



NORTH CAROLINA AGRICULTURAL
AND TECHNICAL STATE UNIVERSITY

Exploiting Unintended Feature Leakage in Collaborative Learning

(Luca Melis, Congzheng Song, Emiliano De Cristofaro, Vitaly Shmatikov)
Presented at: 40th IEEE Symposium on Security and Privacy (Oakland),
2019

Presented by: Ahmed Yiwere
in
ECEN 885002: Machine Learning in Cyber Security
(Professor: Dr. Mahmoud N. Mahmoud)

April 14, 2020

AGGIES **DO**



Outline

- ***Introduction***
- ***Proposed Attack Models***
- ***Experiments***
- ***Results and Analysis***
- ***Code Demonstration***
- ***Countermeasures***
- ***Limitations***
- ***Related Work***
- ***Conclusion***

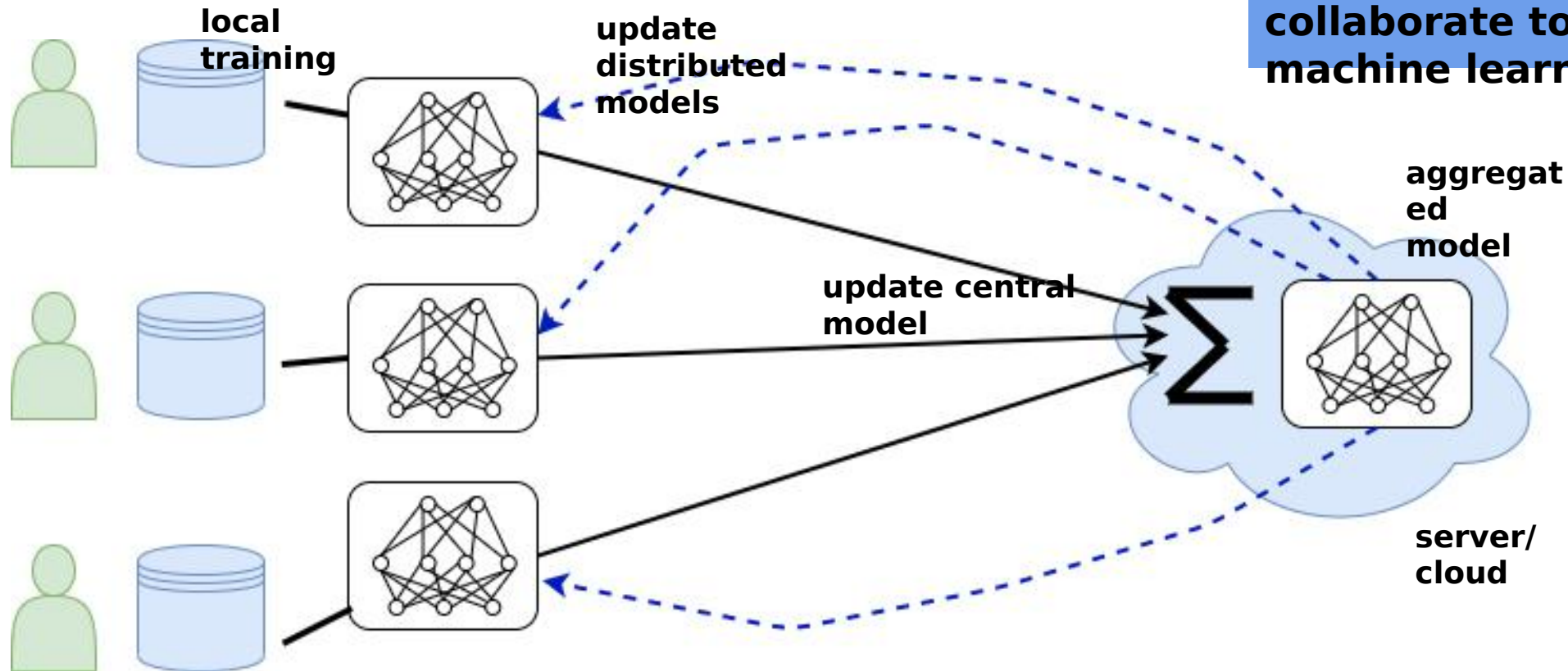


What is Collaborative Learning?



What is Collaborative Learning?

Multiple participants
collaborate to build a
machine learning model





What is Collaborative Learning?

Algorithm 1 Parameter server with synchronized SGD

Server executes:

```
Initialize  $\theta_0$ 
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  $\nabla L(b; \theta)$ 
```

Algorithm 2 Federated learning with model averaging

Server executes:

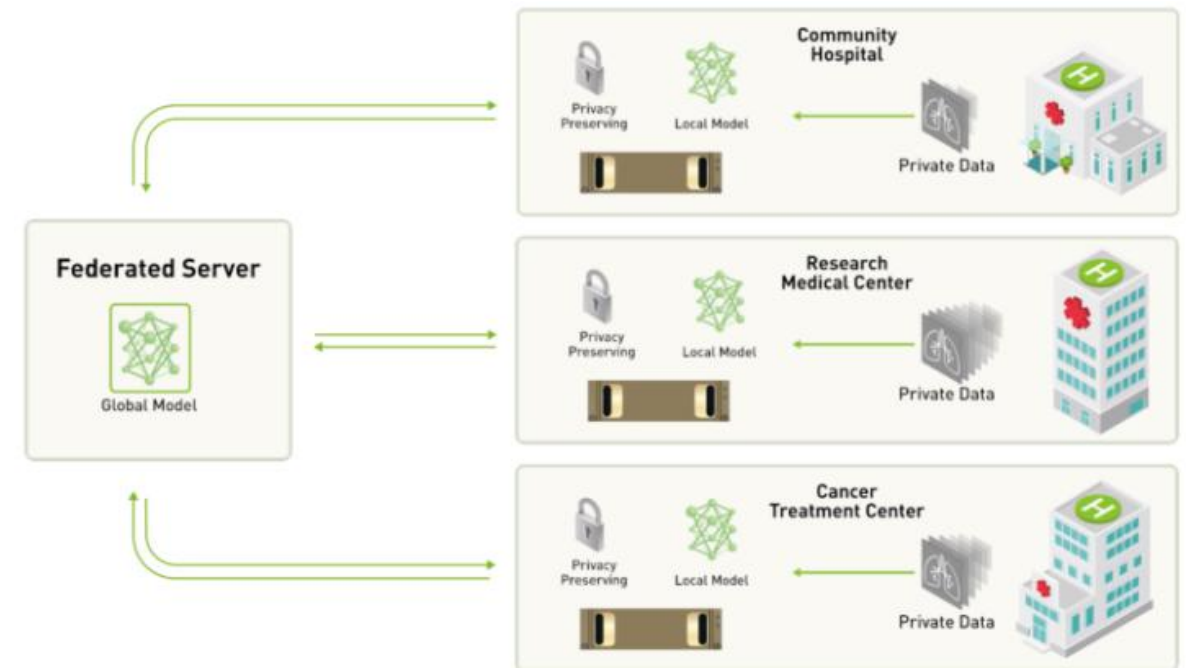
```
Initialize  $\theta_0$ 
 $m \leftarrow \max(C \cdot K, 1)$ 
for  $t = 1$  to  $T$  do
   $S_t \leftarrow$  (random set of  $m$  clients)
  for each client  $k \in S_t$  do
     $\theta_t^k \leftarrow \text{ClientUpdate}(\theta_{t-1})$ 
  end for
   $\theta_t \leftarrow \sum_k \frac{n^k}{n} \theta_t^k$   $\triangleright$  averaging local models
end for
```

