

Exploiting Unintended Feature Leakage in Collaborative Learning

Luca Melis, Congzheng Song, Emiliano De
Cristofaro, Vitaly Shmatikov

What this paper is about?

- Questions
 - do model updates in collaborative learning leak information about the training data;
 - if so, can an adversary obtain that information about training data;
- Contributions
 - Inferred “unintended” features, i.e., properties that hold for certain subsets of the training data, but not generically for all class members.
 - Pointed out the limitations of the attacks
 - Proposed possible defenses

What this paper is about?

- Goal:
 - find out what can be inferred about a participant's training dataset from the model updates revealed during collaborative model training?
- Focused on showing that apart from accurately classifying inputs, a classifier model may reveal the features that characterize a given class or help construct data points that belong to this class.

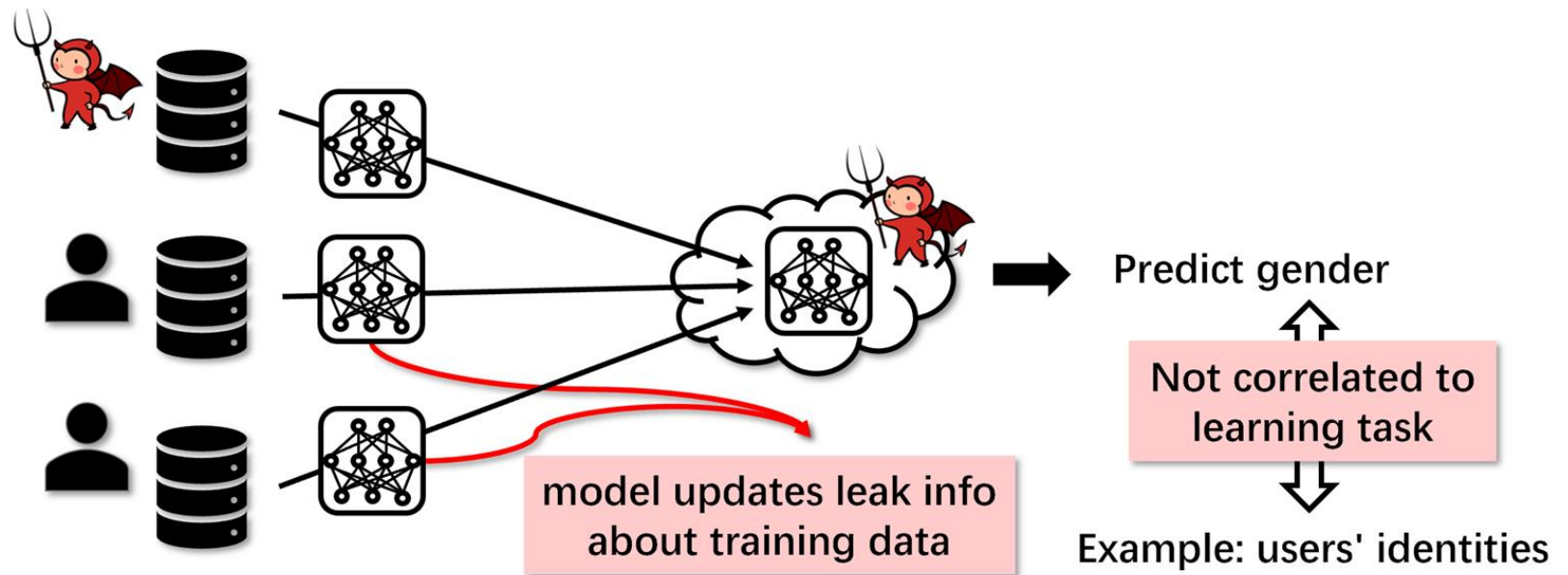
Background

Collaborative learning:

- partition the training dataset
- concurrently train separate models on each subset
- exchange parameters via a parameter server
- Each local model pulls the parameters from server, calculates the updates based on its current batch of training data, pushes updates back to the server.
- The server updates the global parameters.
- Saves time and resources

Background

Collaborative learning



*Here, the adversary has white-box access to the classifier model

Background

Collaborative learning with synchronized gradient updates:

- In each iteration each participant:
 - downloads the global model from the parameter server
 - locally computes gradient updates based on one batch of his training data
 - sends updates to the server
- The server:
 - waits for gradient updates from all participants
 - applies aggregated updates using stochastic gradient descent

Collaborative learning with synchronized gradient updates

Algorithm 1 Parameter server with synchronized SGD

Server executes:

Initialize θ_0

for $t = 1$ to T **do**

for each client k **do**

$g_t^k \leftarrow \text{ClientUpdate}(\theta_{t-1})$

end for

$\theta_t \leftarrow \theta_{t-1} - \eta \sum_k g_t^k$ \triangleright synchronized gradient updates

