A Novel Attribute Reconstruction Attack in Federated Learning

Lingjuan Lyu[1] , **Chen Chen**[2]
[1] Sony AI [2] Zhejiang University
lingjuanlvsmile@gmail.com, cc33@zju.edu.cn

Abstract

Federated learning (FL) emerged as a promising learning paradigm to enable a multitude of participants to construct a joint ML model without exposing their private training data. Existing FL designs have been shown to exhibit vulnerabilities which can be exploited by adversaries both within and outside of the system to compromise data privacy. However, most current works conduct attacks by leveraging gradients on a small batch of data, which is less practical in FL. In this work, we consider a more practical and interesting scenario in which participants share their epoch-averaged gradients (share gradients after at least 1 epoch of local training) rather than per-example or small batch-averaged gradients as in previous works. We perform the first systematic evaluation of attribute reconstruction attack (ARA) launched by the malicious server in the FL system, and empirically demonstrate that the shared epoch-averaged local model gradients can reveal sensitive attributes of local training data of any victim participant. To achieve this goal, we develop a more effective and efficient gradient matching based method called cos-matching to reconstruct the training data attributes. We evaluate our attacks on a variety of real-world datasets, scenarios, assumptions. Our experiments show that our proposed method achieves better attribute attack performance than most existing baselines.

1 Introduction

Federated learning (FL) allows multiple participants to collaboratively construct a better global model [McMahan *et al.*, 2017]. In FL, each participant trains a local model on its own data and periodically shares model parameters or updates with a parameter server. Since training data never leave the participants, FL provides a privacy-aware solution for scenarios where data is sensitive (e.g., biomedical records, private images, personal text and speech, and personally identifiable information like location, purchase etc.). However, recent works have pointed out that FL may not provide sufficient privacy guarantees [Kairouz *et al.*, 2019; Lyu *et al.*, 2020b; Lyu *et al.*, 2020a], as communicating model updates throughout the training process can nonetheless leak private training data information, from shallow [Melis *et al.*, 2019; Hitaj *et al.*, 2017] to deep leakage [Zhu *et al.*, 2019; Zhao *et al.*, 2020].

For shallow leakage from gradients, [Hitaj *et al.*, 2017] used GANs [Goodfellow *et al.*, 2014] to synthesize images that look similar to the training data from the gradient. However, this attack only works when all class members look alike (e.g., face recognition). [Melis *et al.*, 2019] developed learning-based methods to demonstrate that these exchanged updates can leak unintended information about participants' training data. It should be noted that all these attacks require true class labels.

For deep leakage from gradients, [Zhu *et al.*, 2019] demonstrated *Deep Leakage from Gradients* (DLG) and proposed an optimization algorithm that can obtain both the training inputs and the labels in few iterations. A follow-up work called *Improved Deep Leakage from Gradients* (iDLG) [Zhao *et al.*, 2020] pointed out that DLG has difficulty in convergence and discovering the ground-truth labels consistently. However, there are several inherent weaknesses in both DLG and iDLG that may limit their applicabilities in FL, including: (1) both works adopted a second-order optimization method called L-BFGS, which is more computationally expensive compared with the first-order optimization we adopted; (2) both works are only applicable to the gradient computed on a small batch of samples, i.e., at most B=8 in DLG, and B=1 in iDLG, which are less practical in FL, in which gradient is normally shared after at least 1 epoch of local training on each participant's local data for communication efficiency [McMahan *et al.*, 2017]; (3) both works used untrained model (pre1 in our work), neglecting gradients over multiple communication rounds.

In addition to the above risks in existing literatures, in this work, we further ask the question: is it possible to steal other private information from epoch-averaged gradients instead of small batch-averaged gradients? To answer this question, we initiate a new attack called *attribute reconstruction attack* (ARA), i.e., we aim to accurately reconstruct the sensitive attributes of the training examples of the victim participant by exploiting the privacy vulnerabilities of the released local model gradients.

In summary, the following main contributions are made:

- We initiate a more practical and interesting attack against FL system, called *attribute reconstruction attack* (ARA), in which participants share their gradients/updates after

at least 1 epoch of local training instead of small batches.

- We propose a more effective and efficient gradient matching based method called cos-matching, which utilizes the cosine similarity between the true update from the victim participant and the virtual update.
- We conduct a comprehensive analysis and investigate various practical settings, including both the isolated and non-isolated settings, and the adversary with a range of prior knowledge (attribute distribution, membership and true label). Extensive experiments validate the effectiveness of our method.