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  $\theta$ 
```

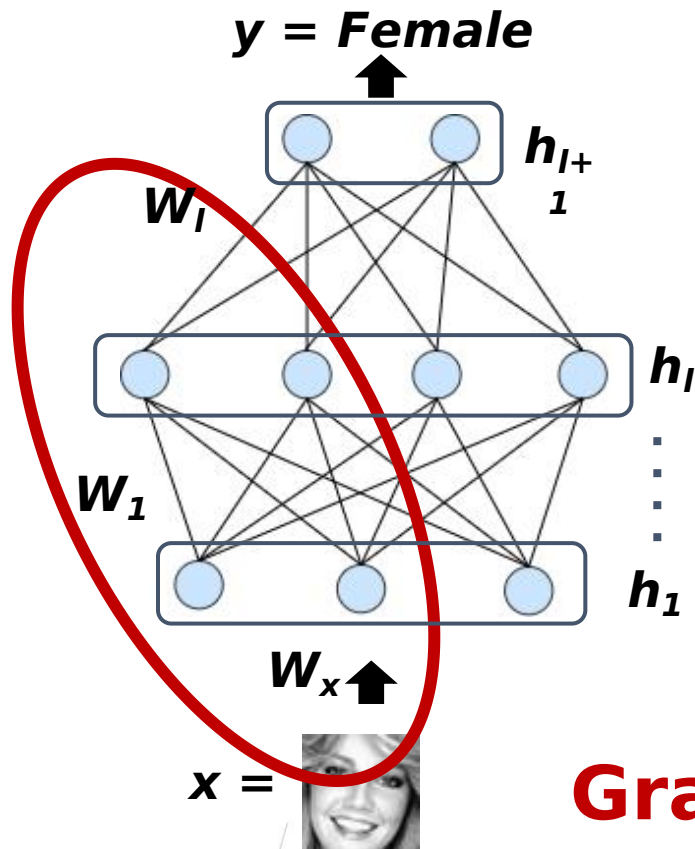
Why Collaborative Learning?

- Privacy of user data
- Access to more data with more variety
- Collaboration among organisations eg. hospitals
- Taking advantage of the current boom in edge computing eg. sensor networks, mobile phones
- Reduce data communication volume





Deep Learning Overview



- Map input \mathbf{x} to layers of features \mathbf{h} , then to output \mathbf{y} connected by \mathbf{W}

- Learn parameters to minimize loss:

$$\mathbf{W} = \operatorname{argmin}_{\mathbf{w}} L(\mathbf{x}, \mathbf{y})$$

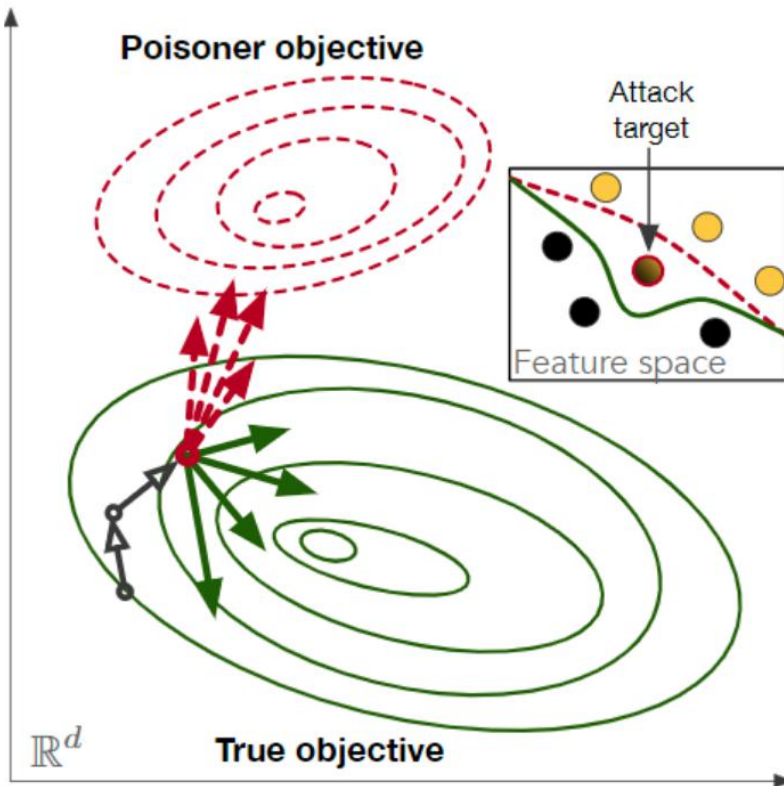
- Gradient descent on parameters:
 - In each iteration, train on a batch
 - Update \mathbf{W} based on gradient of

Gradients reveal information about the data

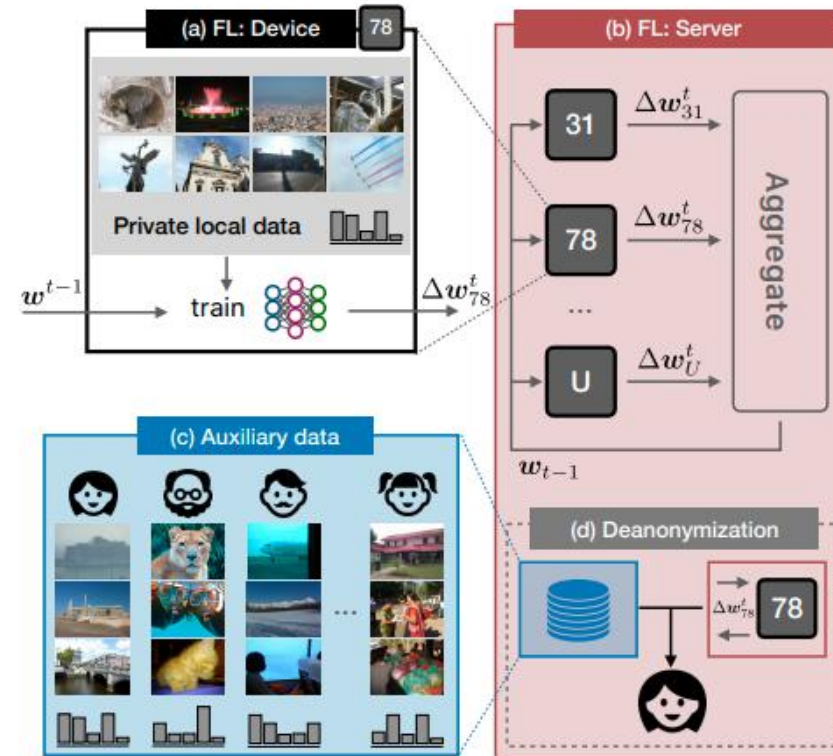


Security Vulnerabilities of Federated Learning

Poisoning Attacks



Inference Attacks



Membership Inference

Determine whether a particular data sample was used in training

Attribute Inference

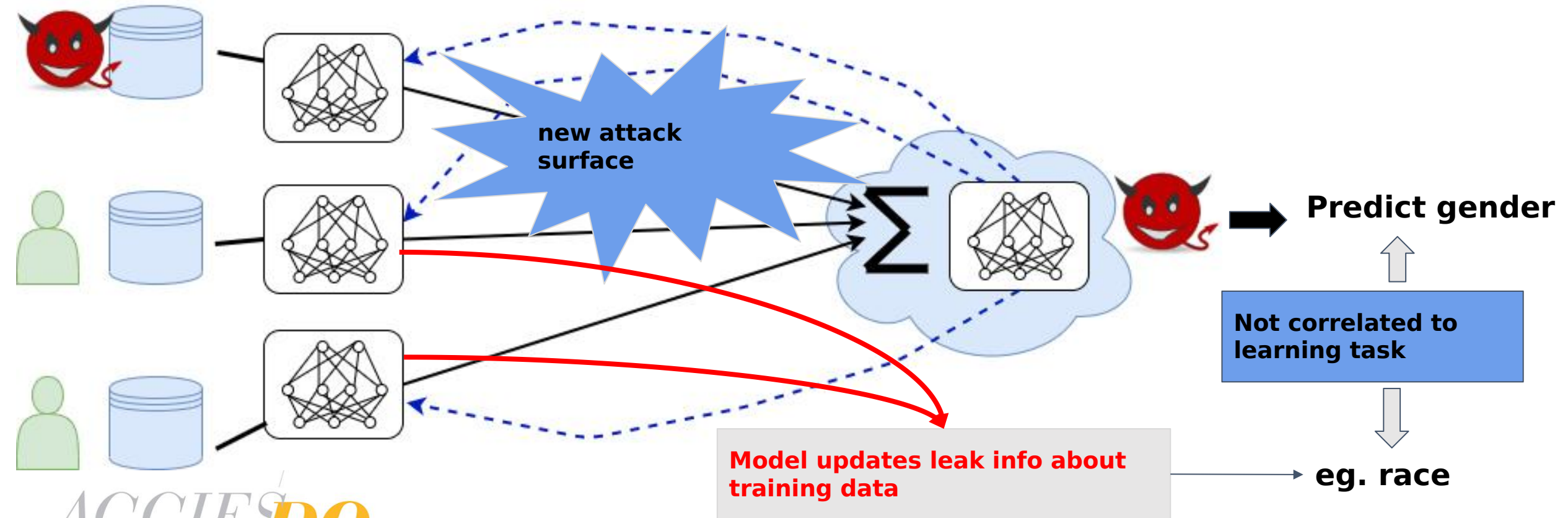
Identify properties that hold true for certain subsets of data

Model Inversion

Identify properties/features that characterize a class

Overview of The Paper

Goal: What can be inferred about a participant's training dataset from the model updates revealed during collaborative model training?