end for

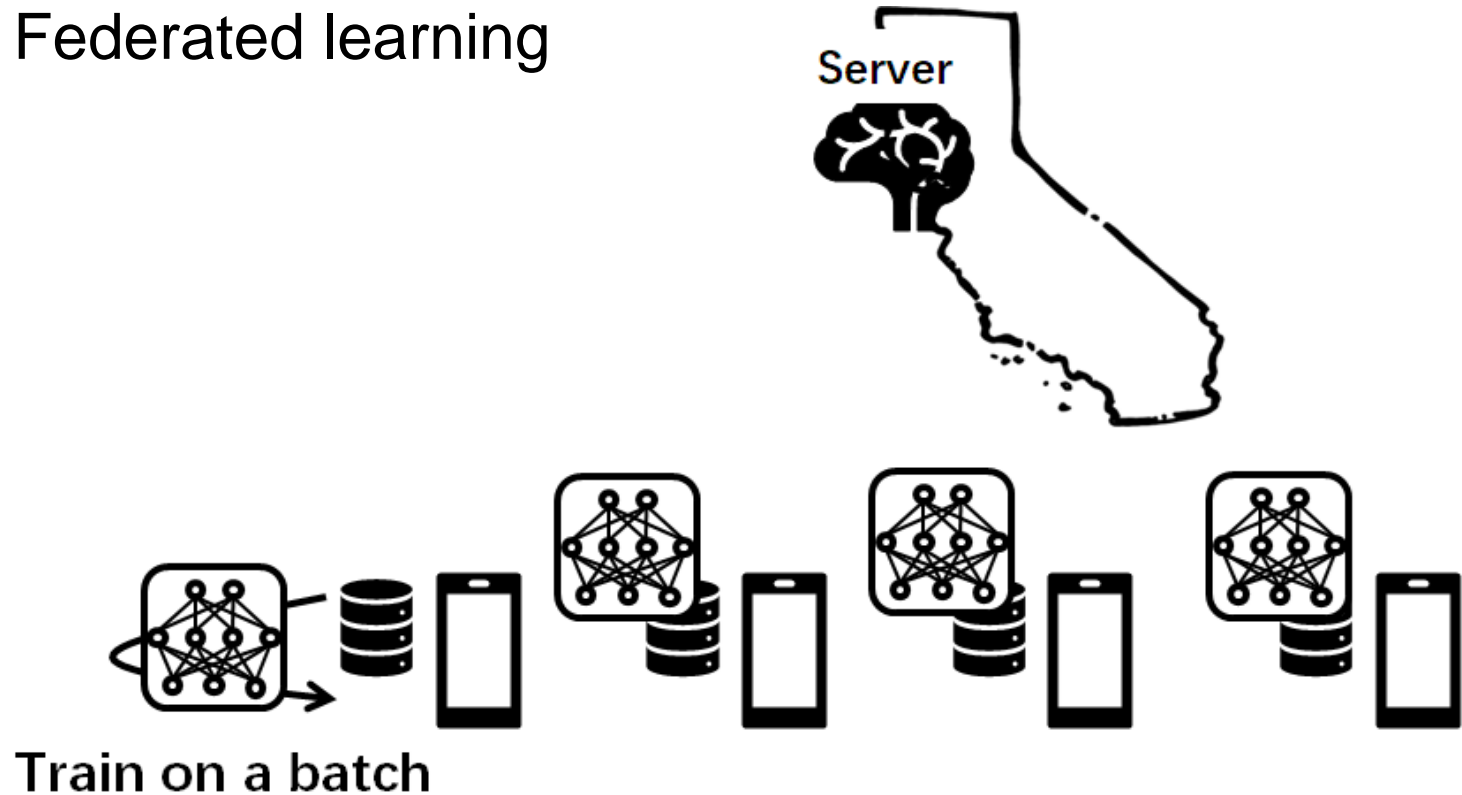
ClientUpdate(θ):

 Select batch b from client's data

return local gradients $\nabla L(b; \theta)$

Background

Federated learning



Background

Federated learning with model averaging:

- In each round:
 - k -th participant locally takes several steps of SGD on the current model using his entire training dataset of size n
 - each participant submits the resulting model to the server
- Server computes a weighted average
- Aggregation steps:
 - every participant aggregates the gradients computed on each local batch.
 - the server aggregates the updates from all participants.
 - ***this means globally visible updates are based not on batches but on participants' entire datasets**

Federated learning with model averaging

Algorithm 2 Federated learning with model averaging

Server executes:

Initialize θ_0

$m \leftarrow \max(C \cdot K, 1)$ **C (here, 1) = fraction of participants updating the model**

for $t = 1$ to T **do**

$S_t \leftarrow$ (random set of m clients)

for each client $k \in S_t$ **do**

$\theta_t^k \leftarrow \text{ClientUpdate}(\theta_{t-1})$

end for

$\theta_t \leftarrow \sum_k \frac{n^k}{n} \theta_t^k$

end for

n = total size of the training data (sum of all n^k)

▷ averaging local models

ClientUpdate(θ):

for each local iteration **do**

for each batch b in client's split **do**

$\theta \leftarrow \theta - \eta \nabla L(b; \theta)$

end for

end for

return local model θ

*The convergence rate of both collaborative learning approaches heavily depends on the learning task and the hyperparameters (e.g., number of participants and batch size).

INFERENCE ATTACKS

- K participants train a collaborative learning model
- One adversary with goal:
 - infer information about the training data of another, target participant by analyzing periodic updates to the joint model during training
- The updates depend on K and process of training.
- Inputs of adversary:
 - model updates in each round of collaborative learning.
 - If $K > 2$, aggregation of gradient update.

Leakage from model updates

- Model updates are computed from gradient descent

$$y = W \cdot h, \quad \frac{\partial L}{\partial W} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial W} = \frac{\partial L}{\partial y} \cdot \mathbf{h}$$

\mathbf{h} = features of \mathbf{x} learned to predict \mathbf{y} and gradient updates reveal the \mathbf{h} , hence leak properties of \mathbf{x} , ***uncorrelated*** with \mathbf{y} .

