaussian). It restricts the model updates within the given clipping bound B by $Clip(\nabla w_{t,i}, B) = \frac{\nabla w_{t,i}}{\max(1, \|\nabla w_{t,i}\|_2/B)}$. Then it injects Gaussian noise on the clipped model updates before sharing by $\varphi_g(\nabla w_{t,i}, \mu, \sigma, B) = Clip(\nabla w_{t,i}, B) + \mathcal{N}(\mu, \sigma^2)$, where $\mathcal{N}(\cdot)$ is a normal distribution, μ and σ are the mean and standard deviation of the noise.

(2) **DP (Laplace)** [31]: Similar to DP (Gaussian), given the location μ, the scale b and clipping bound B, the transformation function of adding Laplace noise is $\varphi_l(\nabla w_{t,i}, \mu, b, B) = Clip(\nabla w_{t,i}, B) + Lap(\mu, b)$, where $Lap(\cdot)$ denotes the Laplace distribution.

(3) **Gradient Sparsification** [30]: Gradient sparsification is originally proposed to reduce the communication cost in FL and is later proved to be also effective in defending against certain gradient leakage attacks [55]. Given a sparsification rate $p \in (0, 1)$, a binary mask M is first calculated by $\mathcal{M} \leftarrow \|\nabla w_{t,i}\| > p$ of $\|\nabla w_{t,i}\|$, then the mask is applied to the original model update according to $\varphi_s(\nabla w_{t,i}, p) = \mathcal{M} \odot \nabla w_{t,i}$, where $\odot$ denotes the point-wise multiplication operation.

(4) **Soteria** [44]: A recent solution to defend against data reconstruction attacks in FL. Soteria perturbs the representation of the input data that learned from a fully-connected layer L (called the defend layer) to maximize the reconstruction error. Suppose the global model of FL consists of a feature extractor before the defend layer and the classifier denoted as f_r. f_r first learns to map the target input data sample $x_{i,n} \in \mathbb{R}^d$ to a l-dimension representation $r \in \mathbb{R}^l$. Then, the classifier maps the learned representation r to the classes of the training task. Specifically, given a pruning rate $p \in (0, 1)$, the client first evaluates the impact ι_i of each element $r_i \in r$ by calculating $\iota_i = \|r_i(\nabla_{x_{i,n}} f_r(r_i))^{-1}\|_2$. Then, the client prunes the $p \times l$ elements in the defender layer to the largest value in $\iota_i, i \in [0, l-1]$ to get a perturbed representation r' of the input dataset D_i. Finally, the client computes and shares the update computed on the perturbed representation r'. Therefore, the defensive transformation function of Soteria can be considered as applying a mask $\mathcal{M}$ only to the update of the defend layer, which can be written as $\varphi_{sot}(\nabla w_{t,i}, p) = \mathcal{M} \odot \nabla w_{t,i}$.

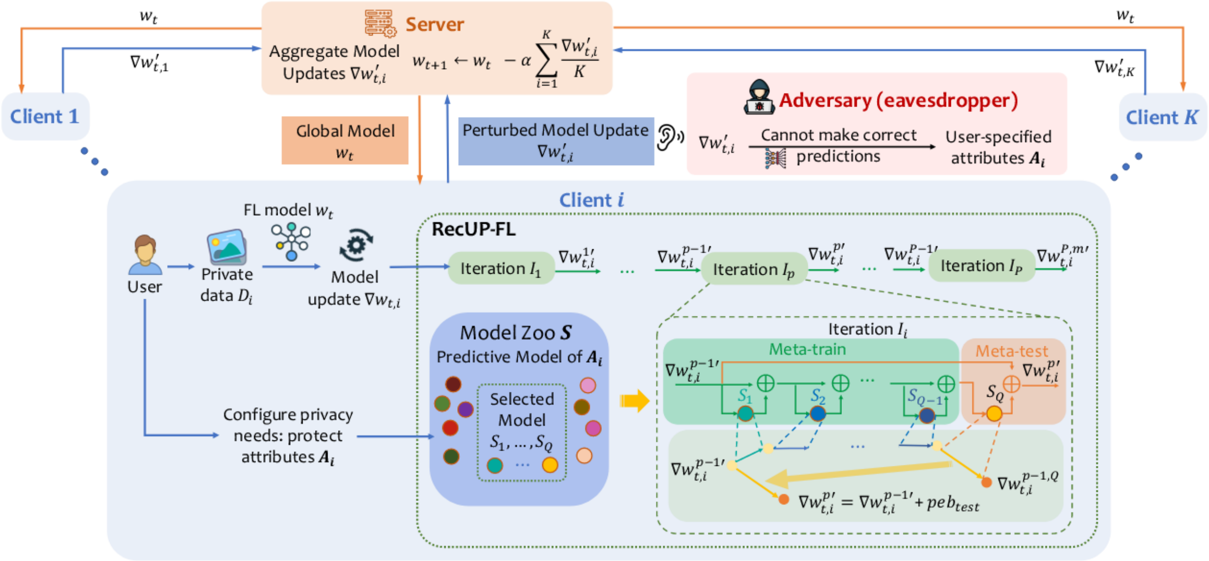

Figure 1: Overview of RecUP-FL.

While the above-mentioned client-level defenses can prevent privacy leakage in certain cases, the applied local transformation $\varphi(\cdot)$ on the model update will inevitably degrade its fidelity and thereby greatly impacts the utility of the resulting model. Additionally, we can observe from their transformation functions that all of them treat the whole model update as an entire entity and regard all attributes as equally important. However, sensitive attributes usually have unequal emphases at different layers/positions of the gradients [38], and more importantly, users may value different aspects of privacy differently [41]. Therefore, to meet actual privacy constraints, such a general treatment, which provides either redundant protection on non-sensitive gradients or insufficient protection on sensitive gradients, can hardly achieve an optimal utility-privacy trade-off.

3.2 Design Objectives

To address the limitations of existing efforts, we propose RecUP-FL, a user-configurable local privacy defense framework that seeks to reconcile the utility and privacy in FL. Unlike existing approaches, RecUP-FL can achieve a relaxed notion of privacy by providing effective protection only on the sensitive attributes specified by each user (i.e., client) according to their preference. This way the defense can be more flexible and can better adjust to different users' privacy requirements in practice while obtaining significant improvements in model utility. Specifically, RecUP-FL allows each client to specify a set of attributes $\mathbf{A}_i = \{a_{i,1}, ..., a_{i,M}\}$ to protect its local training data before the FL training. To protect the user-specified attributes, RecUP-FL will leverage the idea of adversarial learning to find the optimal (minimal) perturbation $p_{t,i}$ to be added to the original model update $\nabla w_{t,i}$ for the client C_i at the t-th communication round before sharing. This can be formally described as $\varphi_{\text{RecUP-FL}}(\nabla w_{t,i}) = \nabla w_{t,i} + p_{t,i}$. The objectives of RecUP-FL are twofold:

- **Objective I (Privacy)**: The perturbed model update should be able to prevent the adversary from inferring the user-specified sensitive attributes.
- **Objective II (Utility)**: RecUP-FL should find the minimal perturbation to maintain a good level of utility of the global model.

To achieve **Objective I**, we require the attributes inferred from the perturbed updates to be as different from the genuine attributes as possible:

$$\arg\max_{p_{t,i}} d(\mathcal{F}(p_{t,i} + \nabla w_{t,i}), \mathbf{A}_i), \tag{2}$$

where $d(\cdot)$ measures the distance between the inferred and genuine attributes and $\mathcal{F}(\cdot)$ denotes the adversarial prediction function, which takes model update as input and outputs the predicted sensitive attributes. That is, instead of attempting to degrade the reconstructed data quality, we only aim to prevent the updates from leaking information about the user-specified sensitive attributes.

To achieve **Objective II**, we seek to minimize the global model's training loss:

$$\arg\min_{p_{t,i}} \mathcal{L}(w_t + \alpha \sum_{i=1}^{N} (p_{t,i} + \nabla w_{t,i}),\ D_{test}), \tag{3}$$

where $\mathcal{L}(\cdot)$ is the training loss function (e.g., cross-entropy) and D_{test} is the evaluation set.

4 DESIGN OF RECUP-FL

4.1 System Overview

As shown in Figure 1, the proposed RecUP-FL can be implemented as an add-on defense module on the client side to prevent privacy leakage. To improve model utility while still meeting user's privacy requirements, RecUP-FL provides users with more control over their privacy and allows each client to configure their privacy needs before the FL training by identifying a set of sensitive attributes (i.e., $\mathbf{A}_i$ for the i-th client). In practical applications, the sensitivity and configuration of attributes only rely on users' personal preferences instead of any external regulations. For example, for most people, voice biometrics in general are often considered more private than emotion in speech data. They may only identify voice biometrics as a sensitive attribute to protect. This way we can achieve a relaxed notion of privacy that only needs to meet users' actual privacy needs. Specifically, in FL, at the communication round t, the i-th client needs to calculate the model update $\nabla w_{t,i}$ first through learning its private data with the FL model w_t. To protect the identified sensitive attributes while maintaining a good level of utility, we formulate the defense as an adversarial machine learning attack problem that seeks to find a minimal perturbation to be added to the model update before sharing to mislead the adversary model so that the adversary cannot reveal any private information from the perturbed update, i.e., $\nabla w'_{t,i}$. Note that RecUP-FL does not require the server to execute any additional tasks besides aggregation. We use FedAvg [35] in this work for simplicity, but in practice, RecUP-FL can also work with other secure aggregation rules, such as Krum [8], and Median [48], etc.