Overview of The Paper

Goal: What can be inferred about a participant's training dataset from the model updates revealed during collaborative model training?

Attacks demonstrated in this paper :

- Attribute Inference (*Property Inference*)
- Membership Inference
- Poisoning Attacks (*Active Property Inference*)



Threat Model



- Assume K participants in training ML model. $K \geq 2$
- One participant is an adversary
- Adversary's Goal: infer information about training data of other participants

Difference between consecutive snapshots of joint model:

$$\Delta\theta_t = \bar{\theta}_t - \theta_{t-1} = \sum_k \Delta\theta_t^k$$

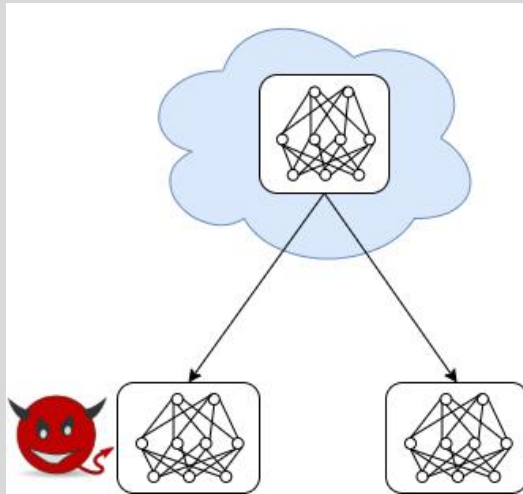
Aggregated updates from all participants except adversary:

$$\Delta\theta_t - \Delta\theta_t^{\text{adv}}$$

Threat Model

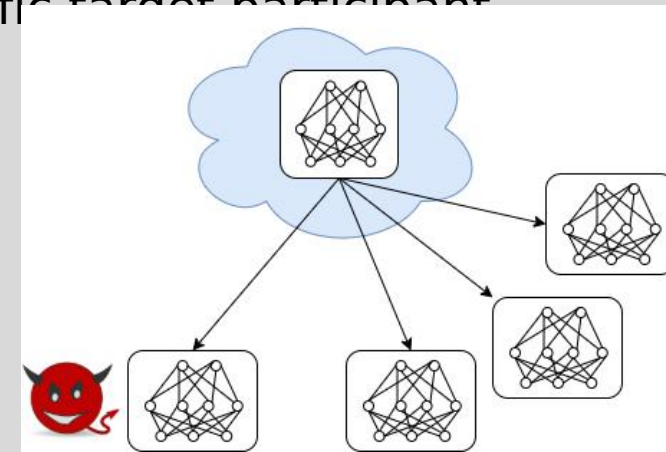
Two-Party

- $K = 2$
- One participant is an adversary
- Adversary's Goal: infer information about training data of the other participant



Multi-Party

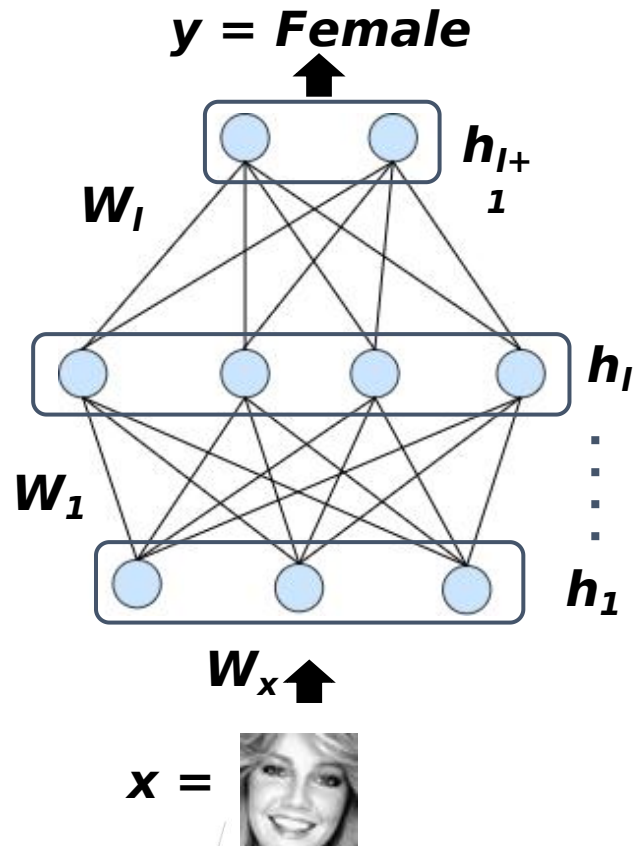
- $K > 2$
- One participant is an adversary
- Adversary's Goal: infer information about training data of the all other participants
- Difficult to trace inferred information to a specific target participant





Leakage from model updates

Leakage from gradients



Forward Pass

$$h_1 = x * W_x$$

$$h_2 = h_1 * W_1 = (x * W_x) * W_1$$

\vdots