Given the adversary has examples of data with those properties, he can use supervised learning to infer properties from observed updates

Leakage from model updates

- Adversary observes gradient updates:
 - $g_{\text{obs}} = \Delta\theta_t - \Delta\theta_t^{\text{adv}}$
- Adversary feeds observed gradient updates to the batch property classifier f_{prop}

Leakage from model updates

Algorithm 3 Batch Property Classifier

Inputs: Attacker's auxiliary data $D_{\text{prop}}^{\text{adv}}, D_{\text{nonprop}}^{\text{adv}}$

Outputs: Batch property classifier f_{prop}

$G_{\text{prop}} \leftarrow \emptyset$ \triangleright Positive training data for property inference

$G_{\text{nonprop}} \leftarrow \emptyset$ \triangleright Negative training data for property inference

for $i = 1$ to T **do**

 Receive θ_t from server

 Run **ClientUpdate**(θ_t)

 Sample $b_{\text{prop}}^{\text{adv}} \subset D_{\text{prop}}^{\text{adv}}, b_{\text{nonprop}}^{\text{adv}} \subset D_{\text{nonprop}}^{\text{adv}}$

 Calculate $g_{\text{prop}} = \nabla L(b_{\text{prop}}^{\text{adv}}; \theta_t), g_{\text{nonprop}} = \nabla L(b_{\text{nonprop}}^{\text{adv}}; \theta_t)$

$G_{\text{prop}} \leftarrow G_{\text{prop}} \cup \{g_{\text{prop}}\}$

$G_{\text{nonprop}} \leftarrow G_{\text{nonprop}} \cup \{g_{\text{nonprop}}\}$

end for

Label G_{prop} as positive and G_{nonprop} as negative

Train a binary classifier f_{prop} given $G_{\text{prop}}, G_{\text{nonprop}}$

Datasets

- **Labeled Faces In the Wild**

- 13,233 62x47 RGB face images for 5,749 individuals with labels: gender, race, age, hair color, and eyewear

- **FaceScrub**

- 76,541 50x50 RGB images for 530 individuals with the gender label; the authors used a subset of 100 individuals with the most images, for a total of 18,809

- Collaborative models: CNN with three spatial convolution layers with 32, 64, and 128 filters, kernel size (3, 3), max pooling layers size 2, followed by two fully connected layers of size 256 and 2. ReLU is the activation function for all layers. Batch size is 32, SGD learning rate is 0.01.

Datasets

- **Labeled Faces In the Wild**

- 13,233 62x47 RGB face images for 5,749 individuals with labels: gender, race, age, hair color, and eyewear

52.5% male in the original dataset. What about the used fraction?

- **FaceScrub**


- 76,541 50x50 RGB images for 530 individuals with the gender label; the authors used a subset of 100 individuals with the most images, for a total of 18,809

- Collaborative models: CNN with three spatial convolution layers with 32, 64, and 128 filters, kernel size (3, 3), max pooling layers size 2, followed by two fully connected layers of size 256 and 2. ReLU is the activation function for all layers. Batch size is 32, SGD learning rate is 0.01.

Datasets

- **People in Photo Album (PIPA)**
 - 60,000 photos of 2,000 individual; 18000 images used
- **Yelp-health**
 - 17,938 reviews for 10 types of medical specialists
- **Yelp-author**
 - 16,207 reviews for 10 reviewers
- **FourSquare**
 - For experiments, a subset of 15,548 users who checked in at least 10 locations in New York with their gender was used
- **CLiPS Stylometry Investigation (CSI) Corpus**
 - student-written essays and reviews
 - 1,412 reviews, 80% of the reviews by females, 66% by authors from Antwerpen, rest from other parts of Belgium and the Netherlands

Datasets

- **People in Photo Album (PIPA)**
 - 60,000 photos of 2,000 individual; 18000 images used
- **Yelp-health**  How many specialists?
 - 17,938 reviews for 10 types of medical specialists
- **Yelp-author**
 - 16,207 reviews for 10 reviewers
- **FourSquare**
 - For experiments, a subset of 15,548 users who checked in at least 10 locations in New York with their gender was used
- **CLiPS Stylometry Investigation (CSI) Corpus**
 - student-written essays and reviews
 - 1,412 reviews, 80% of the reviews by females, 66% by authors from Antwerpen, rest from other parts of Belgium and the Netherlands

Results (property inference)

Target label		Property		LFW				Correlation	
Main T.	Infer T.	Corr.	AUC	Main T.	Infer T.	Corr.	AUC		
Gender	Black	-0.005	1.0	Gender	Sunglasses	-0.025	1.0		
Gender	Asian	-0.018	0.93	Gender	Eyeglasses	0.157	0.94		
Smile	Black	0.062	1.0	Smile	Sunglasses	-0.016	1.0		
Smile	Asian	0.047	0.93	Smile	Eyeglasses	-0.083	0.97		
Age	Black	-0.084	1.0	Race	Sunglasses	0.026	1.0		
Age	Asian	-0.078	0.97	Race	Eyeglasses	-0.116	0.96		
Eyewear	Black	0.034	1.0	Hair	Sunglasses	-0.013	1.0		
Eyewear	Asian	-0.119	0.91	Hair	Eyeglasses	0.139	0.96		

TABLE III: AUC score of single-batch property inference on LFW. We also report the Pearson correlation between the main task label and the property label.