2 Problem Definition

Attribute inference or reconstruction attack aims to determine the value of a particular attribute given that the adversary knows all the other non-sensitive attributes of one record and has access to either a trained model [Melis *et al.*, 2019] or model embedding [Song and Raghunathan, 2020]. However, none of the previous works systematically investigate attribute reconstruction attack (ARA) in FL. To fill in this gap, we establish attribute reconstruction attack in FL, which targets on the sensitive attributes of the training samples in any victim participant's data set, as illustrated in Fig 1.

The intuition behind ARA is that the observations of local model gradients can be used to infer sensitive attributes, which are in turn based on the participants' private training data. The snapshots of the gradients across communication rounds can also be exploited to generate different views of the target participant's local training data, thus inferring the *exact* value of the sensitive attributes. We remark that our objective of reconstructing the exact value of x_s is different from model inversion attack [Yeom *et al.*, 2018], in which the attacker aims to learn the private attribute distribution $p(x_s|y, x_{ns})$ given all the non-sensitive attributes x_{ns} and the true label y. Our attack assumption is more practical and reasonable in FL setting. We did not assume that the adversary has access to the joint distribution of the private feature and label $p(X_s, X_{ns}, Y)$ as in [Yeom *et al.*, 2018]. More importantly, we suppose **each participant will send local model updates after 1 epoch of local training on its whole data**, rather than updates on a small batch of data as in previous works [Zhu *et al.*, 2019; Zhao *et al.*, 2020]. Moreover, the attacker may not know the membership of the target record as a prior knowledge.

3 Attack Methodologies

In FL, both the server and participants may be malicious, and compromise the FL system. In particular, we focus on the malicious server who aims to infer the sensitive attributes (usually discrete) [Song and Raghunathan, 2020], which are normally binary or categorical demographic related attributes. This malicious server can use the target participant's model updates revealed in each round of the collaborative training process and adaptively update the reconstructed attribute.

Concretely, each record of interest can have K candidate records by enumerating the unknown sensitive attribute value from the corresponding attribute value range of size K. For any unknown attribute x_s within a limited range (categorical

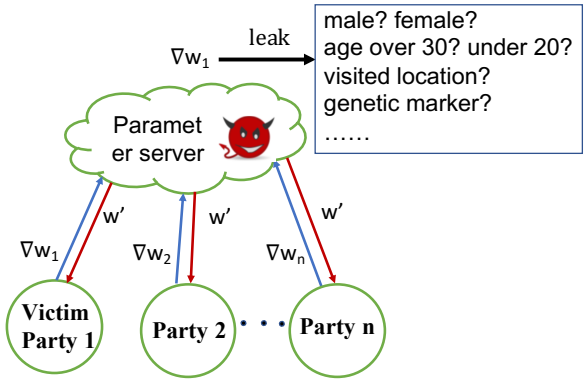

Figure 1: Attribute reconstruction attack (ARA) in federated learning. The malicious server can reconstruct victim participant's attributes from its local model updates.

or discretizing each numeric attribute into bins of equal width), the attacker can enumerate the target attribute x_s with all the possible values (with size K) for each record $\boldsymbol{x}$, resulting in K candidate records/guess points.

3.1 Attacker types and prior knowledge

In this work, we consider both the active attacker and the passive attacker. The active attacker can isolate the target participant, and create a local view of the model for it. In this scenario, the attacker can isolate the target participant and segregate the target participant's learning process [Nasr *et al.*, 2019]. By contrast, the passive attacker strictly follows the FL protocol, and sends the aggregated parameters of all participants to the victim participant, but tries to reconstruct the sensitive attributes of the victim participant's training records.

In real scenarios, the attacker may have no prior knowledge about the membership before conducting ARA. We next explore several practical scenarios based on the attacker's prior knowledge about the membership.

3.2 When membership is known

Given membership information (the true label Y is known), the attacker can directly reconstruct any attribute by using our cos-matching method as shown in Algorithm 1. Here cosine similarity is used to measure the angular distance between the true update ∇w_t from the victim participant and the virtual update $\nabla w'_t$. The objective function is formulated as:

$$J = cosinesim(\nabla w'_t, \nabla w_t) \tag{1}$$

We denote the local data of the victim participant as: $\mathbf{X} = [X_{ns}, X_s]$, where $X_{ns} = [X[:, : i-1], X[:, i+1 :]]$ represents the non-sensitive attributes, while $X_s = X[:, i]$ represents the sensitive/target attribute at i-th index. Based on the prior knowledge of the attribute of interest, we consider two cases to initialize the sensitive attribute: 1) if the prior distribution is known, then we initialize the virtual attribute from the prior distribution p of the target attribute over K potential values; 2) if the prior distribution is unknown, then we initialize the virtual attribute from a random distribution $\mathcal{N}(0, 1)$. We then generate the multinomial distribution for the sensitive attribute,