$$h_l = h_{l-1} * W_{l-1} = (((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}$$

$$h_{l+1} = h_l * W_l = (((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$

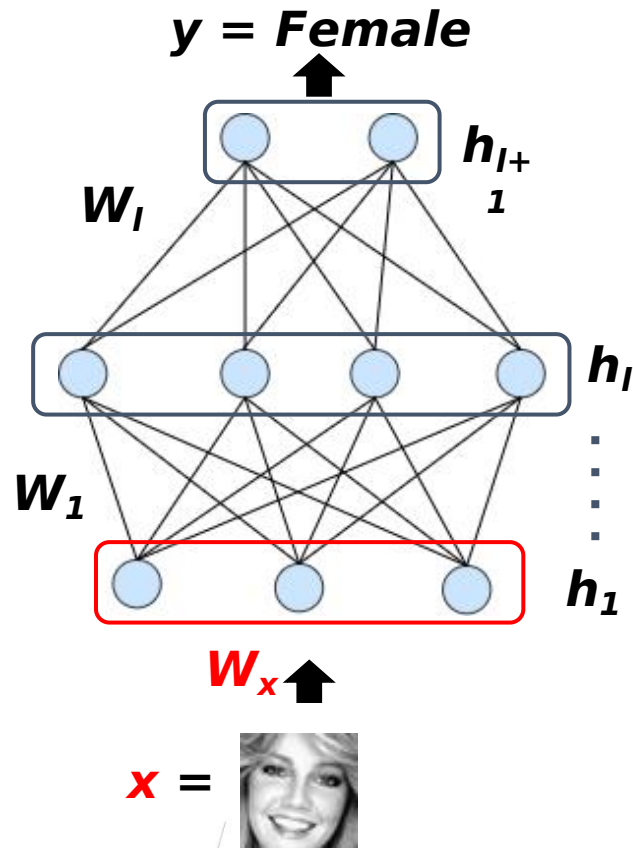
$$y = \text{activation}(h_{l+1})$$

$$y = \text{activation}((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$



Leakage from model updates

Leakage from gradients



Forward Pass

$$h_1 = x * W_x$$

$$h_2 = h_1 * W_1 = (x * W_x) * W_1$$

\vdots

$$h_l = h_{l-1} * W_{l-1} = (((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}$$

$$h_{l+1} = h_l * W_l = (((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$

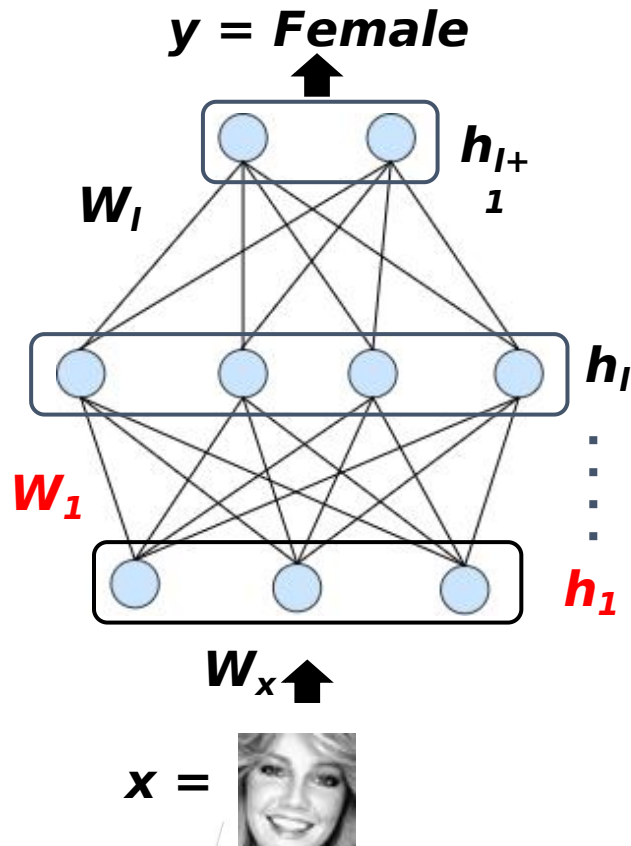
$$y = \text{activation}(h_{l+1})$$

$$y = \text{activation}((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$



Leakage from model updates

Leakage from gradients



Forward Pass

$$h_1 = x * W_x$$

$$h_2 = h_1 * W_1 = (x * W_x) * W_1$$

\vdots

$$h_l = h_{l-1} * W_{l-1} = (((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}$$

$$h_{l+1} = h_l * W_l = (((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$

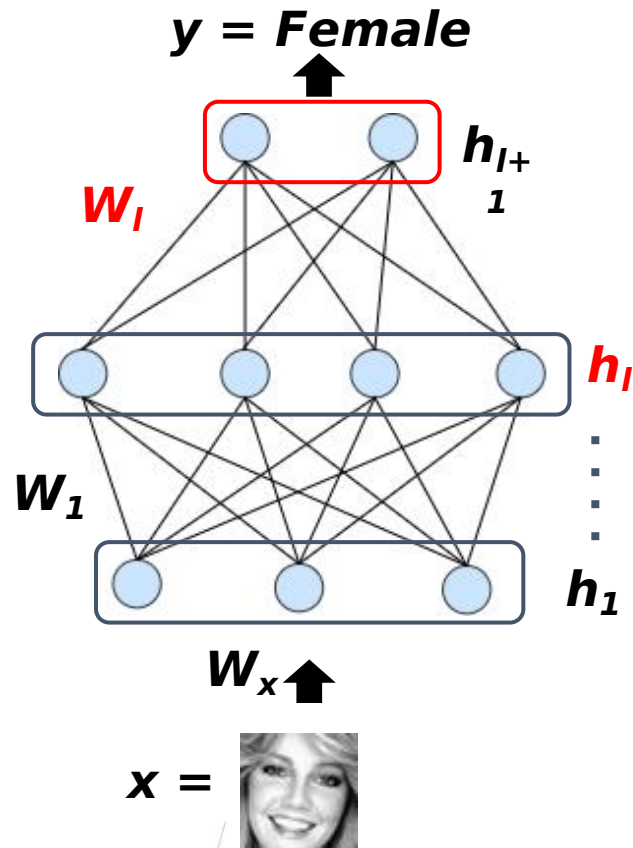
$$y = \text{activation}(h_{l+1})$$

$$y = \text{activation}((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$



Leakage from model updates

Leakage from gradients



Forward Pass

$$h_1 = x * W_x$$

$$h_2 = h_1 * W_1 = (x * W_x) * W_1$$

\vdots