Although the client-level privacy defense can be formulated as launching an adversarial attack against the privacy-leakage adversary model, solving such an optimization problem to generate gradient perturbations is not trivial. Unlike existing adversarial attacks (mostly in white-box [10, 17] or black-box [25] settings), for the privacy defense purposes in FL, we need to consider a very restricted no-box setting as clients do not possess any knowledge about the configuration of the adversary model (e.g., model architecture and parameters) and, more importantly, they are not able to query the adversary model to counterfeit its functionality. Additionally, existing studies [27, 36] showed the feasibility of using non-neural-network approaches (e.g., Random Forest and Support Vector Machines) to infer sensitive attributes from model parameters. To make our gradient perturbation applicable to arbitrary adversary models regardless of their model type, architecture, and parameters, we need to improve the cross-model transferability of our generated defensive gradient perturbations.

Meta-learning [40] is proposed for solving unseen tasks by *learning to learn.* A meta-learning model first learns knowledge and seeks the inner connections from multiple training tasks (i.e., meta-train). Then later the model is adapted to the unseen task by fine-tuning with only few training samples (i.e., meta-test). A detailed introduction to meta-learning can be founded in Appendix A. Inspired by the meta-learning technique, we propose to use a two-step iterative method to generate perturbations for unseen and unquerable adversary models. As shown in Figure 1, at each iteration, we have: (1) *Meta-learn*: using a set of known defender models (i.e., **S**), as substitutes of the adversary model, to sequentially generate a *universal* adversarial perturbation (equivalent to launching white-box attacks); and (2) *Meta-test*: leveraging the prior knowledge in its learned universal perturbation to fine-tune itself to a new unseen defender model (equivalent to launching black-box attacks). By iteratively conducting white-box and black-box attacks, RecUP-FL can narrow the gap between the gradient directions in white-box and black-box attacks and gradually learn to adjust the perturbation from known defender models to the unknown adversary model. Further, we repeat the above two-step iterative method several times. The different compositions of substitute models each time make the generated perturbation will not bias to any specific model.

4.2 Methodology

Suppose RecUP-FL generates defensive perturbations to protect the model update $\nabla w_{t,i}$ calculated by the i-th client at the t-th communication round, and the client specifies its sensitive attribute set $\mathbf{A}_i = \{a_{i,1}, ..., a_{i,M}\}$. For simplicity, we first use the single attribute protection as an example to introduce our method and then expand it multi-attribute protection. The complete process of generating defensive perturbations can be found in Appendix Algorithm 1. A theoretical analysis can be found in Appendix Section I.

Single Attribute Protection. In this case, we consider $a_{i,m} \in \mathbf{A}_i$ as the targeted attribute to provide protection. Before the FL training, a model zoo, $\mathcal{S}$, consisting of multiple diverse pre-trained substitute models (referred to as defender models) needs to be created. These models all mimic the behavior of the adversary model, that is, predicting the $a_{i,m}$ through model updates. In practice, we can create an ensemble of models for every possible sensitive attribute prior to the FL training and pre-load the corresponding ensemble models to the clients according to their attribute specifications. To improve the transferability of the defensive perturbations to the unseen and un-queryable adversary model, we randomly sample $\{S_1, ..., S_Q\}$ from $\mathcal{S}$ and perform the following *meta-train* and *meta-test* to compute defensive gradient perturbations.

- **Meta-train.** Meta-train utilizes the first Q-1 models from the selected Q models to simulate white-box adversarial attacks to generate defensive perturbations. Considering clients usually only have limited computational resources, we adopt the Fast Gradient Sign Method (FGSM) [17] as our white-box attack method due to its fast execution and "free" adversarial training features compared to other computationally expensive attacks such as projected gradient decent (PGD)-based attacks [10].Specifically, FGSM directly utilizes the gradient information of the targeted model by modifying the benign input to the opposite direction of the correct prediction. The perturbation generation in FGSM can be described as:
$$\nabla w^q_{t,i} = \nabla w_{t,i} + \frac{\epsilon}{Q} \cdot sign(\nabla g_q(S_q(\nabla w_{t,i}), a_{i,m})), \tag{4}$$
where $\nabla w^q_{t,i}$ denotes the perturbed model update that aims to fool the defender model S_q, ϵ denotes the perturbation budget, and $g_q(\cdot)$ denotes the loss function of the model S_q. To generate a universal perturbation that can be applied to the first Q-1 defender models, we employ the iterative FGSM as:
$$\nabla w^q_{t,i} = \nabla w^{q-1}_{t,i} + \frac{\epsilon}{Q} \cdot sign(\nabla g_q(S_q(\nabla w^{q-1}_{t,i}), a_{i,m})), q \in [0, Q-1], \tag{5}$$
where $\nabla w^0_{t,i} = \nabla w_{t,i}$. After Q-1 iterations, the universal adversarial example (perturbed model update), $w^{Q-1}_{t,i}$, can be obtained.
- **Meta-test.** After gaining prior knowledge from the meta-train, the meta-test step is used to fine-tune the perturbation to make it adapt to the unseen model. In this step, we perform the black-box attack against the last sampled model S_Q to improve the generality of the perturbation obtained from the meta-train step. As we cannot access the model's loss function in the black-box setting, we adopt FGSM onto the cross-entropy ($\mathcal{L}_{\text{CE}}$) between the model predictions and ground-truths to generate perturbations, which can be formulated as:
$$\nabla w^Q_{t,i} = \nabla w^{Q-1}_{t,i} + \epsilon \cdot sign(\nabla \mathcal{L}_{\text{CE}}(S_Q(\nabla w^{Q-1}_{t,i}), a_{i,m})). \tag{6}$$
The perturbation generated by the meta-test step ($peb_{test} = \nabla w^Q_{t,i} - \nabla w^{Q-1}_{t,i}$) is the defensive perturbation we are seeking for. It relies on the prior knowledge from the meta-train (i.e., using the perturbed model update as the basis) and fine-tuning it to cover the unseen model in the meta-test. Then we can add it back

to the original model update $w_{t,i}$ as follows:

$$\nabla w'_{t,i} = \nabla w_{t,i} + (\nabla w^{Q}_{t,i} - \nabla w^{Q-1}_{t,i}) = \nabla w_{t,i} + peb_{test}. \quad (7)$$

- **Avoiding Bias.** We notice that only performing meta-train and meta-test for one time results in bias to some of the models, and the performance will be highly dependent on the one-time choice. Therefore, we propose to iteratively repeat the two-step process by composing different combinations of various models to improve the transferability further. Specifically, RecUP-FL takes the original model update $\nabla w_{t,i}$ as input, and repeats the meta-train/meta-test for P iterations in total. Each iteration takes the output of the last iteration as its input. It can be formulated as follows:

$$\nabla w^{p\prime}_{t,i} = \nabla w^{p-1\prime}_{t,i} + peb^{p}_{test}, p \in [0, P], \quad (8)$$

where $\nabla w^{0\prime}_{t,i} = \nabla w_{t,i}$ and $\nabla w^{P\prime}_{t,i}$ will be the final perturbation for the single attribute protection, by repeating the meta-train/meta-test steps, the final perturbation will not bias to any model and thus obtain a better transferability to mislead the unseen and unqueyable adversary model.

Multi-Attribute Protection. To expand RecUP-FL to protect more than one attribute simultaneously, we will first conduct single attribute protection for each attribute $a_{i,m} \in \mathbf{A}_i$ individually, and then combine all the attribute-specific defensive perturbations by taking into account their protection levels.

$$peb_{t,i} = \sum_{m=1}^{M} \gamma_m \cdot \nabla w^{P,m\prime}_{t,i}, \quad (9)$$

where $\gamma_m \in (0, 1)$ denotes the protection level on the m-th attribute given by the client's preference. Finally, $peb_{t,i}$ is the perturbation to be added on the model update $\nabla w_{t,i}$, which can effectively protect attributes $\mathbf{A}_i$.