Algorithm 1 Attribute reconstruction based on cos-matching

1: **Input:** Victim participant records $(\mathbf{X}, \boldsymbol{Y})$ of size $(N, d+1)$ and $(N, 1)$. The i-th column of $\mathbf{X} = \{X[:, : i-1], X_s, X[:, i+1 :]\}$ is unknown (corresponding to the sensitive attribute X_s); Each element of X_s falls into $\{1, \cdots, K\}$; Victim participant target model $f(w_t)$ and loss function ℓ; Gradient ∇w_t of f computed w.r.t. $\mathbf{X}$; Gumbel softmax parameter γ; attribute prior.
2: **Output:** Predicted sensitive attribute X_s.
3: **Execute:**
4: **if** prior==known **then**
5: $X_s \leftarrow log(prior)$
6: **else**
7: $X_s \leftarrow \mathcal{N}(0, 1)$
8: **end if**
9: Generate multinomial distribution for the sensitive attribute X_s in $\mathbf{X}$, by computing the gumbel softmax of X_s, i.e., Softmax(X_s/γ,dim=1).
10: Compute the predicted attribute value $a = \sum_{n=1}^{K} n[Softmax(X_s/\gamma, dim = 1)]_n$.
11: Concatenate the predicted attribute with other non-sensitive attributes to get virtual input $X' = Concat(X[:, : i-1], a, X[:, i+1 :])$.
12: Given the true gradients for T epochs, the attacker solves the following optimization problem w.r.t. X_s (refer to Equation 3, Equation 4):

$$X_s^* = \underset{\mathrm{X_s}}{\operatorname{argmax}} \sum_{t=1}^{T} \operatorname{cosinesim}\left(\frac{\partial \ell(X', Y; w_t)}{\partial w_t}, \nabla w_t\right)$$

13: **return** predicted attribute X_s

and compute the predicted attribute value as follows:

$$a = \sum_{n=1}^{K} n[Softmax(X_s/\gamma, dim = 1)]_n \tag{2}$$

We adopt Gumbel-softmax [Jang *et al.*, 2016] to compute the multinomial distribution parameter over all the potential attribute values (line 9 Algorithm 1), where γ is the Gumbel softmax parameter. Afterwards, we concatenate the predicted attribute with other non-sensitive attributes to get "virtual inputs" $X' = Concat(X[:, : i-1], a, X[:, i+1 :])$, then feed these "virtual inputs" into the victim participant's model and get the "virtual gradients".

$$\nabla w_t' = \frac{\partial \ell(X', Y; w_t)}{\partial w_t} \tag{3}$$

Matching the virtual gradients with the received gradients from the victim participant in order to update the target attribute can be cast as an optimization problem. When the virtual gradients become closer to the original gradients, virtual attribute also becomes closer to the real training data attribute. Given gradients at a certain iteration t, we obtain the training data attribute by maximizing the following objective:

$$X_s^* = \underset{\mathrm{X_s}}{\operatorname{argmax}} \operatorname{cosinesim}(\nabla w_t', \nabla w_t) \tag{4}$$

The objective function *cosinesim* is differentiable w.r.t the target attribute X_s, and thus can be optimized using the standard gradient-based methods. Note that our proposed cos-matching method is also applicable to the scenario when the true label Y is unknown, the only modification to Algorithm 1 is to initialize the virtual label from either the label prior or a random distribution.

3.3 When membership is unknown

If the membership information is unknown, we first need to determine the membership of the target data points by launching a membership inference attack (MIA). To achieve this goal, we propose to apply a *Gaussian Mixture Model* (GMM) with two components (Member and Non-member) on the last-layer gradient variances of all the candidate records (generated by enumerating the unknown sensitive attribute from the attribute value range) to distinguish members from non-members. Our design rationale include: (1) we found that the gradient distribution for the member and non-member instances are distinguishable. In particular, the gradient variances of all the candidate training examples (Member) are all prone to 0, while the gradient variances of all the candidate test examples (Non-member) generally follow a more uniform distribution; (2) the gradients from the last layer leak the most membership information, as the last layer already contains the membership information that leaks from the output of the previous layers [Nasr *et al.*, 2019].

After we derive the membership information from GMM, the problem becomes similar as the scenario when the membership is known (Section 3.2).

4 Experimental Setup

4.1 Datasets

Purchase. This dataset contains 600 different products (attributes), and each user has a binary record which indicates whether she has bought each of the products (a total of 197,324 data records). The records are clustered into 100 classes based on the similarity of the purchases, and our objective is to identify the class of each user's purchases [Shokri *et al.*, 2017]. For Purchase100 dataset, we randomly choose two attributes with different distributions, which correspond to 130-th and 595-th attribute. We further generate a Purchase2 dataset with 2 classes to investigate how ARA performs when the true label is unknown (Section 5.4).