Results (property inference)

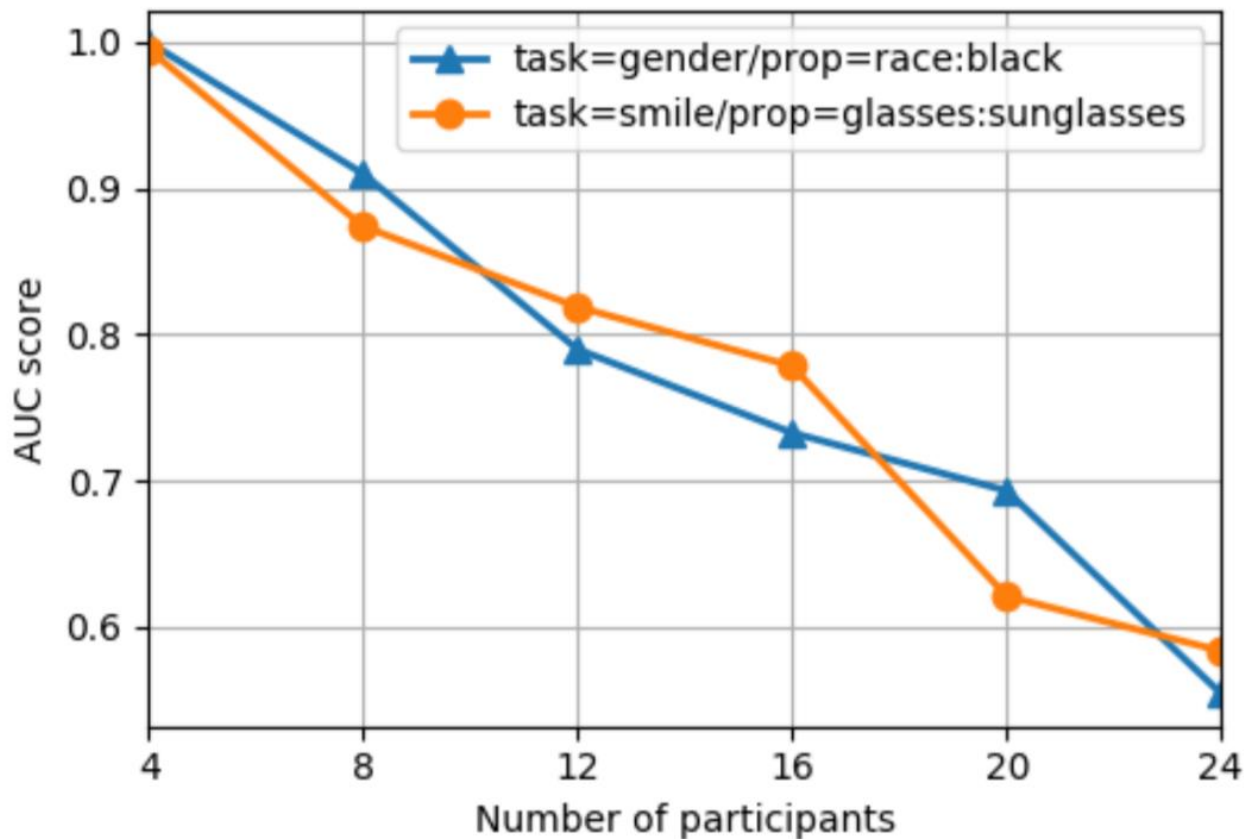
Target label		Property		LFW				Correlation	
Main T.	Infer T.	Corr.	AUC	Main T.	Infer T.	Corr.	AUC		
Gender	Black	-0.005	1.0	Gender	Sunglasses	-0.025	1.0		
Gender	Asian	-0.018	0.93	Gender	Eyeglasses	0.157	0.94		
Smile	Black	0.062	1.0	Smile	Sunglasses	-0.016	1.0		
Smile	Asian	0.047	0.93	Smile	Eyeglasses	-0.083	0.97		
Age	Black	-0.084	1.0	Race	Sunglasses	0.026	1.0		
Age	Asian	-0.078	0.97	Race	Eyeglasses	-0.116	0.96		
Eyewear	Black	0.034	1.0	Hair	Sunglasses	-0.013	1.0		
Eyewear	Asian	-0.119	0.91	Hair	Eyeglasses	0.139	0.96		

TABLE III: AUC score of single-batch property inference on LFW. We also report the Pearson correlation between the main task label and the property label.

**Main task and property are
not correlated**

Results (property inference)

LFW



Results (property inference)

The attack reaches 0.98 AUC after only 2 epochs and improves as the training progresses and the adversary collects more updates.