5 EXPERIMENTAL SETUP

5.1 Datasets

We use four datasets to evaluate RecUP-FL: (1) *AudioMNIST* [7] contains 30,000 audio recordings of spoken English digits, (2) *Adult Income* [12] contains income records of 48,842 individuals, (3) *Labeled Faces in the Wild (LFW)* [24] contains 13,233 facial images from 5,749 people, each image is cropped to 62×47 pixels with RGB channels, and (4) *CelebFaces Attributes (CelebA)* [32] contains 202,599 RGB facial images of 32 × 32 pixels covering 10,177 identities.

Unless stated otherwise, we divide each dataset into D_{train} and D_{test} with 80% and 20% randomly selected samples, respectively. For the FL setting, by default, we assume each client possesses one sample in D_{train}. We will study the case where each client possesses multiple data samples in Section 6.4.

5.2 FL Model

We adopt different FL model architectures for the four datasets. More details of the architectures can be found in Appendix B.

(1) **AudioMNIST**. We use a CNN model to perform a 10-class classification of the spoken digits (i.e., zero to nine). We first process the audio into a spectrogram and then apply a model containing three convolutional layers, one hidden layer, and one output layer. The model is trained on the cross-entropy loss function with a learning rate of 10^{-4}.

(2) **Adult Income**. We use a 2-layer fully-connected neural network to perform binary classification on the income level (i.e., $> 50k$ or not). The model is trained on the Mean Squared Error (MSE) loss function with a learning rate of 0.01.

(3) **LFW**. We use a CNN model to perform binary emotion classification (i.e., smiling or not). The model consists of three convolutional layers, one hidden layer, and one output layer.

(4) **CelebA**. We adopt a CNN model to perform binary classification on gender. The model composes three convolutional layers, one hidden layer, and one output layer and is trained on the cross-entropy loss function with a learning rate of 0.01.

5.3 Defense Settings

Model Zoo Setting. To achieve good defense generalizability on various types of adversary models, the model zoo should include a sufficient number of defender models with diverse structures. Unless mentioned otherwise, in our implementation, we choose to construct the model zoo with 20 defender models for better transferability and utility-privacy trade-off. We also study the impact of model zoo size in Section 6.6. Our empirical study finds that using defender models with deeper structures (e.g., 4 or 5 layers) would not provide any significant performance benefit over shallow models. Therefore, to maintain a minimal model size and enable more computational efficiency on clients' local devices, we configure the defender models as 3-layer fully-connected neural networks with varying numbers of neurons for each layer (i.e., 128 to 2,048).

Parameter Setting. Unless otherwise stated, we empirically set the number of iterations, P, to 10 and the number of selected models for each iteration, Q, to 5 for all the datasets to improve the transferability of the defensive perturbations while maintaining a reasonable level of computational cost. Additionally, we explore different perturbation budget ranges for different datasets: $\epsilon \in [5 \times 10^{-5}, 0.5]$ for AudioMNIST dataset; $\epsilon \in [1 \times 10^{-5}, 0.1]$ for Adult Income dataset; $\epsilon \in [5 \times 10^{-5}, 0.5]$ for LFW dataset; and and $\epsilon \in [5 \times 10^{-6}, 0.5]$ for CelebA dataset. We study the impact of privacy budgets in Appendix D.

5.4 Adversary Models

Attribute Inference Attack. The adversary uses a pre-trained classifier to infer sensitive attributes from gradients. We train the adversary classifiers by the same training set as the defender models (i.e., D_{train}) to simulate the strongest adversary who has the knowledge of the defender models' training set. Following a prior study [36], we assume that the adversary applies max-pooling on the received gradients to reduce dimensionality. We explore the following adversary model settings:

(1) **Structure-known Neural Network (Stru-NN).** We assume the adversary model architecture is included in the defender model zoo (i.e., a 3-layer fully-connected neural network). Stru-NN is trained on the cross-entropy loss function using a Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 for 80 epochs.

(2) **Unknown Neural Network (Unkwn-NN).** We assume that the adversary model architecture is not included in the defender model zoo. It is a 4-layer fully-connected neural network, each layer contains 1,024, 1,024, 512, and 128 neurons, respectively. The training setting of Unkwn-NN is the same as Stru-NN.

(3) **Support Vector Machine (SVM).** We assume the adversary model is a Support Vector Machine classifier with an RBF kernel.

(4) **Random Forest (RF).** We assume the adversary model is a Random Forest (RF) classifier containing 120 trees.

Data Reconstruction Attack. Unlike attribute inference attacks, recent studies [49, 52, 55] showed that the adversary can even completely reconstruct private data by solving a gradient-matching problem. In this paper, we adopt the more advanced method proposed by Geiping *et al.* [15] given its capability in reconstructing high-resolution images and robustness against different random initializations. Specifically, the adversary will run 2,000 iterations to match gradients for the data reconstruction.

5.5 Defense Baselines

The following state-of-the-art defense baselines (described in Section 3.1) are used as baselines. To better compare their utility-privacy trade-offs, we also adjust their defense parameters (level of gradient transformations) in different ranges.

(1) **DP (Gaussian)** [46]. We set $\mu = 0$ for all datasets. We set $\sigma \in [5 \cdot 10^{-5}, 0.5]$, $\sigma \in [1 \cdot 10^{-5}, 0.1]$, $\sigma \in [5 \cdot 10^{-5}, 0.5]$, and $\sigma \in [5 \cdot 10^{-6}, 0.5]$ for AudioMNIST, Adult Income, LFW, and CelebA datasets, respectively. We set $B \in [20, 22]$, $B \in [20, 22]$, $B \in [1 \cdot 10^{-8}, 1 \cdot 10^{-3}]$, and $B \in [2 \cdot 10^{-6}, 0.1]$ for AudioMNIST, Adult Income, LFW, and CelebA datasets, respectively.

(2) **DP (Laplace)** [31]. We set $\mu = 0$ for all datasets. We set $b \in [2 \cdot 10^{-5}, 0.5]$, $b \in [1 \cdot 10^{-5}, 0.1]$, $b \in [2 \cdot 10^{-5}, 0.2]$ and $b \in [2 \cdot 10^{-6}, 0.1]$ for AudioMNIST, Adult Income, LFW, and CelebA datasets, respectively. The settings of B follow DP (Gaussian).

(3) **Gradient Sparsification** [30]. We set $p \in [20\%, 90\%]$, $p \in [15\%, 80\%]$, $p \in [80\%, 99\%]$, and $p \in [10\%, 90\%]$ for AudioMNIST, Adult Income, LFW, and CelebA datasets, respectively.

(4) **Soteria** [44]. We set $p \in [25\%, 95\%]$, $p \in [20\%, 90\%]$, $p \in [80\%, 95\%]$, and $p \in [50\%, 90\%]$, for AudioMNIST, Adult Income, LFW, and CelebA datasets, respectively.

5.6 Evaluation metrics

(1) **Learning Loss**: the loss value of the global model evaluated on D_{test}, used as the utility metric. A lower learning loss means better model performance.

(2) **Attack Success Rate (ASR)**: the ratio of correct predictions over the total number of attribute inferences performed by the adversary, used as the privacy metric. A lower ASR means better protection against the attribute inference attack.

(3) **Mean Square Error (MSE)**: the pixel-wise mean-square-error between the reconstructed image and the original image. A higher MSE means better protection against the data reconstruction attack.

6 EXPERIMENTAL RESULTS

6.1 Single Attribute Protection

To evaluate RecUP-FL against single-attribute inference attacks, we use three datasets and investigate three typical stages of the FL process, namely, the beginning, the middle, and the end stage of training. Specifically, according to their convergence speeds, for AudioMNIST, we consider the 1st, 3rd, and 5th rounds with identity (i.e., male native speaker, female native speaker, male non-native speaker and female non-native speaker) as the sensitive attribute; for Adult Income, we consider the 1st, 3rd, and 5th rounds with race (i.e., white, asian-pac-islander, amer-indian-eskimo, black and the other) as the sensitive attribute; and for LFW, we consider the 1st, 10th, and 50th rounds with race (i.e., asian, white and black) as the sensitive attribute.

Utility-Privacy Trade-off. For each defense, we tweak its parameter as mentioned in Section 5.5 to get a set of privacy and utility value pairs and plot the utility-privacy trade-off curve accordingly. It is worth noting that some curves such as gradient sparsification and Soteria are shorter than others. This is because their impacts on the gradients are limited by the parameters' range (e.g., the maximum sparsity is 100%).

(1) **AudioMNIST**. The results of the AudioMNIST dataset are shown in Figure 2. Since we aim at achieving lower ASR and learning loss, the left bottom corner is the optimal point. Our general observation is RecUP-FL achieves the best utility-privacy trade-off (i.e., closer to the optimal point). For example, when defending against the Stru-NN at the 1st round, RecUP-FL can achieve a low ASR (< 30%) without scarifying much utility while other baselines keep a relatively large ASR (i.e., around 70%). When defending against the Unkwn-NN at the 5th round, other baselines can provide sufficient protection only when increasing the perturbation budget to relatively large values, while RecUP-FL can reduce the ASR to around 25% by adding a much smaller perturbation.