Genome. We implement a public genome dataset called HS3D (Homo Sapiens Splice Sites Dataset)[1] for splice site prediction. It is a binary classification task in computational genetics which determines whether the target nucleotide sequence contains certain functional unit. The prepared dataset consists of total 28800 negative and 2880 positive samples. All the genome sequences are of length 20 (#attributes=20). Similarly, for Genome dataset, we randomly choose two attributes with different attribute distributions, which correspond to 7-th and 17-th attribute.

Location. We use a curated location dataset from the publicly available set of mobile users' location "check-ins" in the

[1] http://www.sci.unisannio.it/docenti/rampone/

Dataset	Purchase100		Location		Genome	
Target y	purchase type		geosocial type		sequence detection	
Attribute X_s	130-th	595-th	1-st	69-th	7-th	17-th
X_s variance	26.20e+8	11.85e+8	2.6e+6	2070	138019	77610
Cramer's V	0.2302	0.2310	0.1747	0.0412	0.0017	0.0016

Table 1: Summary of datasets. Cramer's V captures statistical correlation between Y and X_s (0 indicates no correlation and 1 indicates perfectly correlated).

Foursquare social network [Shokri *et al.*, 2017]. Each record in the resulting dataset has 446 binary attributes, representing whether the user visited a certain region or location type. We cluster the location dataset into 30 classes, each representing a different geosocial type. The classification task is to predict the user's geosocial type given his or her record.

For all datasets, we randomly select 10% records as public set, and partition the rest data into training set and test set with a ratio of 8:2. A summary of all datasets is given in Table 1.

4.2 Experiment Configuration

For all datasets, we take *Multi-Layer Perceptron* (MLP) model. For each participant in FL, we train an MLP model with one hidden layer of 128 units. We take ReLU (rectifier linear units) as the activation function, and SGD as the optimizer. We vary batch size from $\{8, 32, |D_v|\}$, where $|D_v|$ refers to the local data size of the victim participant. For practicality, we adopt FedAvg protocol [McMahan *et al.*, 2017] and suppose each participant sends local updates after 1 epoch of local training. We set the maximum training epochs to 100. For cos-matching optimization, we adopt Adam optimizer. For all datasets, we set Gumbel softmax parameter via grid search.

In FL, each participant only has a limited amount of local data, henceforth, according to the available data size, we vary the local data size of each participant from {50, 100, 500, 1000} for Purchase100 and Genome datasets, and {50, 100, 300} for Location dataset. We consider gradients of different epochs chosen from $\{1, \cdots, 100\}$: untrained model (pre1); first 2 epochs (pre2); first 5 epochs (pre5); 5 chosen epochs from {10, 20, 30, 40, 50} (gap10); last 5 epochs (last5).

Evaluation Metrics. For ARA, we use *reconstruction accuracy*, which is defined as the fraction of the correct reconstruction for member and non-member attribute. When the membership is unknown, we compute *MIA attack accuracy* as the fraction of the correct membership predictions.

4.3 Baselines

Random guess (random): The attacker randomly picks a value from all the possible values of size K as the predicted attribute, i.e., reconstruction accuracy=1/K.

L2-BFGS [Zhu *et al.*, 2019]: The sensitive attribute is optimized to minimize the L2 distance between virtual gradients and real gradients.

Attack model trained on public set: If the attacker has access to some auxiliary data with true attribute values, the attacker can train a multi-class attack model which takes the non-sensitive attributes $\{(x_1, \cdots, x_d)\}$ of the public data as inputs, and uses the true attribute value $\{x_s\}$ as labels. After the attack model is trained, the attacker can predict the sensitive attribute of any record of interest.

Inference from statistics (stats): When the true label is known, the server can directly infer the true attribute from the statistics calculated on all the enumerated records of interest without performing any optimization. In statistics based method, the attacker can store checkpoints of the victim participant's model across all epochs (suppose 1 epoch corresponds to 1 communication round). For all guess points, the attacker uses the local model of the victim participant to produce a statistic vector of size K in each epoch. In this way, the attacker derives a statistic matrix of size $K * E$, with each row corresponding to a guess point associated with one possible attribute value, and each column corresponding to one epoch.