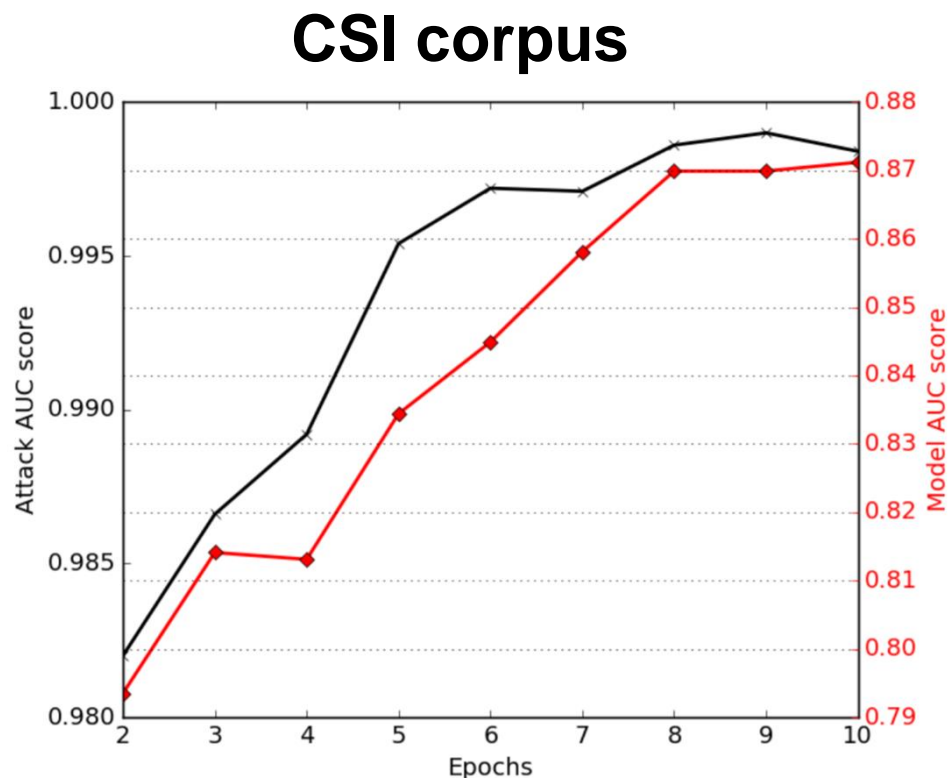


Fig. 6: Attack performance with respect to the number of collaborative learning epochs.

Results

Yelp Health (perfect AUC)

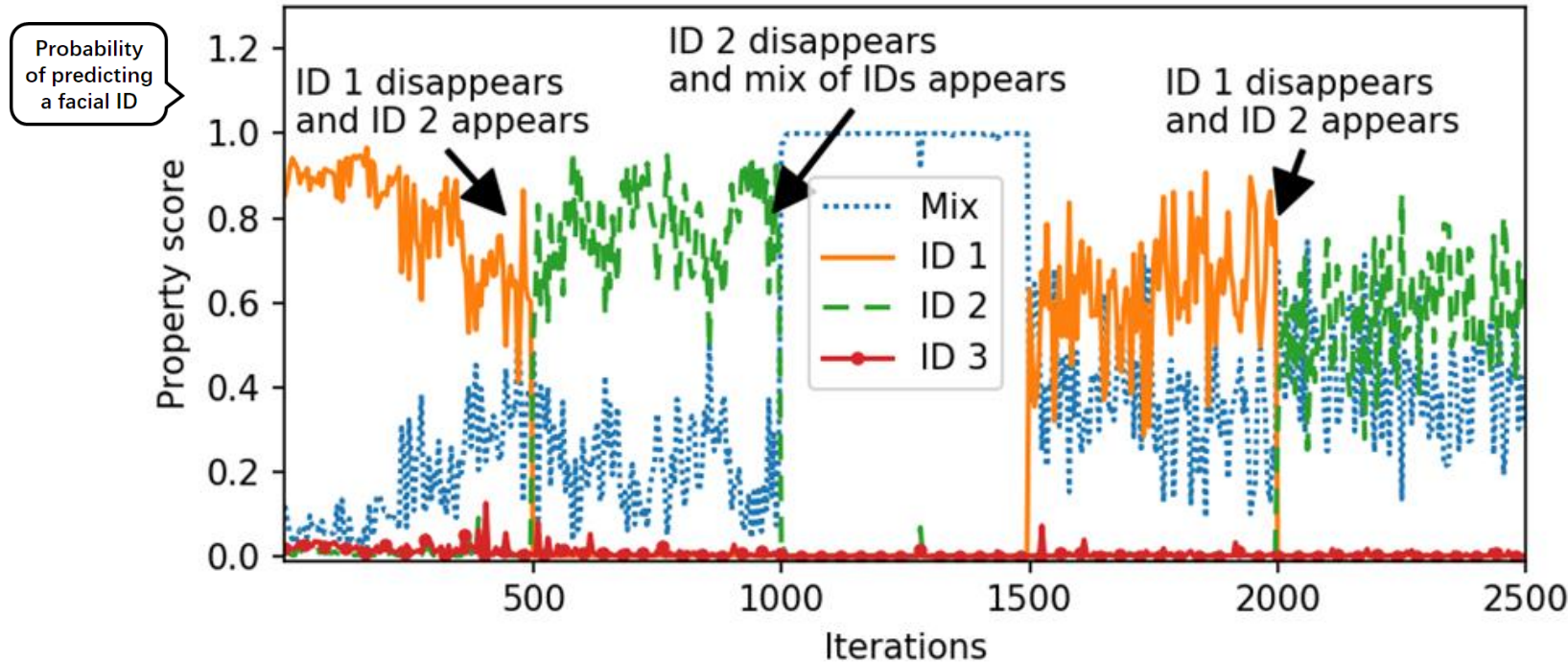
Yelp-health		FourSquare	
Batch Size	Precision	Batch Size	Precision
32	0.92	100	0.99
64	0.84	200	0.98
128	0.75	500	0.91
256	0.66	1,000	0.76
512	0.62	2,000	0.62

TABLE II: Precision of membership inference (recall is 1).