(2) **Adult Income**. The results of the Adult Income dataset are shown in Appendix Figure 10. We observe that the best trade-off still happens in RecUP-FL. For instance, when defending against the Stru-NN at the 3rd round, achieving the same privacy protection (ASR around 35%), RecUP-FL can keep almost zero utility loss, but other baselines suffer a significant drop of learning loss. However, we notice that when defending against the RF, our defense performs similarly to the DP (Gaussian) and all baselines cannot get low ASR when adding small perturbation. One of the possible reasons is that RF is more robust against the added noise [26].

(3) **LFW**. The results of the LFW dataset are shown in Appendix Figure 11. Similarly, we can see that we still achieve the best trade-off. For example, in the case of Stru-NN at the 1st round, when achieving the same learning loss (around 0.53), RecUP-FL can largely reduce the ASR by 33.18% while other baselines can only reduce the ASR by 6.35%.

In summary, RecUP-FL achieves the best utility-privacy trade-off in the three datasets. This is because RecUP-FL only perturbs the neurons that carry more sensitive information about the specified attributes while preserving the neurons that are relevant to the training task. As a result, the generated perturbation would have less negative effects on the FL training while still being able to provide good protection on the sensitive attributes.

Transferability of RecUP-FL. Next, we examine the transferability of RecUP-FL by comparing the performance against different adversary models at the same round.

(1) **AudioMNIST**. As shown in Figure 2, we can see that Unkwn-NN are more difficult to defend than Stru-NN due to the unknown structure. RecUP-FL is still able to achieve an ASR that is on average 23.7% lower than other baselines. Also, we can observe that gradient sparsification can hardly defend against non-neural-network adversaries, especially at the 3rd and 5th rounds.

(2) **Adult Income**. As shown in Appendix Figure 10, RecUP-FL can still achieve the best utility-privacy trade-off regardless of the adversary model architecture. For example, when defending against Unkwn-NN, RecUP-FL achieves 29.79% ASR scarifying with almost

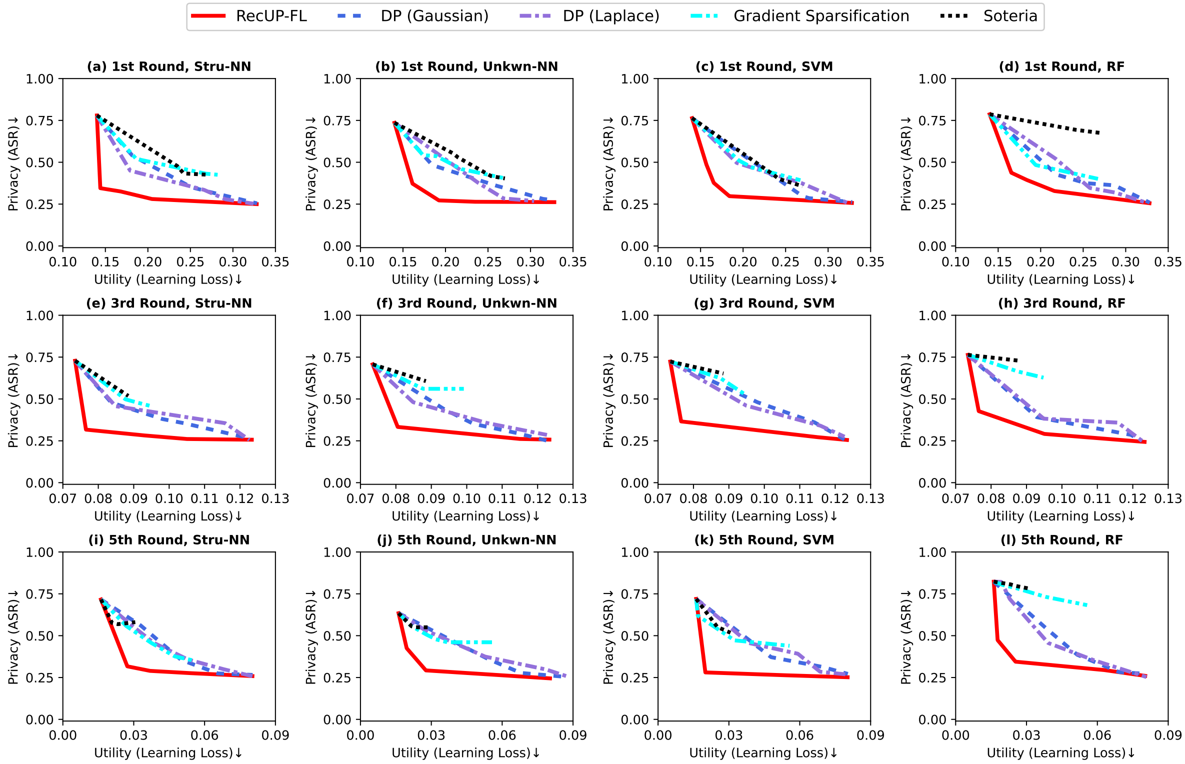

Figure 2: Utility-privacy trade-off curves on AudioMNIST (Some baselines have shorter trade-off curves due to the adjustable range limits of their parameters).

no learning loss at the 1st round. Also, we can observe that RecUP-FL can effectively defend the SVM adversary on the Adult Income dataset by lowering the ASR to around 28%, and other baselines can hardly defend them under similar learning loss.

(3) **LFW**. As shown in Appendix Figure 11, we can observe that RecUP-FL also achieves good transferability on the LFW dataset with the lowest ASR in defending against Unkwn-NN at all three stages. Even when defending against RF which has the best robustness against noise, RecUP-FL can still outperform other baselines.

To sum up, these results show that RecUP-FL has the capability in defending against both neural-network and non-neural-network adversary models. Because the various architectures in the model zoo have a strong ability to fit non-linear mapping functions from gradients to attributes. SVM and RF can also be considered as non-linear functions, and thus the generated perturbation can be utilized to defend against non-neural-network adversary models.

6.2 Multi-Attribute Protection

We further evaluate the effectiveness of RecUP-FL on the LFW dataset when the clients specify multiple sensitive attributes. Specifically, we assume the clients consider their race and age (i.e., baby, child, youth, middle-aged, senior) to be equally important (i.e., $\gamma_{gender} = \gamma_{age} = 0.5$). For the adversary model, we consider two separate Stru-NN adversary models that aim to predict race and age to evaluate their ASR respectively.

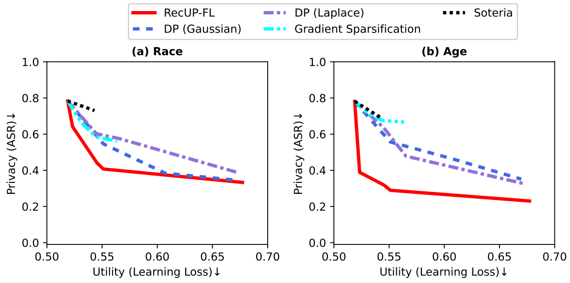

Figure 3: Multi-attribute protection on LFW.

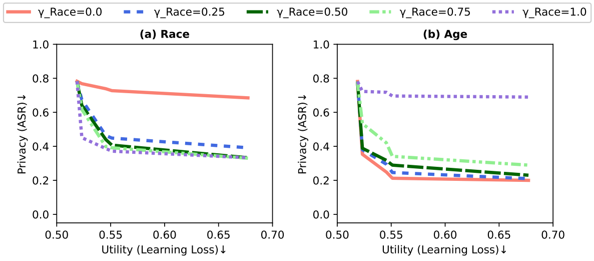

Figure 4: Multi-attribute protection with varying γ on LFW.

Utility-Privacy Trade-off. As shown in Figure 3, compared to other baselines, RecUP-FL can still provide a better level of protection on both attributes under the same utility budget. For example, in Figure 3(a), we can see that when the learning loss is 0.55, RecUP-FL can achieve an ASR that is 2 times lower than DP (Laplace). In addition, as shown in Figure 3(b), when the learning loss reaches 0.68, RecUP-FL can achieve around 0.2 ASR while other baselines cannot provide adequate protection on age (ASR > 30%).

Impact of γ. Following the above-mentioned setting, we set the γ_{race} in the range of $[0, 1]$ with a step 0.25 and $\gamma_{age} = 1 - \gamma_{race}$. As

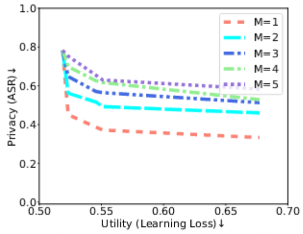

Figure 5: Multi-attribute protection with varying M on LFW.

shown in Figure 4(a), given the same utility budget, as we gradually increase γ_{race}, the ASR against race drops, and the ASR against age increases. This is because increasing γ_{race} results in more protection over the race attribute while making the age attribute more vulnerable. A similar trend can be observed on the age attribute in Figure 4(b). We also notice that even with the $\gamma_{race} = 0$, the ASR of the race inference attack would still drop slightly when a large perturbation is used. Therefore, even when RecUP-FL is set to focus on a particular attribute, it still has a certain degree of effect over other attributes due to the correlation between these attributes.