$$h_l = h_{l-1} * W_{l-1} = (((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}$$

$$h_{l+1} = h_l * W_l = (((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$

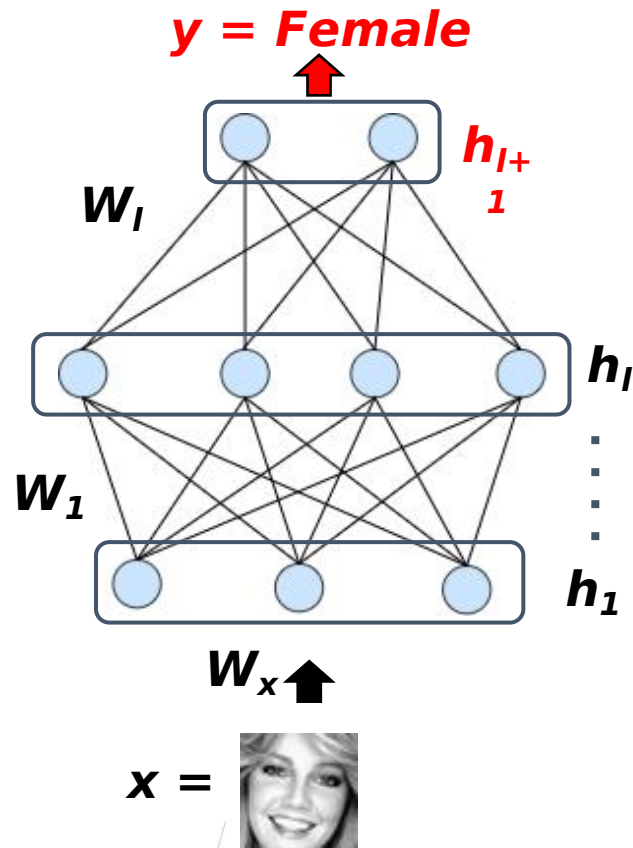
$$y = \text{activation}(h_{l+1})$$

$$y = \text{activation}((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$



Leakage from model updates

Leakage from gradients



Forward Pass

$$h_1 = x * W_x$$

$$h_2 = h_1 * W_1 = (x * W_x) * W_1$$

\vdots

$$h_l = h_{l-1} * W_{l-1} = (((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}$$

$$h_{l+1} = h_l * W_l = (((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$

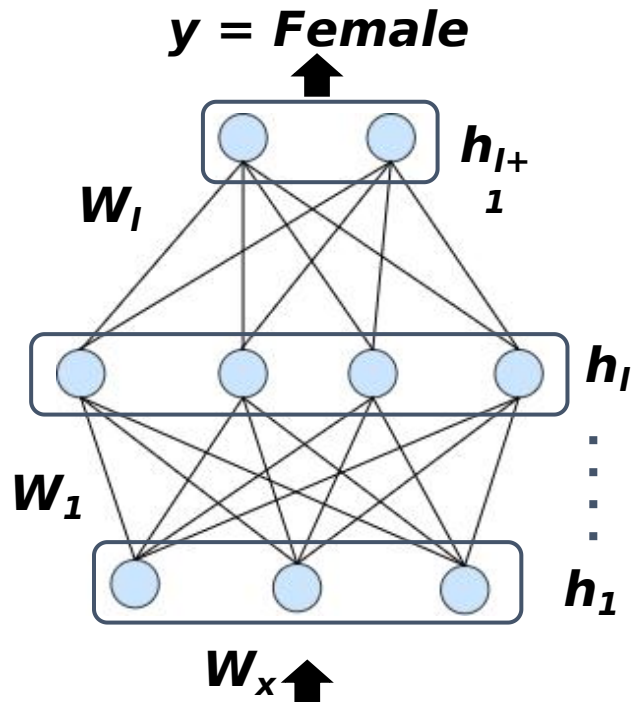
$$y = \text{activation}(h_{l+1})$$

$$y = \text{activation}((((x * W_x) * W_1) * W_2 * \dots) * W_{l-1}) * W_l)$$



Leakage from model updates

Leakage from gradients



Gradient Descent:

Minimize Loss Function

Loss, $L \Rightarrow \text{deviation}(y, y_{\text{true}})$

Generally:

$$\frac{\partial L}{\partial W_l} = \left(\frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h_{l+1}} \cdot \frac{\partial h_{l+1}}{\partial W_l} \right) = \frac{\partial L}{\partial h_{l+1}} * h_l$$

Example:

$$\frac{\partial L}{\partial W_1} = \left(\frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h_{l+1}} \cdot \frac{\partial h_{l+1}}{\partial W_l} \cdot \frac{\partial W_l}{\partial h_l} \cdot \frac{\partial h_l}{\partial W_{l-1}} \cdots \frac{\partial W_2}{\partial h_2} \cdot \frac{\partial h_2}{\partial W_1} \right) = \frac{\partial L}{\partial h_2} * h_1$$