Results (occurrence inference)

FaceScrub

target=gender, property=facial ID



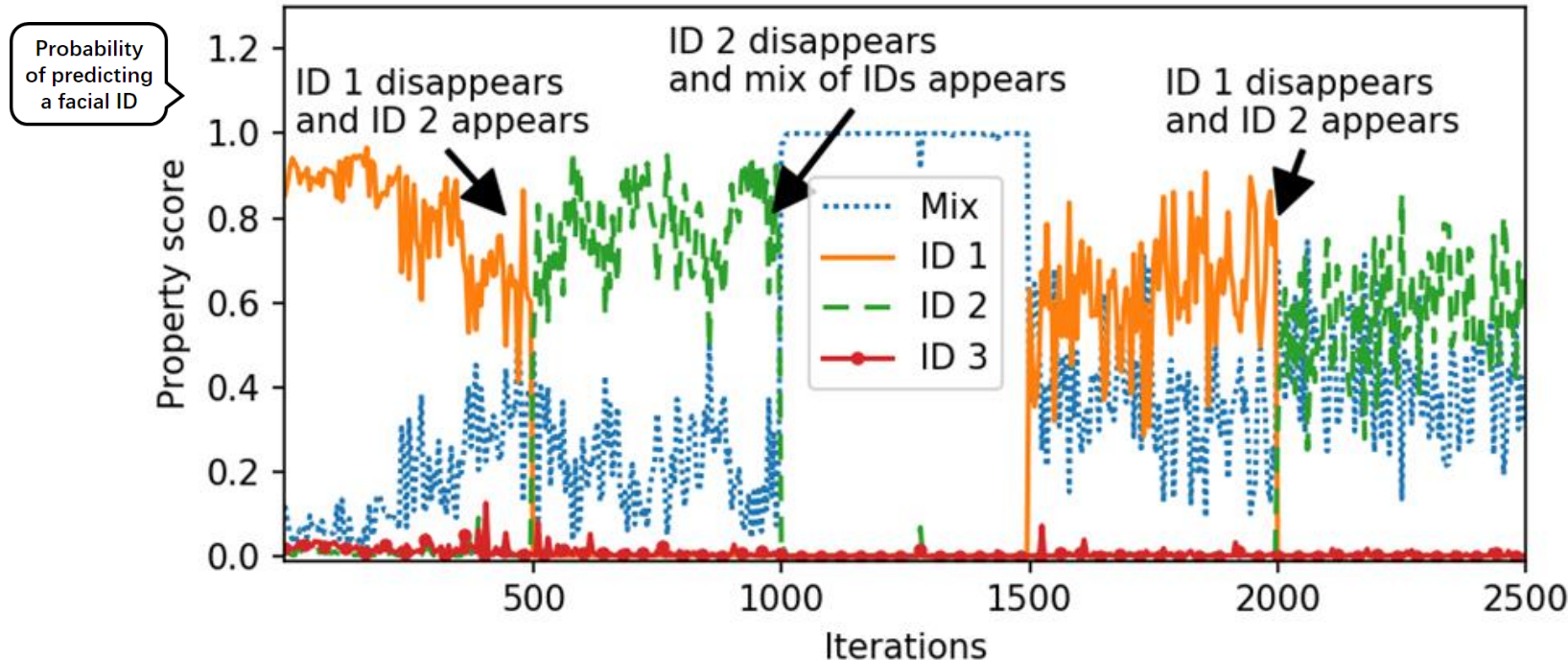
***Participant trains on faces of different people in different iterations**

Results (occurrence inference)

FaceScrub

Adversary can infer when images of a certain person appear and disappear in training data