Impact of the Number of Sensitive Attributes. Next, we conduct experiments on the LFW dataset to study the impact of the number of sensitive attributes M on the performance of RecUP-FL. Specifically, we consider emotion (i.e., smiling or not) as the FL training task and investigate the following five sensitive attributes: (1) *Race*: asian, white and black, (2) *Gender*: male or female, (3) *Age*: baby, child, youth, middle-aged, senior, (4) *Glasses*: eyeglasses, sunglasses, no eyewear, and (5) *Hair*: black hair, blond hair, brown hair, bald. We assume that the clients treat every attribute equally (i.e., $\gamma = \frac{1}{M}$). Figure 5 shows the utility-privacy trade-offs of RecUP-FL against the Stru-NN adversary that aims to infer the client's race information at the 1st round of FL training. We can observe as we gradually increase the number of protected attributes, the utility-privacy trade-off is becoming worse. This is expected since given the same utility budget, protecting a smaller number of attributes allows each attribute to be assigned with a larger weight γ. Moreover, when trying to protect a large number (e.g., 5) of sensitive attributes, RecUP-FL still outperforms existing defenses that consider all private information as a single entity compared with the baselines in Appendix Figure 11. For instance, RecUP-FL can decrease the ASR to 0.65 even protecting five attributes simultaneously, while Soteria only reduces the ASR to around 0.75 when they both achieve 0.55 learning loss. These results verify that by selecting a few important sensitive attributes, RecUP-FL can achieve a relaxed form of privacy and thus provide a better utility-privacy trade-off. In some extreme cases, where users want to protect most (or even all) the possible attributes, they can also choose to use traditional privacy defenses to protect the entire data.

6.3 Defend against Data Reconstruction Attack

To evaluate the effectiveness of RecUP-FL against data reconstruction attacks, we further conduct experiments on the LFW and CelebA datasets. Specifically, for the LFW dataset, we consider the 1st, 10th, and 50th rounds with race as the sensitive attribute; and for the CelebA dataset, we consider the 1st, 5th, and 10th rounds with age as the sensitive attribute. For a fair comparison, we tweak the parameters of each defense to achieve a similar level of learning loss (within 10^{-3}).

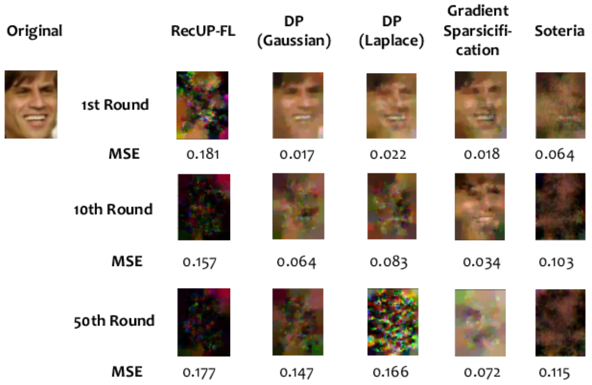

Figure 6: Defending against data reconstruction attack on LFW.

(1) **LFW**. The reconstructed images and the measured MSE on LFW dataset are shown in Figure 6. Some visual information can still be revealed from the reconstructions in some situations (e.g., gradient sparsification at the 1st and 10th rounds). However, with RecUP-FL, the reconstructed images do not show any information about specified attributes (i.e., gender). In addition, our defense outperforms other baselines in terms of the measured MSE. For example, RecUP-FL achieves an 8.2 times greater MSE compared with DP (Gaussian) at the 1st round, a 4.6 times greater MSE compared with gradient sparsification at the 10th round, and a 2.5 times greater MSE compared with gradient sparsification at the 50th round.

(2) **CelebA**. The reconstructions and MSE on the CelebA are shown in Appendix Figure 13. We observe that the facial structure can still be reconstructed after applying DP (Laplace) at the 1st and 5th rounds. In comparison, no useful information can be seen from the reconstruction results when RecUP-FL is applied. Specifically, RecUP-FL can achieve a 3.2 times greater MSE compared with gradient sparsification at the 1st round, a 2.1 times greater MSE compared with DP (Laplace) at the 5th round, and a 2.3 times greater MSE compared with gradient sparsification at the 10th round.

In summary, the results demonstrate RecUP-FL can effectively defend against data reconstruction attacks and achieve better utility-privacy trade-offs compared with other baselines.

6.4 Convergence Results

We examine the convergence of the resulting global model when applying RecUP-FL in a non-IID setting with the FedAvg aggregator. We distribute D_{train} of the LFW dataset to 100 clients with no overlap according to their identities, where each client maintains 81 images on average, and evaluate the learning loss of the trained model on D_{test}. We consider a practical situation where half of the clients select race as their specified attribute and the other clients select gender. The perturbation budget ϵ is set to 0.01, while the local training epoch and batch size are set to 1. We evaluate the impact of the participation ratio of clients (ranging from 0.2 to 1.0). The convergence results are presented in Figure 7. We can see that when RecUP-FL is applied, the model is still able to converge with various participation ratio. We observe that a lower ratio result in a slower convergence speed as expected because less training data is involved. In addition, we observe that when the ratio is greater than 0.8, the final learning loss is around 0.42, which is comparable with the case where no defense is applied.

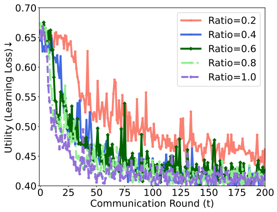

Figure 7: Convergence results on LFW.

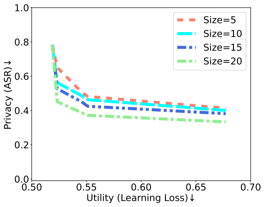

Figure 8: Impact of model zoo size.

Table 1: Time consumption on defenses.

Defense	RecUP-FL	DP (Gaussian)	DP (Laplace)	Gradient Sparsification	Soteria
Time (s)	7.22×10^{-2}	5.92×10^{-3}	1.88×10^{-1}	1.67×10^{-2}	5.18

6.5 Comparison with Different FGSM Variants

To verify the effectiveness of the proposed meta-learning-inspired method, we further conduct experiments to compare RecUP-FL with four versions of FGSM under Stru-NN and Unkwn-NN. We use the same set of parameters as in Section 5.3 and the details of each variant can be found in Appendix E. As shown in Appendix Figure 9, all variants can effectively defend against Stru-NN and Unkwn-NN, but RecUP-FL achieves the best trade-off among them. The one-step FGSM achieves the worst trade-offs in all situations as expected. For example, one-step FGSM only achieves around 0.53 ASR while momentum FGSM achieves 0.36 ASR when defending Stru-NNat the 10th round and both of them reach 0.45 learning loss. One possible reason is that perturbation generated by one-step FGSM only relies on one randomly selected model, which is hard to be coincidentally optimal. The iterative FGSM and average FGSM perform well, reaching similar ASR when defending Stru-NN at all three rounds (ASR differences are within 5%), as they fully utilized all the defender models to generate their perturbation.

6.6 Computational Resource Analysis

Consumed Time Comparison. We compare average consumed time to apply the defenses per communication round (i.e., *ClientUpdate()* in Algorithm 1 line 10 to 25) on LFW dataset using a Nvidia Quadro A100 GPU. As shown in Table 1, our defense achieves a comparable computation time with gradient sparsification and is much faster than Soteria. To evaluate the feasibility of implementing RecUP-FL on users' personal devices such as smartphones and laptops, we estimate the consumed training time via FLOPS (i.e., floating point operations per second). FLOPS measures the computational power of the given hardware and the ratio of FLOPS of two devices can be considered as the ratio of consumed time if they are running the same task [33]. The FLOPS of A100 GPU, popular chips used in smartphones (i.e., Apple A16 Bionic) and laptops (i.e., Intel Core I7 12700H) are 9.7×10^6 FLOPS [5], 2×10^6 FLOPS [3] and 1.69×10^6 [4] FLOPS, respectively. Thus the estimated consumed time of RecUP-FL on smartphones is 0.35 seconds and 0.41 seconds for laptops, which are still acceptable for edge users who typically have a small-sized local dataset. Although the training time of some baselines that do not rely on a model to generate the perturbation (e.g., gradient sparsification) is shorter than our defense, they fail to provide a good level of utility-privacy trade-offs.