Update Weight

$$W_1^t = W_1^{t-1} - \eta \cdot \frac{\partial L}{\partial W_1}$$

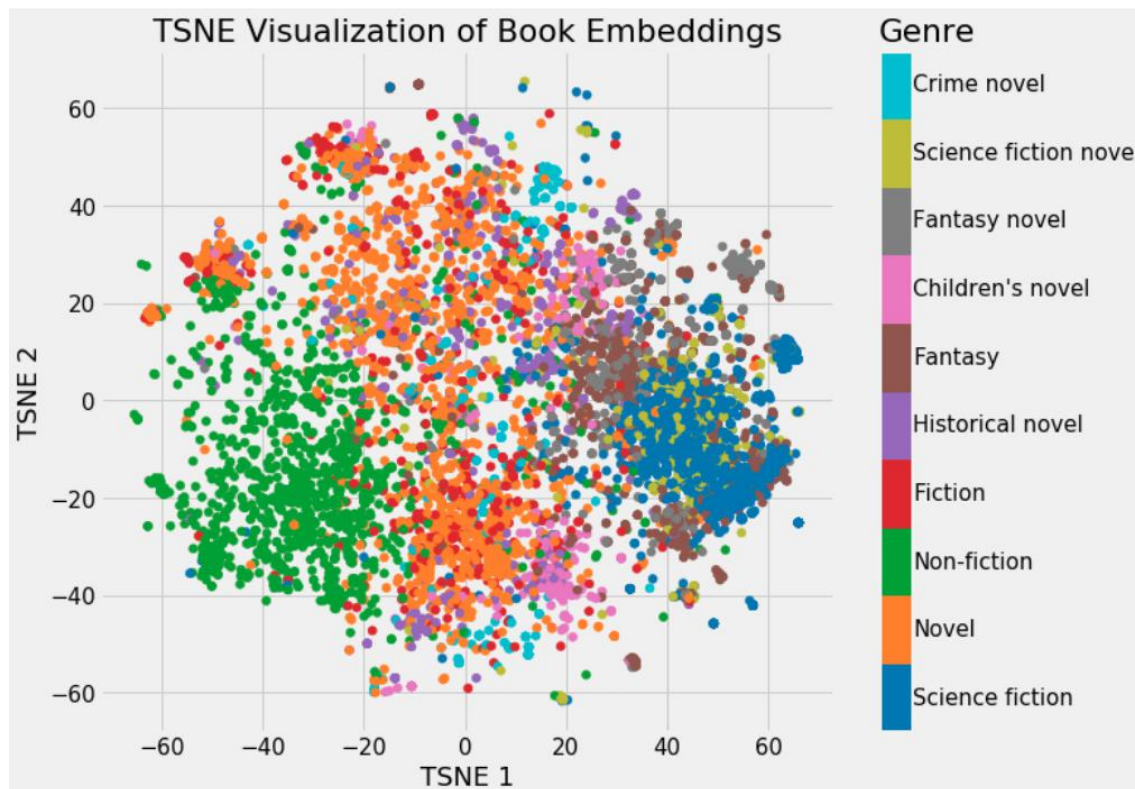
● h = features of x learned to predict y

● h leaks features of x which are uncorrelated with y



Leakage from model updates

Leakage from embedding layer



- **Embedding:** *a mapping of a discrete — categorical — variable to a vector of continuous numbers*

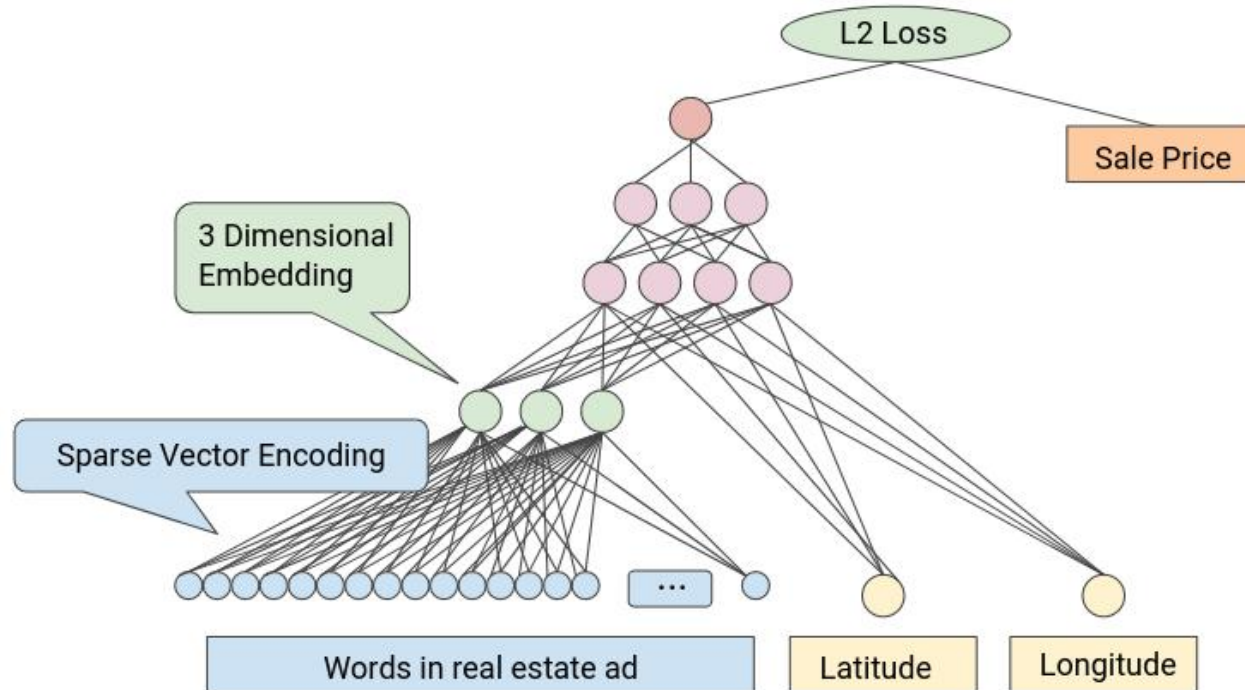
Can be used for:

- Dimensionality reduction
- Data visualization

Leakage from model updates

Leakage from embedding layer

Regression problem to predict home sales prices:



- Embedding layer: *a hidden layer used in neural networks to reduce dimensionality of high-dimension input*
- input is **non-numerical and discrete (categorical)**, and



Sample 1: {1, 0, 0, 0, ..., 1, 1, 1}

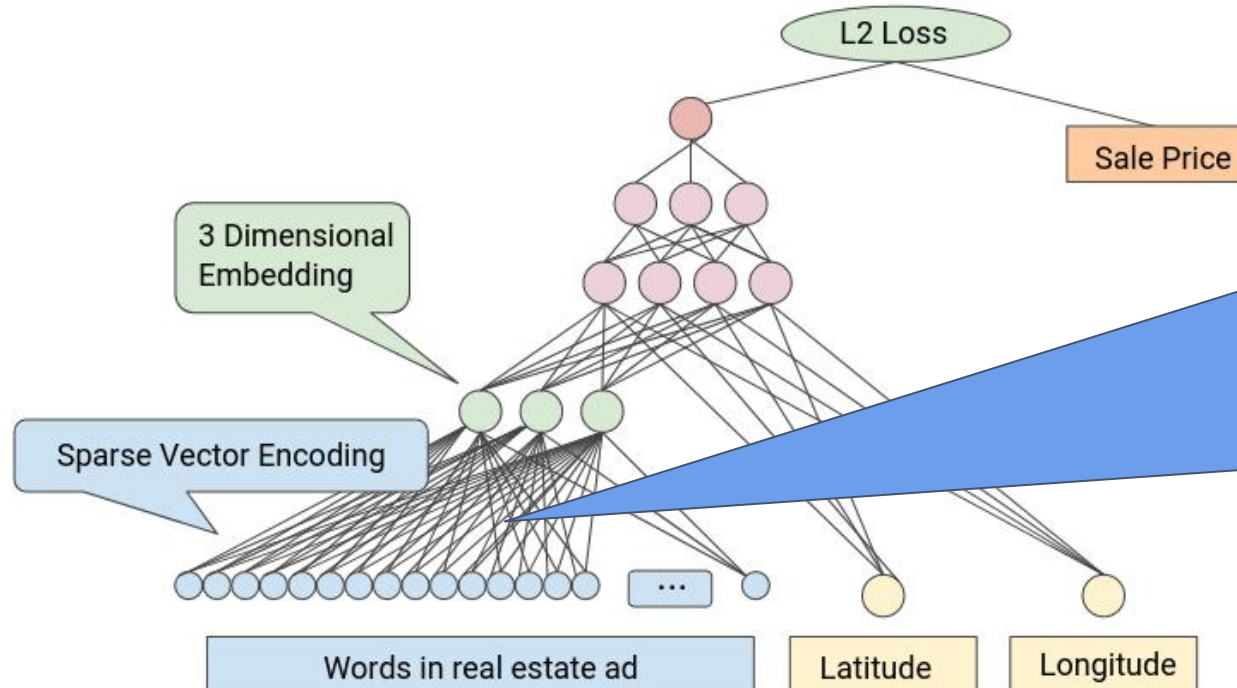
Sample 2: {1, 0, 0, 1, ..., 0, 1, 0}



Leakage from model updates

Leakage from embedding layer

Regression problem to predict home sales prices:



During training:

- Gradient matrix between input and embedding layer is also sparse
- Gradients are only updated for input features that are true (1).
- Features that are not present (0) have a gradient of zero
- The sparsity of the gradients can reveal which data was used in the input for training.