After these statistic matrices are built, the attacker can make decisions on all guess points based on the following heuristics: (1) label status matrix: choose row with the max number of correct predictions across epochs; (2) probability score matrix: choose row with the max probability sum across all epochs; (3) loss norm matrix: choose row with the min loss norm sum across all epochs; (4) final loss matrix: choose row with the min final loss; (5) last-layer gradient norm (gradient w.r.t. w on each guess point x): choose row with the max gradient norm sum across all epochs; (6) gradient true label matrix: last-layer gradient connecting to the true label which corresponds to the only negative gradient value. Choose the guess point with the largest gradient value (closer to 0). (7) majority: choose the majority row from above statistics.

We remark that statistics based baselines require the prior knowledge of the true label and need to calculate per-record gradient using victim participant's model, while our reconstruction method has no such constraints.

5 Experimental Results

For experimentation, we consider different capabilities of the attacker, including membership={known, unknown}, attribute prior={known, unknown}, and true label={known, unknown}. We also investigate three scenarios: {victim participant isolate=1; victim participant isolate=0, #participant=2; victim participant isolate=0, #participant=10}. Moreover, we study the efficacy of the iterative communication rounds in FL by integrating the gradients of different states {pre1, pre2, pre5, gap10, last5} (Sec 4.2). We further investigate different local data size $|D_v| = \{50, 100, 300, 500, 1000\}$ and different local training batch size B=$\{8, 32, |D_v|\}$.

5.1 When membership is known

For all datasets, we first consider the following scenario for illustration purpose: isolate=1, membership=known, true label=known, and local training batch size B=$|D_v|$. Table 2, Table 3 and Table 4 list the attribute reconstruction accuracy for Purchase100, Location, and Genome datasets respectively. In particular, we list the best result for the stats based method and our method across {pre1, pre2, pre5, gap10, last5} epochs (Sec 4.2).

As evidenced by all tables, using our cos-matching method to predict the attribute mostly outperforms the baselines of {random, stats}. Moreover, our method outperforms the baseline models trained on 100/1000 public examples ({public100,

Table 2: 130-th and 595-th attribute reconstruction accuracy for Purchase100 when isolate=1, membership=known, prior=known, and true label=known. Best results are shown in bold.

Attribute	130-th (var=26.20e+8)				595-th (var=11.85e+8)			
$\|D_v\|$	50	100	500	1000	50	100	500	1000
random	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
public100	0.8600	0.8400	0.8160	0.7880	0.6000	0.7000	0.6940	0.7090
public1000	0.8200	0.8500	0.8260	0.8240	0.7000	0.7600	0.7340	0.7090
stats	0.6600	0.7400	0.6500	0.6560	0.7400	0.7800	0.6800	0.6560
L2-BFGS	1.0000	0.9600	0.7740	0.7650	1.0000	**1.0000**	0.7240	0.7330
Our	**1.0000**	**1.0000**	**0.9240**	**0.8310**	**1.0000**	0.9900	**0.8080**	**0.7590**

Table 3: 1-st and 69-th attribute reconstruction accuracy for Location when isolate=1, membership=known, prior=known, and true label=known. Best results are shown in bold.

Attribute	1-st (var=2.6e+6)			69-th (var=2070)		
$\|D_v\|$	50	100	300	50	100	300
random	0.5	0.5	0.5	0.5	0.5	0.5
public100	0.8600	0.8500	0.8200	0.5800	0.5700	0.5600
public1000	0.8800	0.8400	0.8367	0.6000	0.5800	0.6433
stats	0.7800	0.7600	0.6900	0.7800	0.7000	0.6567
L2-BFGS	1.0000	1.0000	0.8500	1.0000	0.7900	0.7100
Our	**1.0000**	**1.0000**	**0.8633**	**1.0000**	**0.9500**	**0.7967**

public1000}) in most cases. This implies that relying on the correlations of only a handful of labeled public examples to train an attack model is not sufficient to reconstruct the extract attribute value. More importantly, our proposed cos-matching method performs better than L2-BFGS for most datasets in most scenarios. We hypothesise that the reason is that Euclidean distance may not be effective in evaluating the difference between high dimensional vectors due to the curse of dimensionality. Moreover, Euclidean distance fails to capture the directional difference between gradient vectors. By contrast, cosine similarity is well-known to be more robust in evaluating similarity between high dimensional model parameters than Euclidean distance [Huang *et al.*, 2021]. However, we also notice that L2-BFGS using L2 distance and L-BFGS performs better than our method on Genome, when local data size $|D_v| = \{50, 100\}$, as shown in Table 4. We speculate that this is due to the fact that L2-BFGS is more suitable to the dataset with smaller number of attributes, but wider range. For example, Genome dataset only has 20 attributes, and the range length of the chosen 7-th and 17-th attributes equals 4, rather than 2 as in Purchase100 and Location.