Memory Usage Analysis. Memory consumption is also critical for the deployment clients' local devices. We further conduct experiments on the LFW dataset to investigate the impact of the model zoo size when defending against Unkwn-NN. Specifically, we vary the model zoo size from 5 to 20 with a step size of 5. When the size is 5, each iteration shares the same substitute models. As shown in Figure 8, we can observe that as the model zoo size increases, the trade-off is also improved due to the increased diversity in the model zoo. What's more, even with only five defender models in the zoo, RecUP-FL still outperforms others compared with Figure 11(b). Specifically, when achieving the same learning loss (around 0.55), RecUP-FL with only five defender models reduces the ASR to less than 0.5, while the ASR of DP (Gaussian) and DP (Laplace) is still higher than 0.6. In our Pytorch implementation, each model is ~45MB. Thus RecUP-FL requires ~900 MB even with a large model zoo size of 20, which is still acceptable on most modern devices. The required storage can be further reduced by using a smaller number of models.

7 CONCLUSION AND LIMITATIONS

Conclusion. We proposed RecUP-FL, the first user-configurable local privacy defense framework seeking to reconcile the utility and privacy in FL. By relaxing the notion of privacy, we focus on the user-specified attributes and thus obtain a significant improvement in model utility. Inspired by meta-learning, RecUP-FL finds a minimal defensive perturbation to add on the model update before sharing by iteratively conducting white-box and black-box attacks against a set of substitute adversary models. Extensive experiments on four datasets under both attribute inference attacks and data reconstruction attacks show that RecUP-FL can effectively meet user-specified privacy constraints while improving the model utility compared with four state-of-the-art privacy defenses.

Limitations and Future Work. Our system has the following limitations: 1) Additional Storage Space: As we rely on a set of defender models to generate defensive perturbations, additional storage space (~hundreds MB) is required on each device. If the device's storage space is not sufficient for the model zoo, we can leverage model compression/quantization techniques to reduce the size of each model; 2) Extra Computational Cost: As the computation of defensive perturbations relies on the proposed two-stage method, extra consumed time (~0.3 seconds on smartphones) is required. We can leverage the approximate computation to further accelerate RecUP-FL. For example, it is not necessary to get the accurate values of gradients in lines 17 and 20 of Algorithm 1 since only the sign of gradient values will be used; 3) Degraded Utility-Privacy Trade-off with a Large Number of Attributes: In some extreme cases, if users want to protect most (or even all) the possible attributes, we can directly apply traditional approaches instead. Our future research includes: 1) theoretically deriving certified robustness guarantee and convergence guarantee to FL; and 2) improving the fairness of FL model by reducing its dependency on unrelated attributes.

ACKNOWLEDGMENTS

This work is supported in part by NSF CNS-2114161, ECCS-2132106, CBET-2130643, the Science Alliance's StART program, and the GCP credits by Google Cloud.

REFERENCES

[1] 2016. General Data Protection Regulation. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.

[2] 2018. California Consumer Privacy Act. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.

[3] 2022. Apple A16 Bionic Chips Performance. https://www.cpu-monkey.com/en/cpu-apple_a16_bionic.

[4] 2022. Intel Core i7 12700H Core Performance. https://www.giznext.com/laptop-chipsets/intel-core-i7-12700h-chipset-gnt.

[5] 2022. NVIDIA A100 GPU. https://www.nvidia.com/en-us/data-center/a100/.

[6] Giuseppe Ateniese, Luigi Mancini, and Giovanni Felici. 2015. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks* 10, 3 (2015).

[7] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. 2018. Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. *CoRR* abs/1807.03418 (2018). arXiv:1807.03418

[8] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*. 119–129.

[9] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*.

[10] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. IEEE.

[11] Gábor Danner and Márk Jelasity. 2015. Fully distributed privacy preserving mini-batch gradient descent learning. In *IFIP International conference on distributed applications and interoperable systems*. Springer, 30–44.

[12] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/adult.

[13] Tiantian Feng, Hanieh Hashemi, Rajat Hebbar, Murali Annavaram, and Narayanan. 2021. Attribute Inference Attack of Speech Emotion Recognition in Federated Learning Settings. *arXiv preprint arXiv:2112.13416* (2021).

[14] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 619–633.

[15] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.

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

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of International Conference on Learning Representations (ICLR)*.

[18] Dhruv Guliani, Françoise Beaufays, and Giovanni Motta. 2021. Training speech recognition models with federated learning: A quality/cost framework. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3080–3084.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[20] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of ACM SIGSAC conference on computer and communications security*.

[21] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 9 (2021), 5149–5169.

[22] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. 2017. Context-aware generative adversarial privacy. *Entropy* 19, 12 (2017), 656.

[23] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. 2018. Generative adversarial privacy. *arXiv preprint arXiv:1807.05306* (2018).

[24] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. 2007. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Technical Report 07-49. University of Massachusetts, Amherst.

[25] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*. PMLR, 2137–2146.

[26] Shotaro Ishii and David Ljunggren. 2021. A Comparative Analysis of Robustness to Noise in Machine Learning Classifiers.

[27] Jinyuan Jia and Neil Zhenqiang Gong. 2018. {AttriGuard}: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning. In *27th USENIX Security Symposium (USENIX Security 18)*. 513–529.

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

[29] John Koetsier. 2021. Apple iOS 15 And Healthtech: This Is The Next Generation Of Health. https://www.forbes.com/sites/johnkoetsier/2021/06/07/apple-ios-15-and-healthtech-this-is-the-next-generation-of-health/?sh=409c97e74730.

[30] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887* (2017).

[31] Xiaoyuan Liu, Hongwei Li, and Miao He. 2020. Adaptive privacy-preserving federated learning. *Peer-to-Peer Networking and Applications* 13, 6 (2020).

[32] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*. 3730–3738.

[33] Yao Lu, Guangming Lu, Jinxing Li, and Yuanrong Xu. 2021. Fully shared convolutional neural networks. *Neural Computing and Applications* 33, 14 (2021).

[34] Lingjuan Lyu and Chen Chen. 2021. A novel attribute reconstruction attack in federated learning. *arXiv preprint arXiv:2108.06910* (2021).

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

[36] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 691–706.

[37] Mitek. 2021. How does machine learning help with fraud detection in banks? https://www.miteksystems.com/blog/how-does-machine-learning-help-with-fraud-detection-in-banks.

[38] Fan Mo, Anastasia Borovykh, Mohammad Malekzadeh, Hamed Haddadi, and Soteris Demetriou. 2021. Quantifying and Localizing Private Information Leakage from Neural Network Gradients. *arXiv preprint arXiv:2105.13929* (2021).

[39] Payman Mohassel and Yupeng Zhang. 2017. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 19–38.

[40] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[41] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2017. Towards a visual privacy advisor: Understanding and predicting privacy risks in images. In *Proceedings of the IEEE international conference on computer vision*. 3686–3695.

[42] Nisarg Raval, Ashwin Machanavajjhala, and Jerry Pan. 2019. Olympus: Sensor Privacy through Utility Aware Obfuscation. *Proc. Priv. Enhancing Technol.* 2019, 1 (2019).

[43] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[44] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9311–9319.

[45] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE Conference on Computer Communications*. IEEE.

[46] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. 2021. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 797–807.

[47] Xingxing Xiong, Shubo Liu, Dan Li, and Xiaoguang Niu. 2020. A comprehensive survey on local differential privacy. *Security and Communication Networks* (2020).

[48] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498* (2018).

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

[50] Zheng Yuan, Jie Zhang, Yunpei Jia, Chuanqi Tan, Tao Xue, and Shiguang Shan. 2021. Meta gradient adversarial attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*. 7748–7757.

[51] Oualid Zari, Chuan Xu, and Giovanni Neglia. 2021. Efficient passive membership inference attack in federated learning. *arXiv preprint arXiv:2111.00430* (2021).

[52] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610* (2020).

[53] Han Zhao, Jianfeng Chi, Yuan Tian, and Geoffrey J Gordon. 2019. Adversarial privacy preservation under attribute inference attack. (2019).

[54] Haitian Zheng, Kefei Wu, Jong-Hwi Park, Wei Zhu, and Jiebo Luo. 2021. Personalized fashion recommendation from personal social media data: An item-to-set metric learning approach. In *2021 IEEE International Conference on Big Data*.

[55] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems* 32 (2019).

A INTRODUCTION OF META-LEARNING

In traditional machine learning, we decide on a learning algorithm by hand for the desired task and train the model from scratch. However, when the data is expensive or hard to obtain, or computational resources are unavailable, the performance of the traditional scheme will be limited. Meta-learning targets to replace prior hand-designed learners with learned learning algorithms [21]. Such a scheme is also called '*learning to learn*'. Many definitions and perspectives on meta-learning can be founded in the existing literature. The goal of meta-learning is to learn a model initialization such that it can be quickly adapted to new tasks using limited training samples. Inspired by the intuition of meta-learning, which utilizes prior knowledge learned from a wide distribution of models and adopts it to new tasks, we generate perturbations for the unseen and uncurable adversary models by the proposed two-step iterative method as described in Section 4.2.

B FL MODEL ARCHITECTURES

The detailed FL model architectures of four datasets are shown in Appendix Table 2.