- Hence: **Membership Inference!**



Leakage from model updates

Model updates from gradient descent:

- Gradient updates reveal \mathbf{h} :

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

- \mathbf{h} = features of \mathbf{x} learned to predict \mathbf{y}

**leaks properties of \mathbf{x}
which are**

UNCORRELATED with \mathbf{y}
e.g. gender and facial IDs

**How to infer
properties from
observed updates?**



*if adversary has examples
of data with these
properties*

**Use
supervised
learning!**



Property Inference Attacks

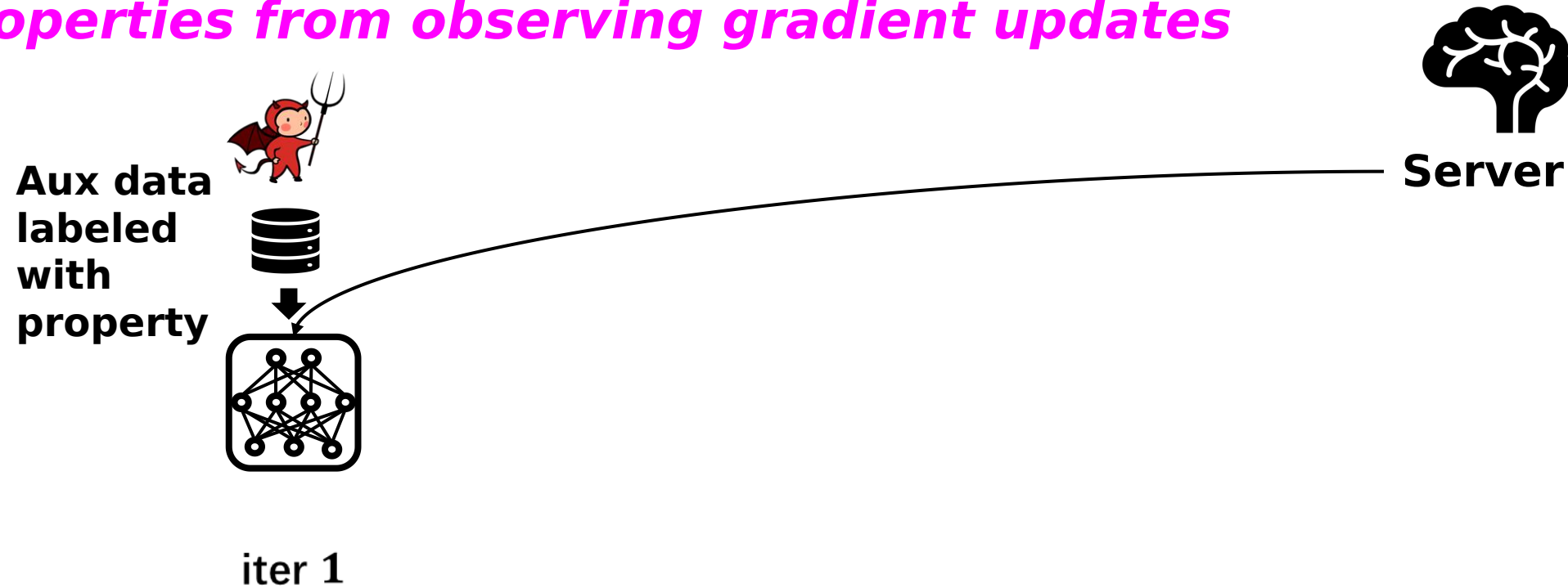
Inferring properties from observing gradient updates

- Assume adversary has auxiliary data consisting of data points sampled from same class as target participant
- Part of the auxiliary data should have the property of D_{prop}^{adv}
- Part of the auxiliary data should **NOT** have the property of $D_{nonprop}^{adv}$
- Use batches of this data to train adversary's local model and update the global model
- For every two consecutive snapshots of the global model, infer the parameter updates (gradients/weights) of all other participants $\Delta\theta_t = \theta_t - \theta_{t-1} = \sum_k \Delta\theta_t^k \rightarrow \Delta\theta_t - \Delta\theta_t^{adv}$
- Label updates as either having property or not (**prop/nonprop**) based on what data the adversary used to train the local model for that round