Impact of the attribute distribution: We find that attribute distribution is closely related to the attribute reconstruction accuracy. Compared with attribute with higher variance, attribute with lower variance is harder to attack. For instance, from Table 2 for Purchase100 dataset, attribute reconstruction accuracy on the 130-th attribute with higher variance is higher than the 595-th attribute. From Table 3 for Location dataset, attribute reconstruction accuracy on the 1-st attribute with higher variance is higher than the 69-th attribute. From Table 4 for Genome dataset, 7-th attribute with slightly higher variance is also slightly higher than the 17-th attribute, especially when the local data size is larger.

Impact of the participant isolation: As shown in Table 5, compared with isolate=1, isolate=0 generally delivers slightly lower reconstruction accuracy. We articulate that the reason is: in isolate=0, the server does not isolate the victim participant,

Table 4: 7-th and 17-th attribute reconstruction accuracy for Genome when isolate=1, membership=known, prior=known, and true label=known. Best results are shown in bold.

Attribute	7-th (var=138019)				17-th (var=77610)			
$\|D_v\|$	50	100	500	1000	50	100	500	1000
random	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
public100	0.2600	0.3300	0.2500	0.2620	0.2800	0.2500	0.2680	0.2660
public1000	0.3200	0.3100	0.2760	0.2840	0.3400	0.3300	0.2840	0.2830
stats	0.2400	0.2800	0.3420	0.3210	0.2800	0.3000	0.3120	0.2950
L2-BFGS	**1.0000**	**0.9100**	0.3180	0.2700	**1.0000**	**0.7300**	0.2820	0.2690
Our	0.5600	0.5200	**0.3540**	**0.3550**	0.5800	0.4800	**0.3720**	**0.3350**

Table 5: Impact of participant isolation on ARA when membership=known, prior=known, and true label=known (We fix local batch size B=$|D_v|$=100).

	ARA					
Data	Purchase100		Genome		Location	
Attribute	130	595	7	17	1	69
isolate=1	**1.0000**	**0.9900**	**0.5200**	**0.4800**	**1.0000**	**1.0000**
isolate=0, participant=2	1.0000	0.8900	0.4200	0.3700	1.0000	1.0000
isolate=0, participant=10	0.9700	0.8600	0.4200	0.3700	1.0000	1.0000

hence the downloaded global model in each communication round integrates all participants' model updates, weakening attack efficacy.

Impact of the local batch size: There is a consistent conclusion on the relationship between local model batch size and attack performance. As evidenced by Table 6, for the investigated batch size B=$\{8, 32, |D_v|\}$, B=$|D_v|$ mostly outperforms B=$\{8, 32\}$, and the attack performance generally becomes better with larger batch size. We attribute this to the more randomness introduced by the smaller batch size during local model training, the released local model gradients cannot capture the full pass over the whole local data, making it harder to conduct ARA.

Impact of the training phases: Attack performance differs a lot when the gradients of different epochs are considered. Generally, the gradients of the previous epochs ({pre1, pre2, pre5}) outperform the gradients of later epochs ({gap10, last5}), as manifested in Table 7. Our interpretation is that at the beginning of the training process, the gradients are large such that more useful information can be utilized for ARA. With the evolution of training, local model tends to converge thus gradients decrease, leading to weak signals contained in the released gradients.

Impact of the local data size: Local data size also impacts attack performance. Generally, ARA becomes less effective when the local data size increases, as demonstrated in Table 8. This is because the released gradients contain the information for all the examples, while attribute reconstruction is inherently a post-hoc per-example task. Therefore, it becomes harder to reconstruct per-example attribute when the victim participant has more data.

Impact of model complexity: To investigate how model complexity impacts ARA, we use Location data and three fully connected models, with hidden layer sizes of 128, 1024, {1024, 256} respectively. As shown in Table 9, the model with higher capacity is more vulnerable to ARA. This partly reflects the fact that the model with higher capacity can memorize more information about the training data [Zhang *et al.*, 2017].

Table 6: Impact of batch size on ARA when isolate=1, membership=known, prior=known, and true label=known (We fix the local data size of the victim participant as $|D_v| = 100$, and vary the victim participant local training batch size B=$\{8, 32, |D_v|\}$).

	ARA					
Data	Purchase100		Genome		Location	
Attribute	130	595	7	17	1	69
B=8	0.8500	0.6900	**0.4000**	0.3500	0.9800	0.9000
B=32	0.9600	0.8000	0.3900	0.4300	0.9300	0.7700
B=$\|D_v\|$	**1.0000**	**0.9900**	0.3300	**0.4600**	**1.0000**	**0.9500**

Table 7: Impact of training phases on ARA when isolate=1, membership=known, prior=known, and true label=known (We fix local batch size B=$|D_v|$=100).