Table 2: FL Models Architecture for four datasets.

Layer Type	Parameters
Input	224 × 224
Convolution	16 × 3 × 3, strides=(2,2)
BatchNorm.	16
Activation	ReLU
Pooling	MaxPooling(2 × 2)
Convolution	32 × 3 × 3, strides=(2,2)
BatchNorm.	32
Activation	ReLU
Pooling	MaxPooling(2 × 2)
Convolution	64 × 3 × 3, strides=(2,2)
BatchNorm.	32
Activation	ReLU
Pooling	MaxPooling(2 × 2)
Flatten	
Fully Connected	32
Activation	ReLU
Fully Connected	10

(a) AudioMNIST

Layer Type	Parameters
Input	1 × 103
Fully Connected	50
Activation	ReLU
Fully Connected	1
Activation	Sigmoid

(b) Adult Income

Layer Type	Parameters
Input	62 × 47
Convolution	32 × 3 × 3, strides=(1,1)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Convolution	64 × 3 × 3, strides=(1,1)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Convolution	128 × 3 × 3, strides=(1,1)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Flatten	
Fully Connected	256
Activation	ReLU
Fully Connected	2
Activation	Sigmoid

(c) LFW

Layer Type	Parameters
Input	32 × 32
Convolution	16 × 3 × 3, strides=(2,2)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Convolution	32 × 3 × 3, strides=(2,2)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Convolution	64 × 3 × 3, strides=(2,2)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Convolution	128 × 3 × 3, strides=(2,2)
Pooling	MaxPooling(2 × 2)
Activation	ReLU
Dropout	0.2
Flatten	
Fully Connected	256
Fully Connected	128
Fully Connected	2
Activation	Sigmoid

(d) CelebA

C UTILITY-PRIVACY TRADE-OFF CURVES

The utility-privacy trade-offs evaluated on the Adult Income and LFW dataset are shown in Appendix Figure 10 and Figure 11, respectively.

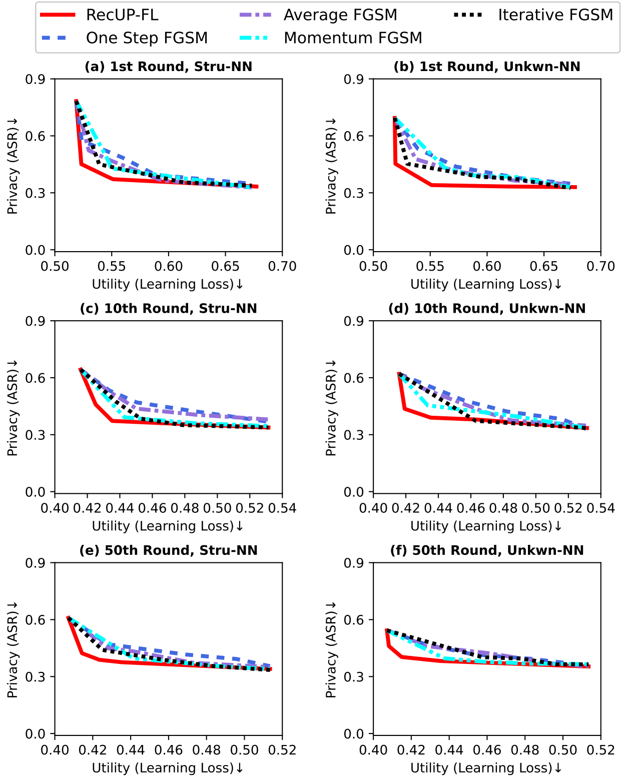

Figure 9: Comparison with four FGSM variants on LFW.

D ANALYSIS OF PRIVACY BUDGETS

The utility of global model when applying different privacy budgets of RecUP-FL (i.e., ϵ) at three training stages on LFW dataset are shown in Appendix Figure 12. We can obviously observe that when the budget increases, the learning loss increases since a larger perturbation is added to the model update.

E IMPLEMENTATION OF FGSM VARIANTS

The computation of each FGSM variant is shown as follows:
(1) **One Step FGSM.** We only randomly select one substitute model S_q from model zoo $\mathcal{S}$ and performs FGSM once to get the perturbation as follows:

$$\nabla w'_{t,i} = \nabla w_{t,i} + \epsilon \cdot sign(\nabla g_q(S_q(\nabla w_{t,i}), a_{i,m})). \tag{10}$$

(2) **Average FGSM.** We randomly select Q substitute models from model zoo $\mathcal{S}$, performs FGSM separately and averages the perturbations as follows:

$$\nabla w'_{t,i} = \nabla w_{t,i} + \frac{1}{Q}\sum_{q=1}^{Q} \epsilon \cdot sign(\nabla g_q(S_q(\nabla w_{t,i}), a_{i,m})). \tag{11}$$

(3) **Iterative FGSM.** We randomly select Q substitute models from model zoo $\mathcal{S}$, performs FGSM iteratively and accumulates the perturbations as follows:

$$\nabla w^q_{t,i} = w^{q-1}_{t,i} + \frac{\epsilon}{Q} \cdot sign(\nabla g_q(S_q(\nabla w^{q-1}_{t,i}), a_{i,m})). \tag{12}$$

(4) **Momentum FGSM.** We randomly select Q substitute models from model zoo $\mathcal{S}$, performs FGSM iteratively and accumulates the

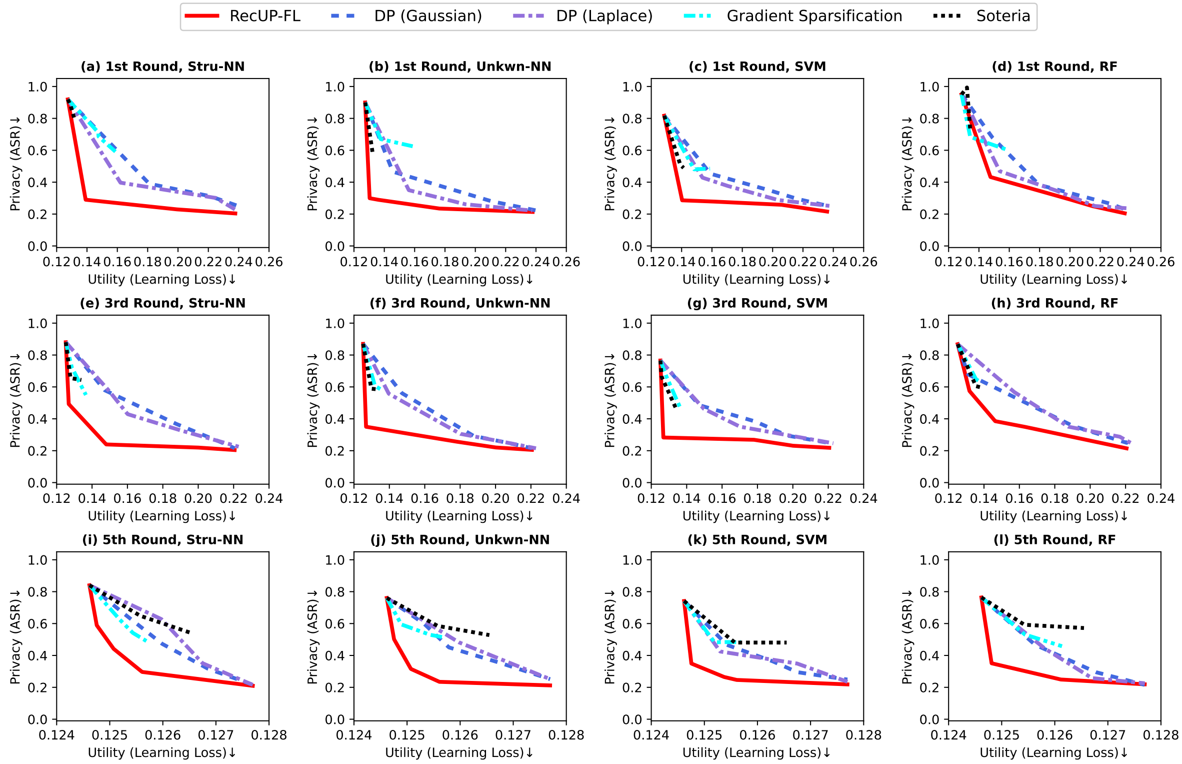

Figure 10: Utility-privacy trade-off curves on Adult Income (Some baselines have shorter trade-off curves due to the adjustable range limits of their parameters).