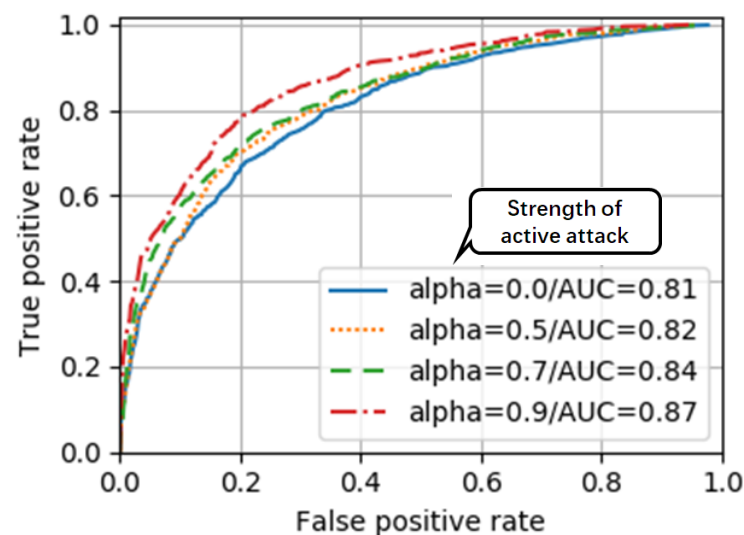
target=gender, property=facial ID



*Participant trains on faces of different people in different iterations

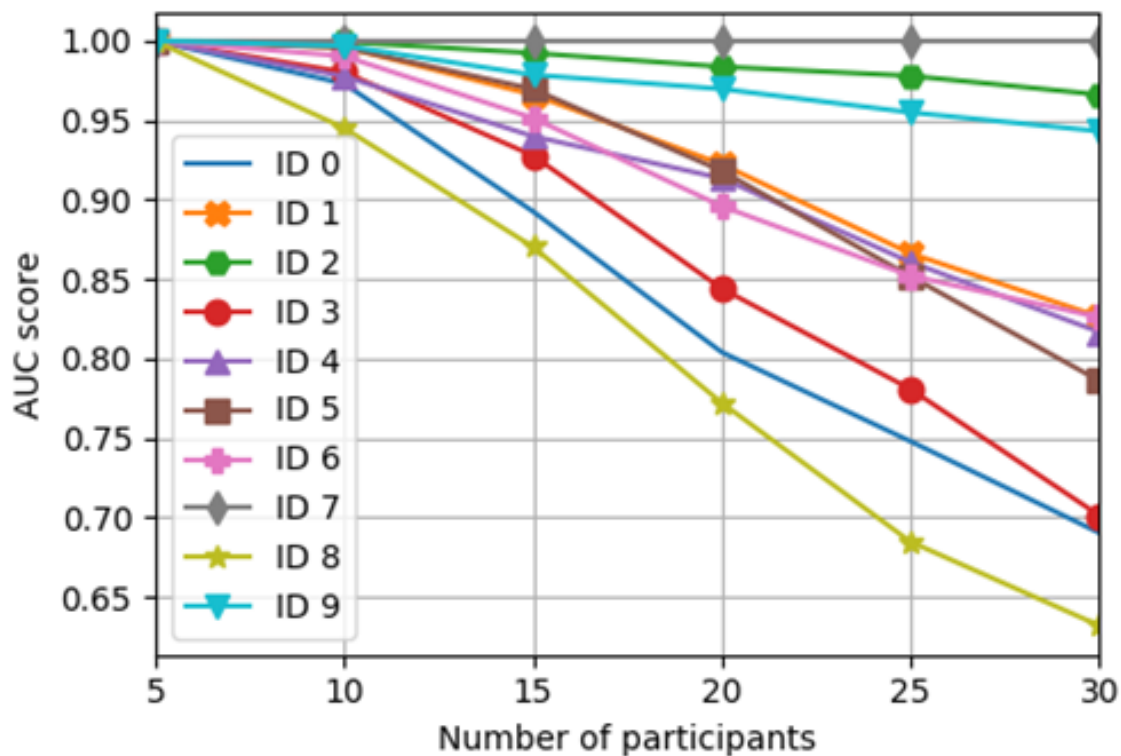
Results (ROC for different α)

- **FaceScrub**
 - Adversary can use multi-task learning to create a model that predicts both label and property
- Updates from this model can influence global model
- **Adversary can actively bias the model to leak property by sending crafted updates**



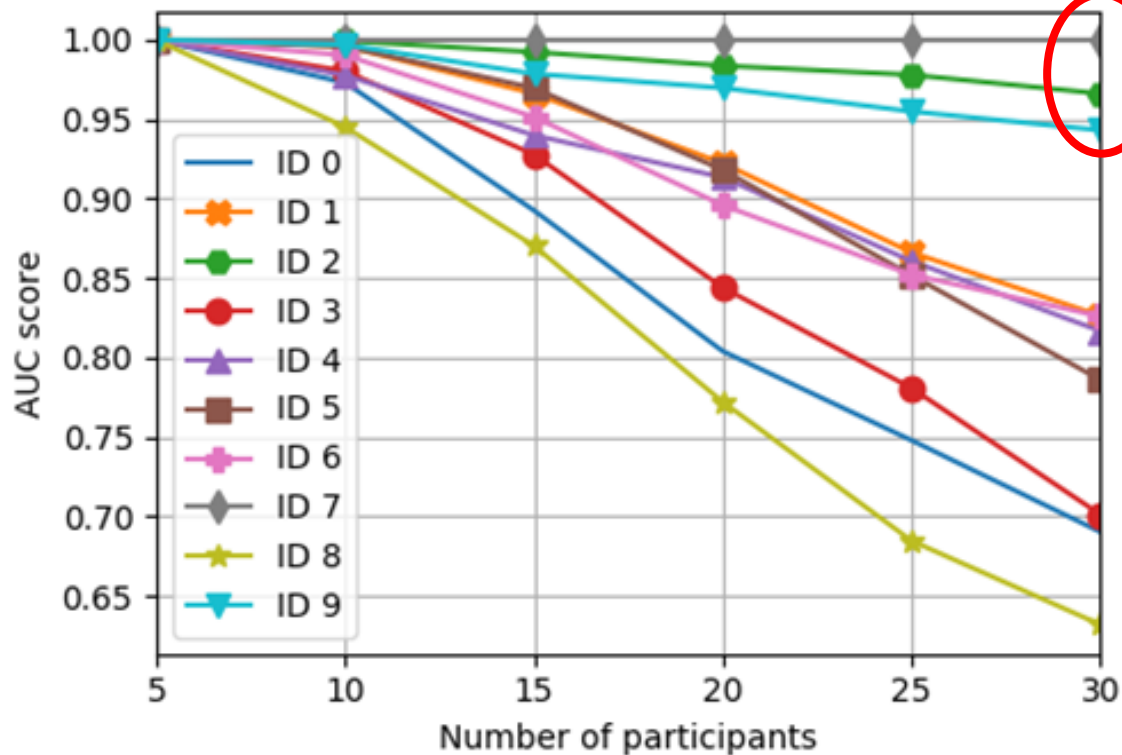
Results (aggregated updates)