	ARA					
Data	Purchase100		Genome		Location	
Attribute	130	595	7	17	1	69
pre1	1.0000	0.9500	0.3300	0.4500	1.0000	0.9200
pre2	1.0000	0.9700	**0.3300**	**0.4600**	1.0000	0.9400
pre5	**1.0000**	0.9700	0.3200	0.4200	**1.0000**	**0.9500**
gap10	0.9800	**0.9900**	0.2600	0.3000	0.9800	0.8100
last5	0.9300	0.9300	0.1900	0.3200	0.9600	0.8100

5.2 When membership is unknown

When the membership information is unknown, we first conduct membership inference attack (MIA) by following our proposed GMM based method in Section 3.3. In particular, we focus on Purchase100 and Location datasets under the scenario of isolate=1 for illustration purpose. As shown in Table 10, on Purchase100 dataset, based on the last-layer gradient variances calculated on all the candidate records with the enumerated 130-th attribute, we get MIA accuracy of $\{0.9900, 1.0000, 0.9805, 0.9568\}$ for {50,100,500,1000} randomly sampled member and non-member records respectively. Similarly, for 595-th attribute, we get MIA accuracy of $\{0.9900, 1.0000, 0.9800, 0.9568\}$ for {50,100,500,1000} randomly sampled member and non-member records respectively. These results significantly outperform the accuracy of 73.4% reported in [Nasr *et al.*, 2019], which relies on a more complex Encoder-Decoder attack model. On Location dataset with enumerated 1-st or 130-th attribute, we also get relatively high MIA accuracy, as shown in Table 11.

After we determine the membership information, we follow Algorithm 1 to reconstruct the member attribute. We observe that the conclusions for the scenario when the membership is known (Section 5.1) also generally hold in the scenario when the membership is unknown.

5.3 When the prior distribution is unknown

We further investigate ARA when the prior distribution is unknown. In this scenario, we initialize the virtual attribute values from a random distribution $\mathcal{N}(0, 1)$. Table 12 shows the 1-st and 69-th attribute reconstruction accuracy for Location dataset across different local training batch size B=$\{8, 32, |D_v|\}$, when isolate=1, membership=known, true label=known, but prior=unknown. Although the attack performance is slightly lower than the attack performance when prior=known as in Table 3, the accuracy results are still mostly higher than the baselines of {random, public100, public1000, stats}. Similar observations also hold for Purchase100, by comparing Table 13 with Table 2.

Table 8: Impact of local data size $|D_v|$ when isolate=1, membership=known, prior=known, and true label=known (We adopt gradients of pre5, and fix the victim participant batch size B=$|D_v|$).

	ARA					
Data	Purchase100		Genome		Location	
Attribute	130	595	7	17	1	69
$\|D_v\| = 50$	**1.0000**	**1.0000**	**0.4400**	**0.4800**	**1.0000**	**1.0000**
$\|D_v\| = 100$	1.0000	0.9700	0.3200	0.4200	1.0000	0.9400
$\|D_v\| = 500$	0.9000	0.7840	0.2940	0.2960	-	-
$\|D_v\| = 1000$	0.8310	0.7480	0.2750	0.2550	-	-
$\|D_v\| = 300$	-	-	-	-	0.8667	0.7967

Table 9: Impact of model complexity. ARA for Location when isolate=1, membership=known, prior=known, and true label=known (We fix the victim participant local training batch size B=$|D_v|$). Best results are shown in bold.

Attribute	1-st (var=2.6e+6)			69-th (var=2070)		
$\|D_v\|$	50	100	300	50	100	300
128	1.0000	1.0000	0.8633	1.0000	0.9500	0.7967
1024	1.0000	1.0000	1.0000	1.0000	1.0000	0.9633
{1024,256}	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.9833**

5.4 When the true label is unknown

When the true label is unknown, we can initialize the virtual label either from a random distribution if the label distribution is unknown, or from the prior label distribution if the label distribution is known. In this scenario, the attacker needs to optimize both the virtual attribute and virtual label in Algorithm 1. For experimentation, we investigate Purchase2 and Genome, all with 2 classes. For both datasets, our method can reconstruct the true label with 100% accuracy. In particular, given the prior label distribution, it takes <200 iterations for the extract label reconstruction, while it takes longer when the label distribution is unknown.

When isolate=1, membership=known, prior=known, but true label=unknown, for Purchase2, we achieve ARA of $\{1.0000, 1.0000, 0.8360, 0.7570\}$ for 130-th attribute with $|D_v| = \{50, 100, 500, 1000\}$ respectively. Similarly, we achieve ARA of $\{1.0000, 0.9800, 0.7300, 0.7380\}$ for 595-th attribute. For Genome, ARA performance is similar to the scenario when the true label is known (Table 4).