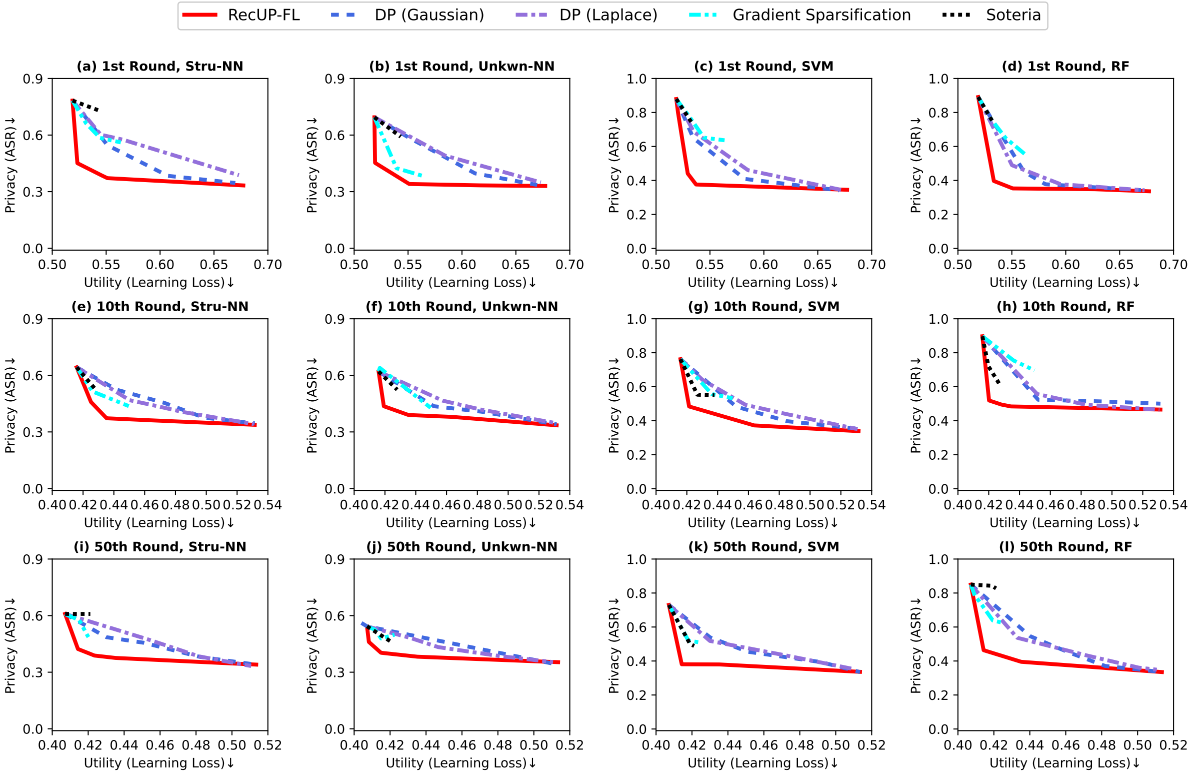

Figure 11: Utility-privacy trade-off curves on LFW (Some baselines have shorter trade-off curves due to the adjustable range limits of their parameters).

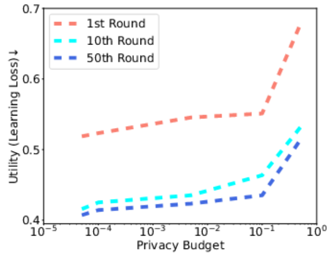

Figure 12: Utilities with varying privacy budgets on LFW.

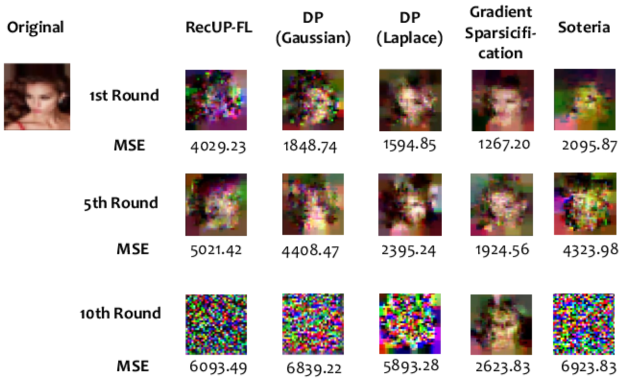

Figure 13: Defending against data reconstruction attack on CelebA.

perturbations with a momentum factor $\mu = 0.9$ to constrain the direction of the perturbation as follows:

$$u_q = \mu \cdot u_{q-1} + \frac{g_q(S_q(\nabla w_{t,i}^{q-1}), a_{i,m}))}{\|\nabla g_q(S_q(\nabla w_{t,i}^{q-1}), a_{i,m}))\|_1}, \tag{13}$$

$$\nabla w_{t,i}^{q} = w_{t,i}^{q-1} + \frac{\epsilon}{Q} \cdot sign(u_q). \tag{14}$$

F COMPARISON WITH OTHER FGSM VARIANTS

The utility-privacy trade-offs of other FGSM variants and our defense on the LFW dataset are shown in Appendix Figure 9.

G DEFEND AGAINST DATA RECONSTRUCTION ATTACK

The reconstructed images and measured MSE values on the CelebA dataset are shown in Appendix Figure 13.

H ALGORITHM

The complete process of generating defensive gradient perturbations can be found in Appendix Algorithm 1.

Algorithm 1 RecUP-FL in FL Pipeline

```
Input: Learning rate $\alpha$, number of communication rounds $T$
Output: Trained global model $\mathcal{G}$ with weight $w_T$
Server:
Initialize global model $\mathcal{G}$ with weight $w_1$
for each communication round $t \in [1, T]$ do
    Randomly select $K$ clients from the entire population
    Send global model with weights $w_t$ to $K$ clients
    $\nabla w_{t,i}' = ClientUpdate(w_t)$ from the $i$-th client, $i \in [1, K]$
    Wait for model updates from $K$ clients
    Aggregate model updates using FedAvg: $w_{t+1} \leftarrow w_t - \alpha \frac{1}{K} \sum_{i=1}^{K} \nabla w_{t,i}'$
end for
ClientUpdate($w_t$):
Initialize $peb_{t,i} = 0$
Train model on client's local data $D_i$ and calculate the overall model update by $\nabla w_{t,i} = \frac{\partial \mathcal{L}(D_i, w_t)}{\partial w_t}$
for each configured attribute $a_{i,m} \in \mathbf{A}_i$ do
    for iteration $p$=1, ..., $P$ do
        Randomly select a subset of models $\{S_1, ..., S_Q\}$ from the pre-loaded model zoo $\mathcal{S}$
        Meta-train:
        for $q$=1,2,...,$Q$-1 do
            $\nabla w_{t,i}^{q} = \nabla w_{t,i}^{q-1} + \frac{\epsilon}{Q} \cdot sign(\nabla g_q(S_q(\nabla w_{t,i}^{q-1}), a_{i,m}))$
        end for
        Meta-test:
        $\nabla w_{t,i}^{Q} = \nabla w_{t,i}^{Q-1} + \epsilon \cdot sign(\nabla \mathcal{L}_{\mathrm{CE}}(S_Q(\nabla w_{t,i}^{Q-1}), a_{i,m}))$
        $\nabla w_{t,i}^{p\prime} = \nabla w_{t,i}^{p-1\prime} + (\nabla w_{t,i}^{Q} - \nabla w_{t,i}^{Q-1})$
    end for
    $peb_{t,i} = peb_{t,i} + \gamma_m \cdot (w_{t,i}^{P\prime} - \nabla w_{t,i})$
end for
return $\nabla w_{t,i} + peb_{t,i}$
```

I THEORETICAL ANALYSIS

To theoretically show the reason why the perturbation's transferability can be greatly improved through these meta-train and meta-test steps, we consider one iteration and single attribute protection as an example for simplicity. Let peb_{train} denotes the final perturbation generated by the meta-train step, then the objective function of meta-test can be written as:

$$\underset{peb_{test}}{\arg\max}\, \mathcal{L}_{\mathrm{CE}}(S_Q(\nabla w_{t,i} + peb_{train} + peb_{test}), a_{i,m}). \tag{15}$$

It means that meta-test tries to find a perturbation for $\nabla w_{t,i}$ on the basis of the meta-train step to maximize the estimated cross-entropy loss function. According to the Tayler first-order expansion rule, we can expand the Equation 15 to the following equation:

$$\begin{aligned} &\underset{peb_{test}}{\arg\max}\, \mathcal{L}_{\mathrm{CE}}(S_Q(\nabla w_{t,i} + peb_{test}), a_{i,m}) + \\ &\qquad peb_{train} \cdot \nabla \mathcal{L}_{\mathrm{CE}}(S_Q(\nabla w_{t,i} + peb_{test}), a_{i,m}). \end{aligned} \tag{16}$$

To maximize the above objective function, the first term can be considered as the objective function of the meta-test step, which is to mislead the defender model S_Q. The second term can be considered as constraining the gradient directions of peb_{train} and peb_{test} to be as similar as possible. In other words, the objective function forces meta-test to generate the most similar perturbation as meta-train. It indirectly requires meta-test to utilize the prior knowledge from meta-train and adapt the perturbation to the new task, which is consistent with our design.