Yelp Review



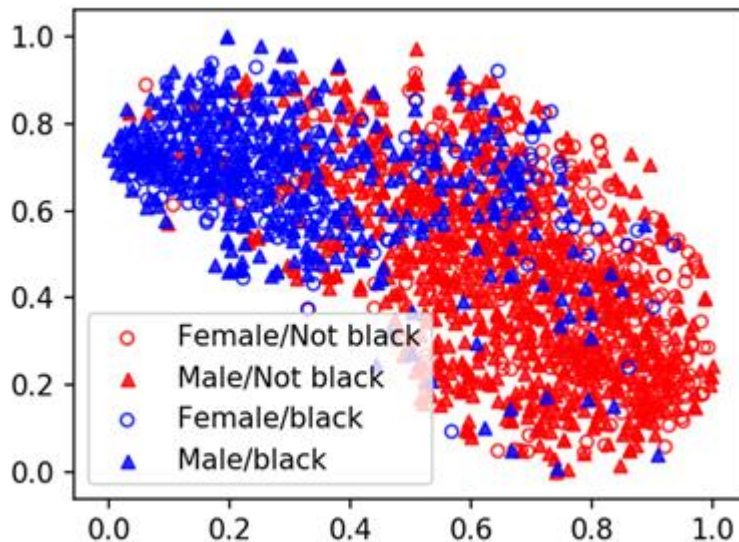
Results (aggregated updates)

Yelp Review



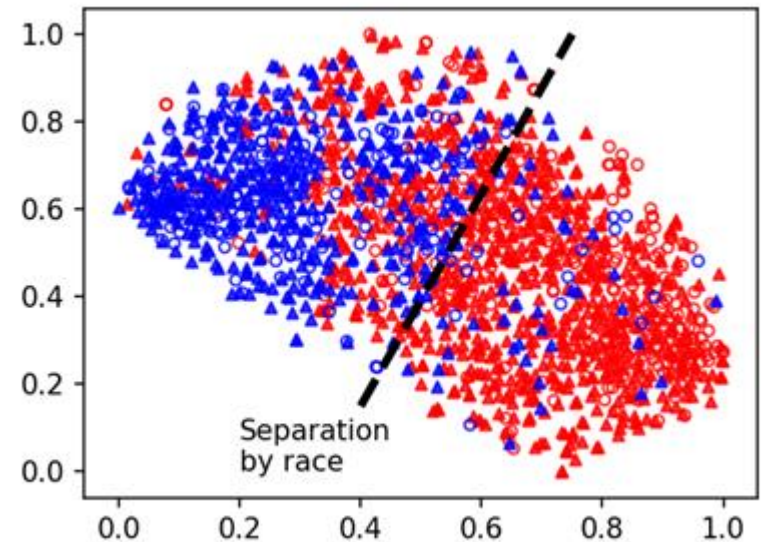
Leakage visualization in feature space

Main task
Circle points = Female
Triangle points = Male



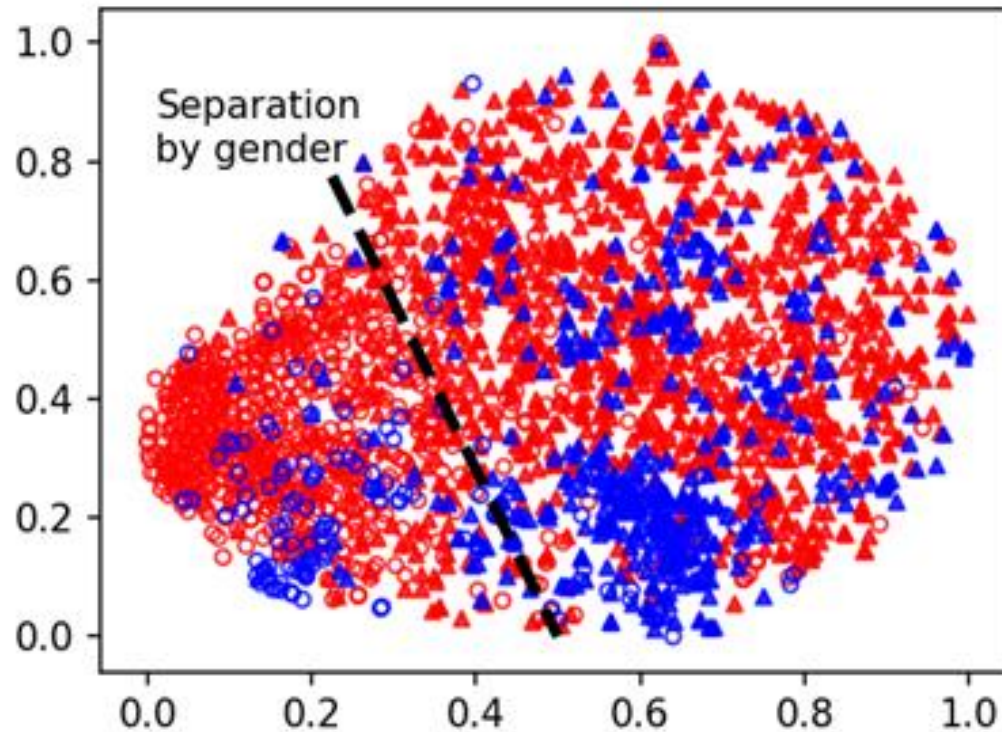
Pool 1 layer

Inferred property
Blue points = Black
Red points = Not black



Pool 2 layer

Leakage visualization in feature space



Final Layer

Limitations of the attack

- Lack of auxiliary training data labeled with the property the adversary wants to infer
- Number of participants in collaborative training
- Undetectable properties from model updates
- Attribution of inferred properties to a specific participant

Conclusion and discussion

- Paper proposed and evaluated inference attacks against **collaborative learning**
- Membership and other properties were inferred using those attacks
- The authors have shown that collaborative learning might unintentionally reveal certain information about the user datasets
- How useful is the attack, given that the performance drops with the increasing number of participants?
- Any specific examples to when this attack can be used and how severe are the possible consequences?

Questions