6 Conclusion

In this work, we initiate a more practical and interesting attack against FL system, called *attribute reconstruction attack* (ARA), in which participants share their epoch-averaged gradients instead of small batch-averaged gradients. In particular, we consider a malicious parameter server in the FL system. To launch ARA, we propose a more effective and efficient method called cos-matching, which utilizes the cosine similarity to measure the angular distance between the true update from the victim participant and the virtual update. Our method exhibits significantly improved performance compared to most existing baselines in all settings we studied. We also conduct a systematic exploration on the factors that may affect ARA. Extensive experiments on real-world datasets and comprehensive

Table 10: Purchase100 MIA using GMM when isolate=1, prior=known, true label=known, but membership=unknown (For the victim participant, we vary its local data size from $|D_v|$={50, 100, 500, 1000}, and local batch size B=$\{8, 32, |D_v|\}$).

	MIA							
$\|D_v\|$	50		100		500		1000	
Attribute	130	595	130	595	130	595	130	595
B=8	0.9900	0.9900	0.9925	0.9800	0.8860	0.8865	0.8245	0.8205
B=32	0.9700	**0.9950**	0.9900	0.9950	0.9005	0.9110	0.8160	0.7990
B=$\|D_v\|$	**0.9900**	0.9900	**1.0000**	**1.0000**	**0.9805**	**0.9800**	**0.9568**	**0.9568**

Table 11: Location MIA using GMM when isolate=1, prior=known, true label=known, but membership=unknown (For the victim participant, we vary local data size of the victim participant from $|D_v|$={50, 100, 300}, and local batch size B=$\{8, 32, |D_v|\}$).

	MIA					
$\|D_v\|$	50		100		300	
Attribute	1	69	1	69	1	69
B=8	0.9650	**0.9850**	0.9450	**0.9650**	0.7958	0.8000
B=32	0.9850	0.9800	0.9325	0.9375	0.8000	0.8000
B=$\|D_v\|$	**0.9850**	0.9800	**0.9600**	0.9575	**0.8958**	**0.8933**

Table 12: 1-st and 69-th attribute reconstruction accuracy for Location when isolate=1, membership=known, true label=known, but prior=unknown. Best results are shown in bold.

Attribute	1-st (var=2.6e+6)			69-th (var=2070)		
$\|D_v\|$	50	100	300	50	100	300
B=8	0.8600	0.8500	**0.7833**	0.8600	0.7100	0.5967
B=32	0.9400	0.8000	0.7433	0.8600	0.6400	0.6300
B=$\|D_v\|$	**0.9400**	**0.8800**	0.7667	**0.9400**	**0.7400**	**0.6500**

analysis offer new insights for designing privacy-preserving FL methods and systems. This work has not investigated the scenario when the attacker is one of the participants in FL system, which would immediately be our future work.

Table 13: 130-th and 595-th attribute reconstruction accuracy for Purchase100 when isolate=1, membership=known, true label=known, but prior=unknown. Best results are shown in bold.

Attribute	130-th (var=26.20e+8)				595-th (var=11.85e+8)			
$\|D_v\|$	50	100	500	1000	50	100	500	1000
B=8	0.8600	0.7800	0.6460	0.5830	0.6800	0.7100	0.6280	0.6380
B=32	0.9000	0.7600	0.6520	0.6440	0.8400	0.7500	0.6820	0.6480
B=$\|D_v\|$	**0.9400**	**0.8400**	**0.7660**	**0.7390**	**0.9200**	**0.8500**	**0.7300**	**0.6900**

References

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Hitaj *et al.*, 2017] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618. ACM, 2017.

[Huang *et al.*, 2021] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[Kairouz *et al.*, 2019] Peter Kairouz, H Brendan McMahan, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[Lyu *et al.*, 2020a] Lingjuan Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S Yu. Privacy and robustness in federated learning: Attacks and defenses. *arXiv preprint arXiv:2012.06337*, 2020.

[Lyu *et al.*, 2020b] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

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

[Nasr *et al.*, 2019] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753. IEEE, 2019.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE, 2017.

[Song and Raghunathan, 2020] Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. *arXiv preprint arXiv:2004.00053*, 2020.

[Yeom *et al.*, 2018] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[Zhang *et al.*, 2017] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

[Zhao *et al.*, 2020] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *CoRR, arXiv:2001.02610*, 2020.

[Zhu *et al.*, 2019] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, pages 14747–14756, 2019.