- Use labeled updates to train a binary classifier for predicting if an update contains the property or not
- Apply this classifier for property inference



Property Inference Attacks

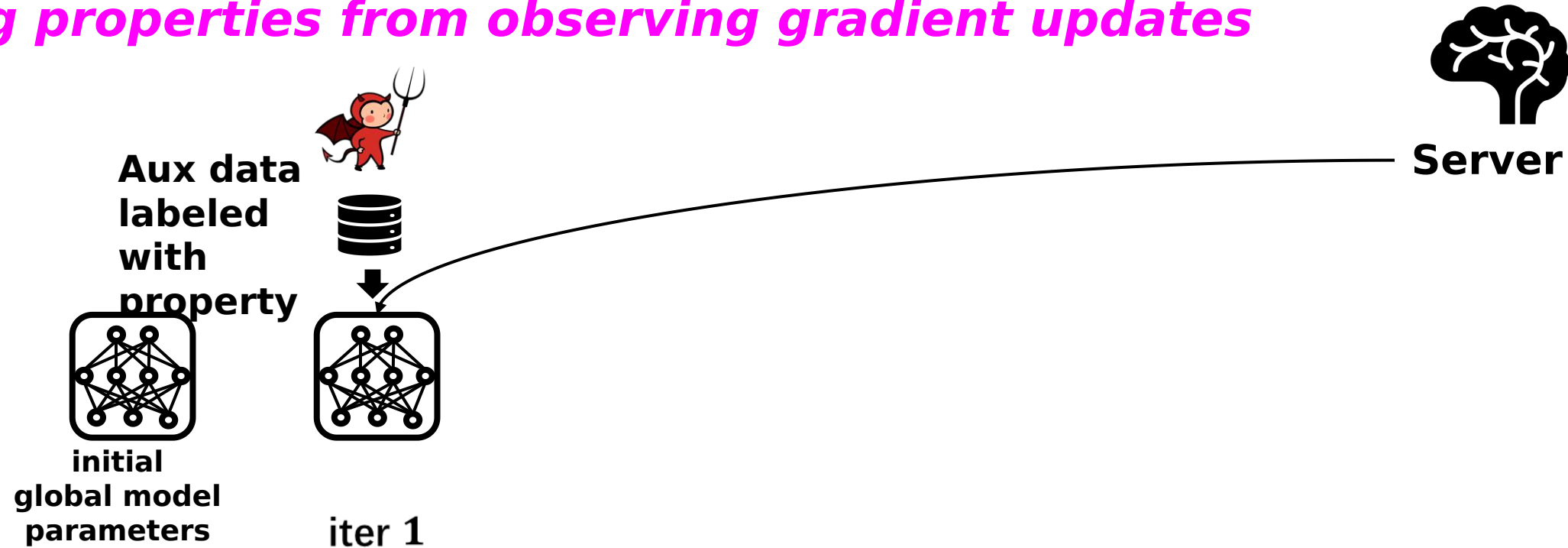
Inferring properties from observing gradient updates





Property Inference Attacks

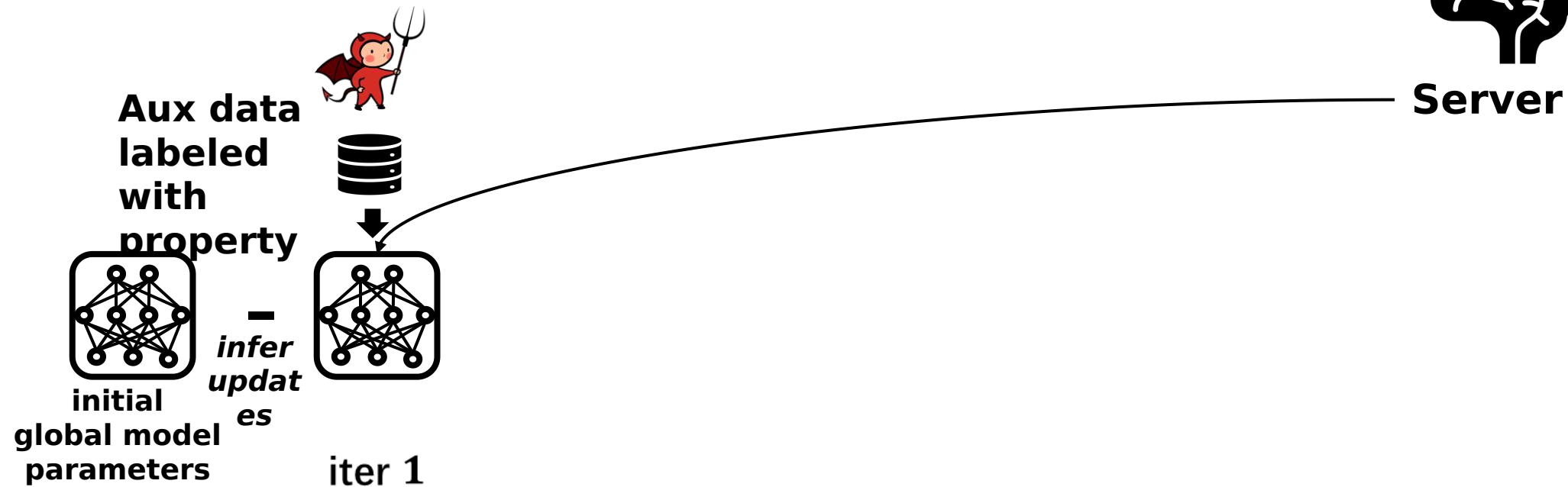
Inferring properties from observing gradient updates





Property Inference Attacks

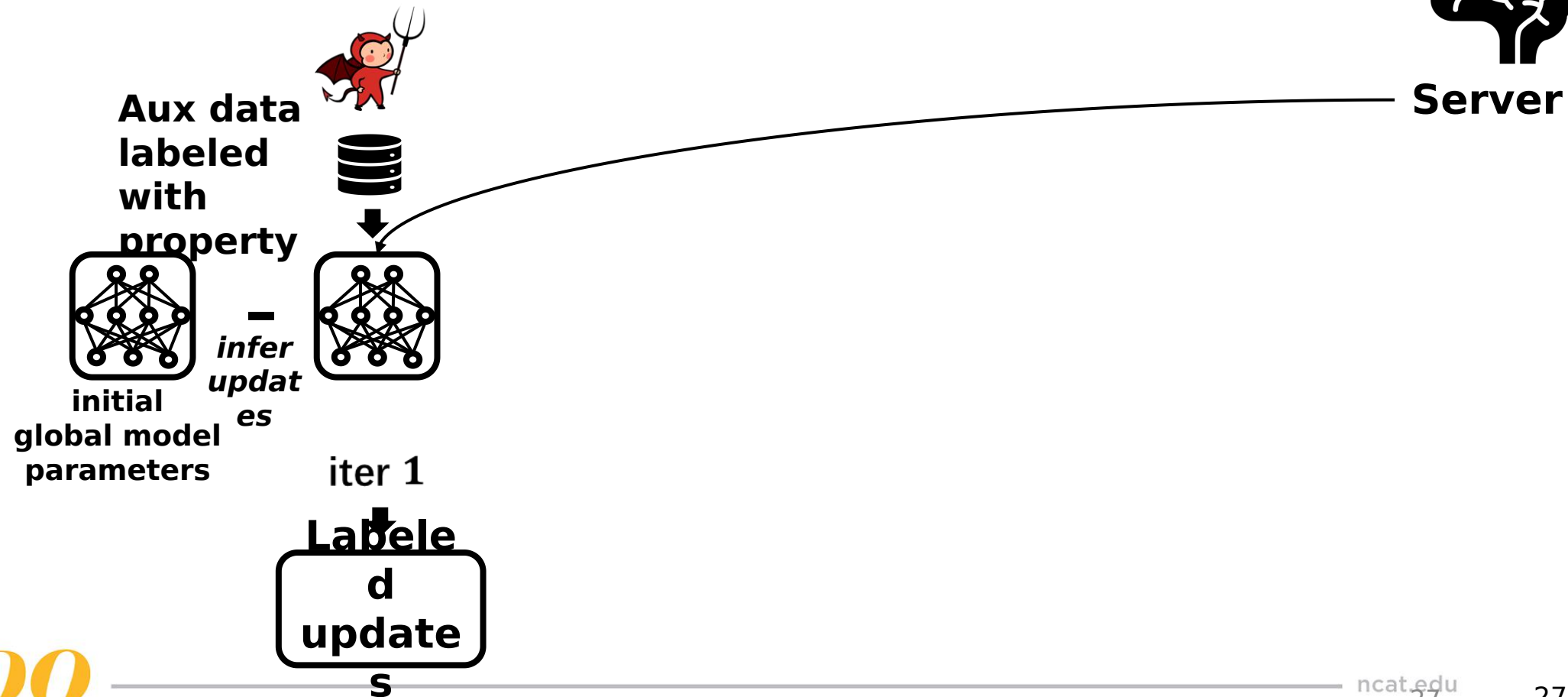
Inferring properties from observing gradient updates





Property Inference Attacks

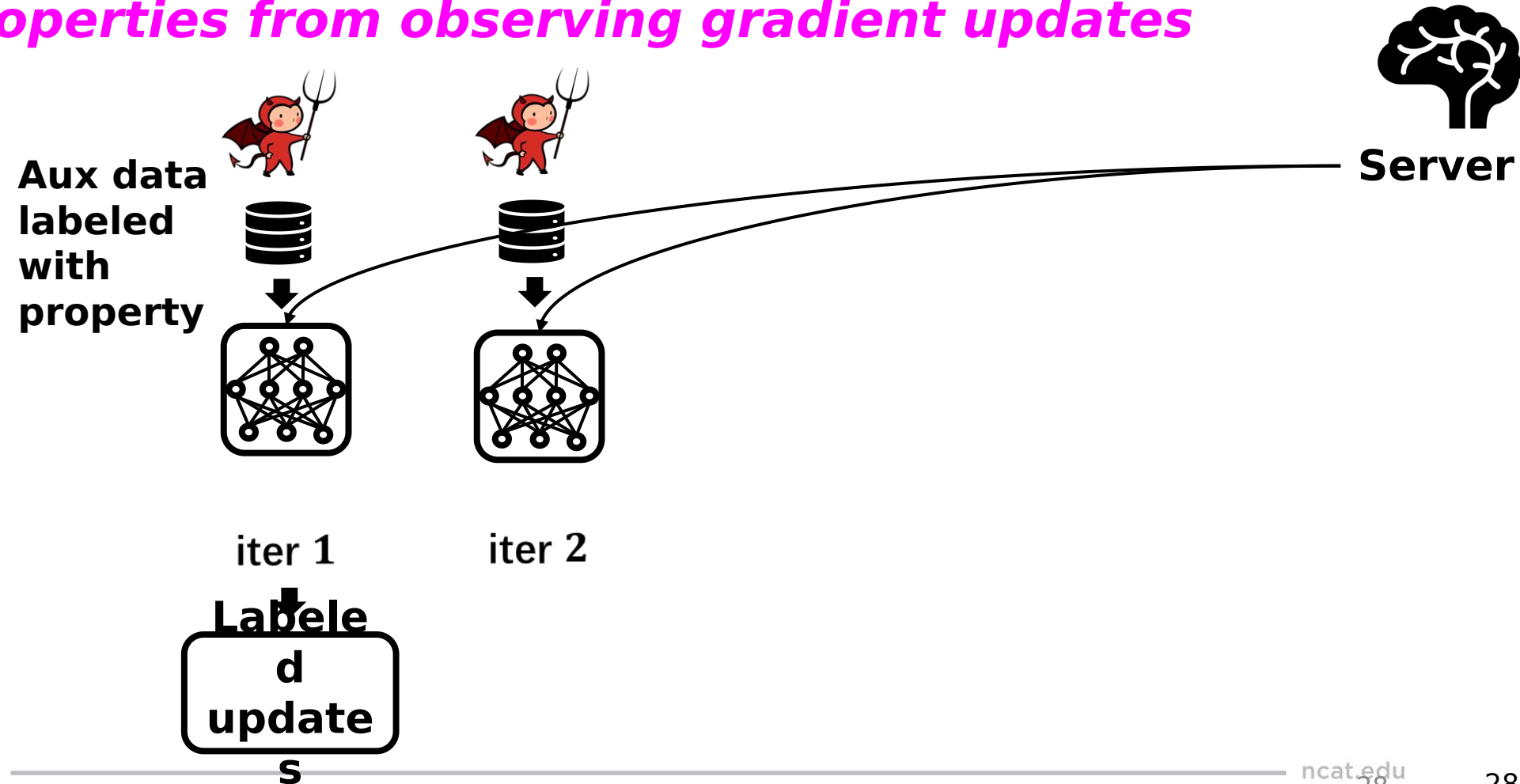
Inferring properties from observing gradient updates





Property Inference Attacks

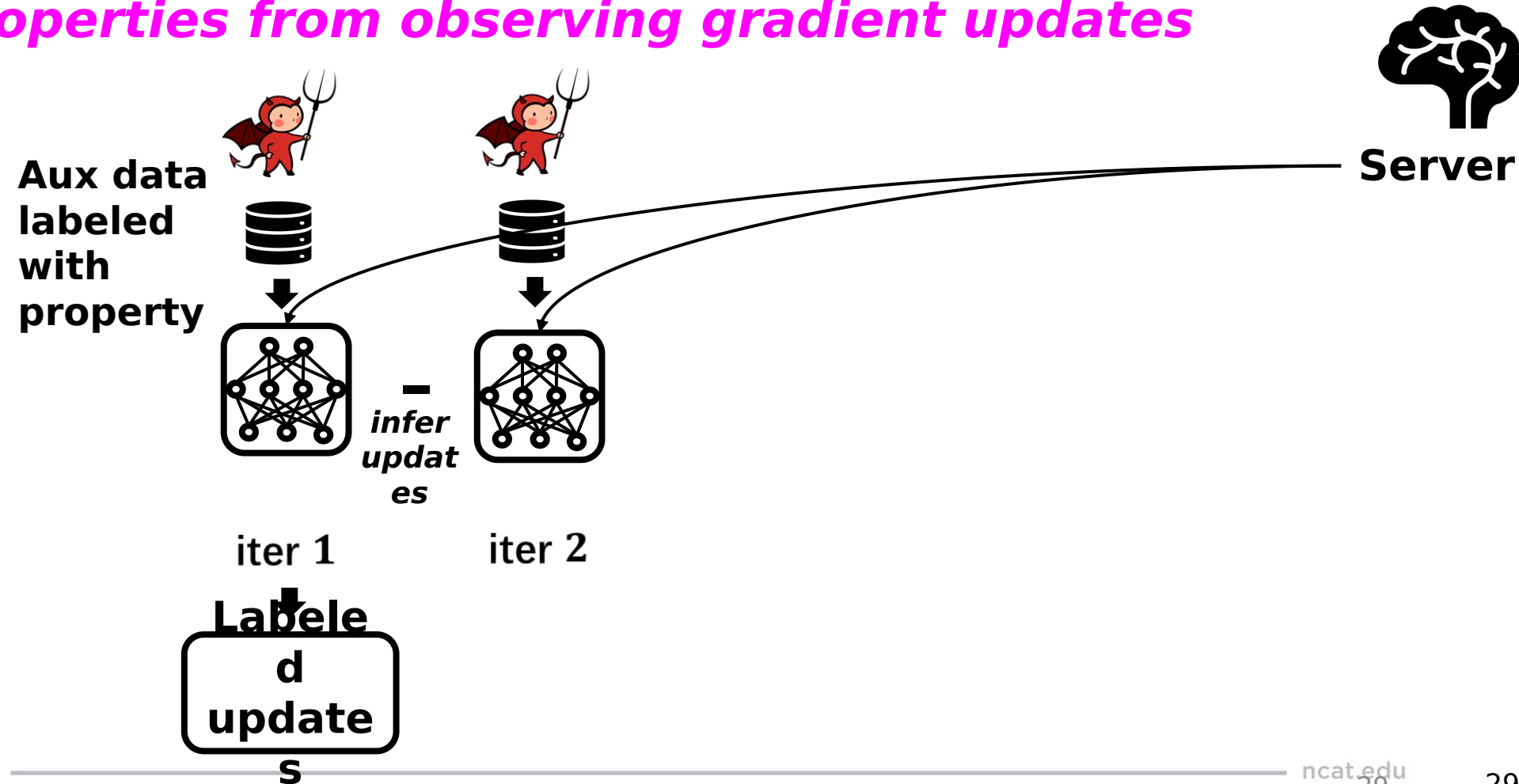
Inferring properties from observing gradient updates





Property Inference Attacks

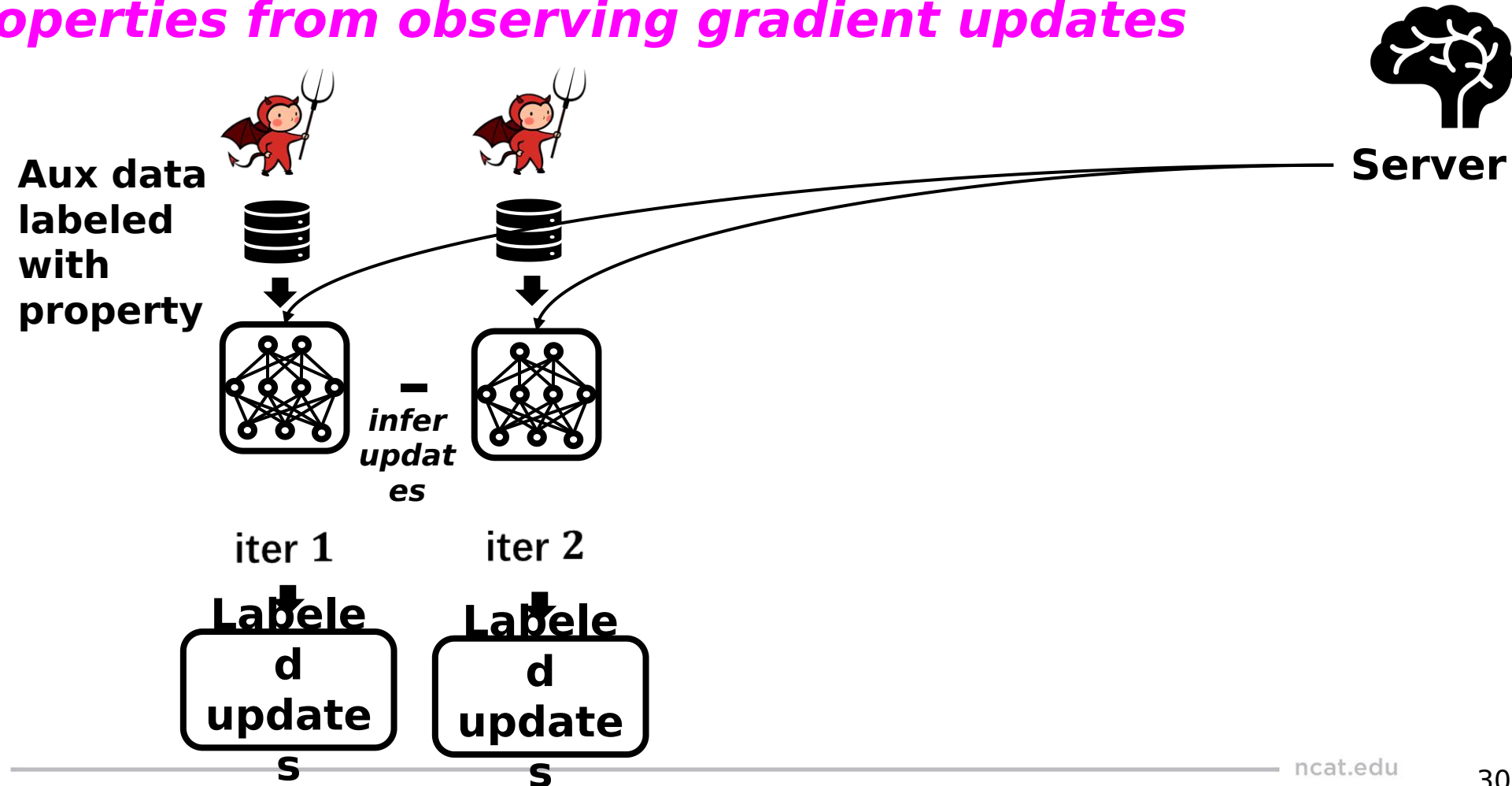
Inferring properties from observing gradient updates





Property Inference Attacks

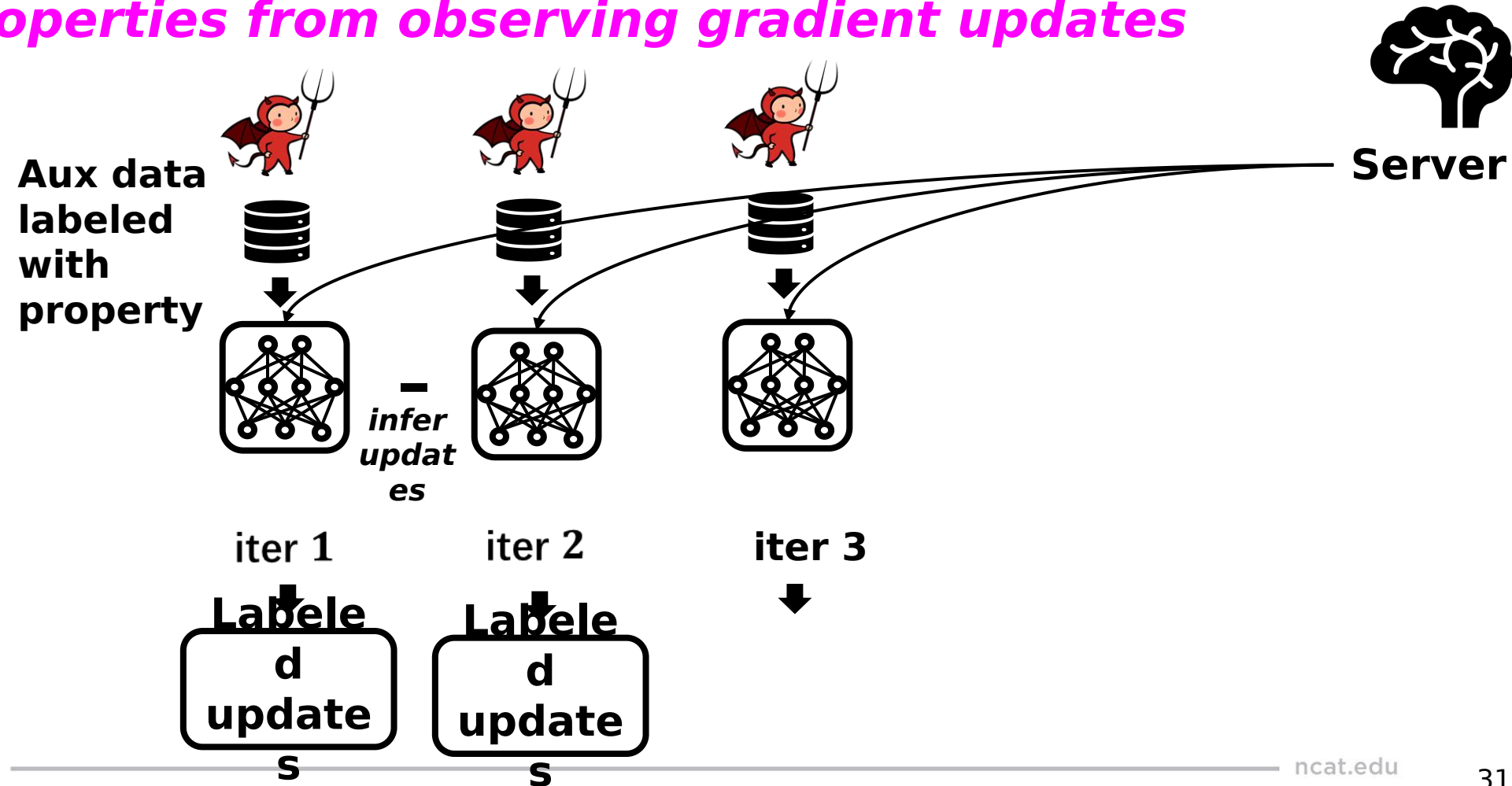
Inferring properties from observing gradient updates





Property Inference Attacks

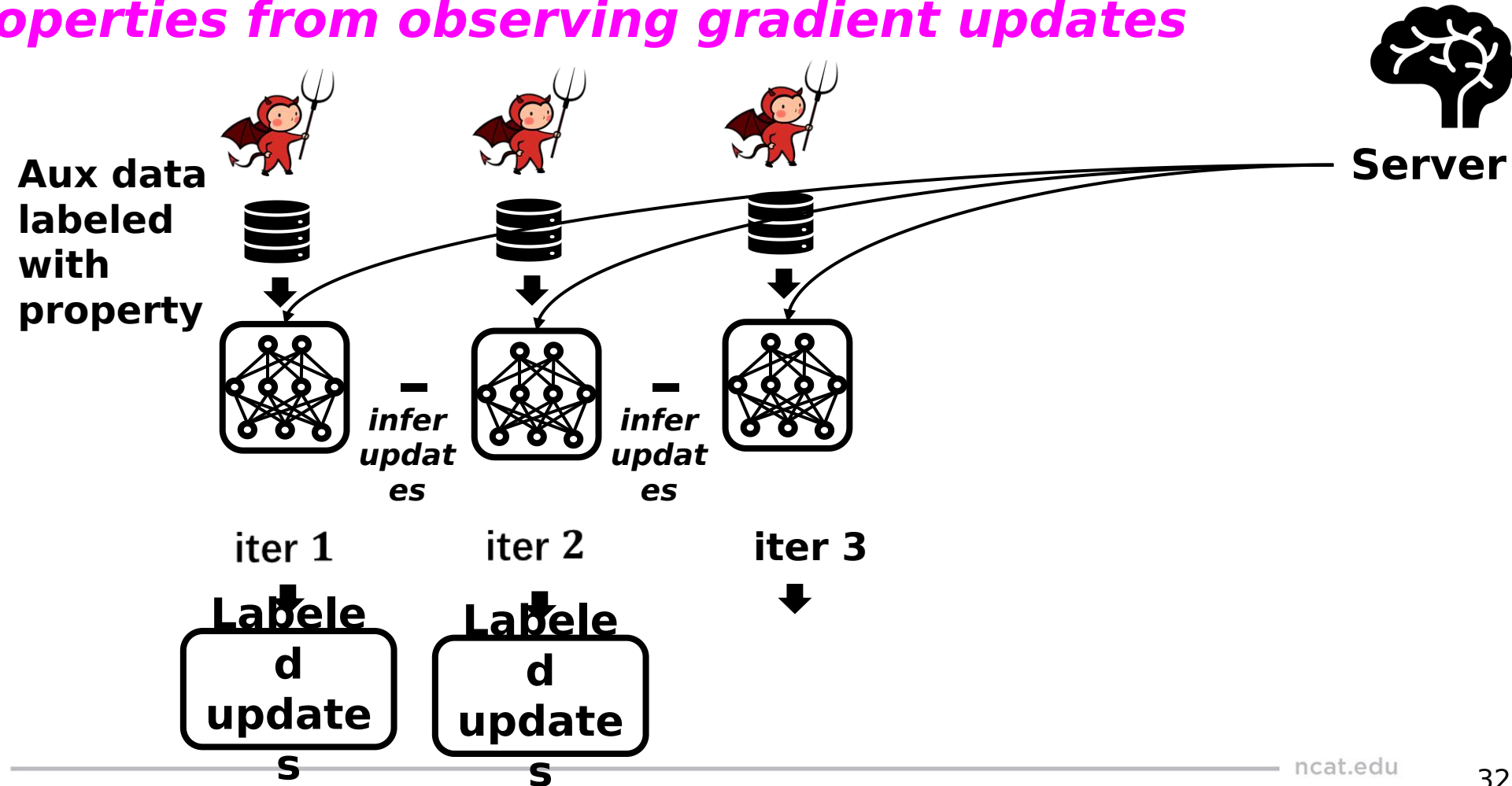
Inferring properties from observing gradient updates





Property Inference Attacks

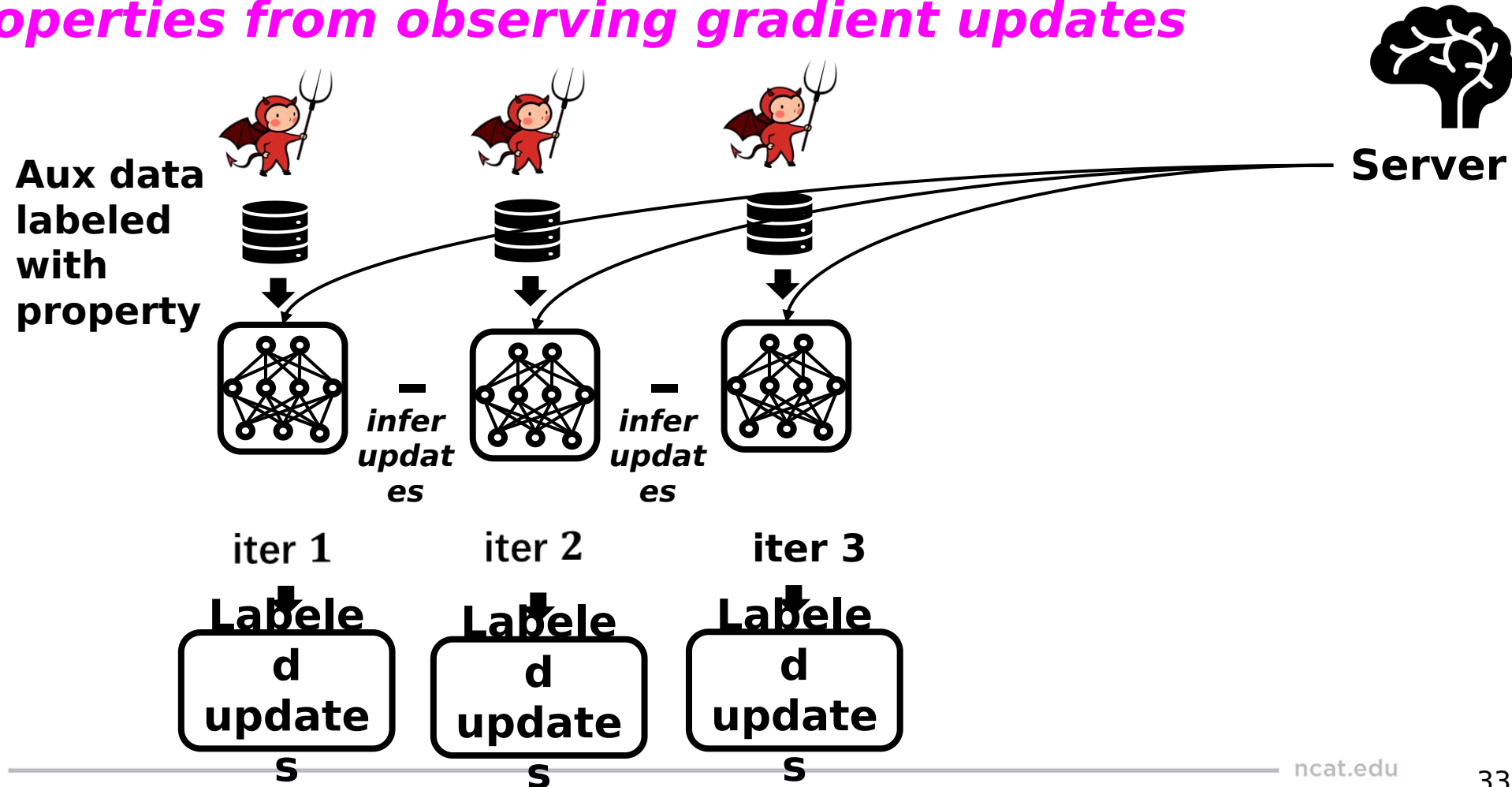
Inferring properties from observing gradient updates





Property Inference Attacks

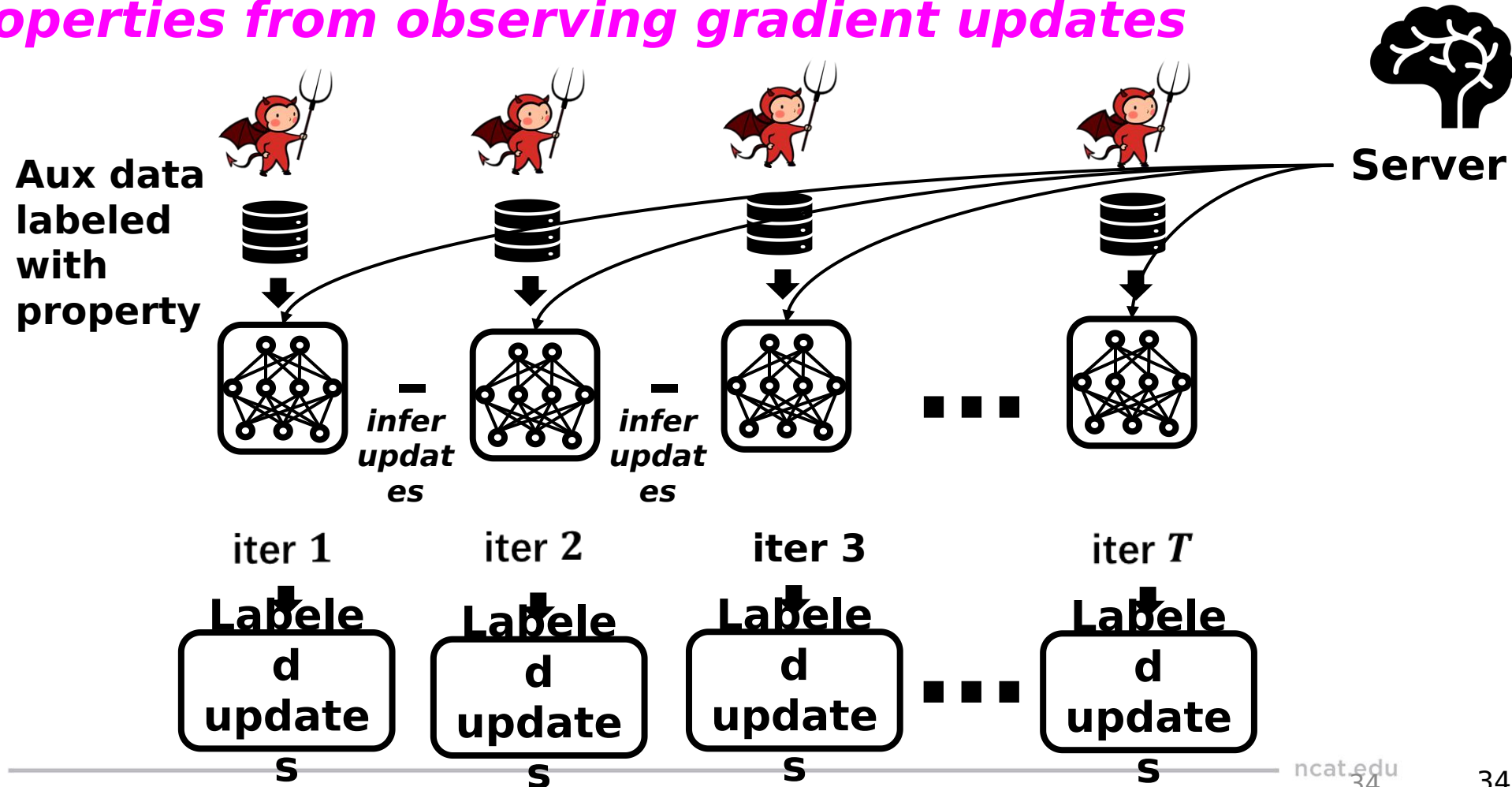
Inferring properties from observing gradient updates





Property Inference Attacks

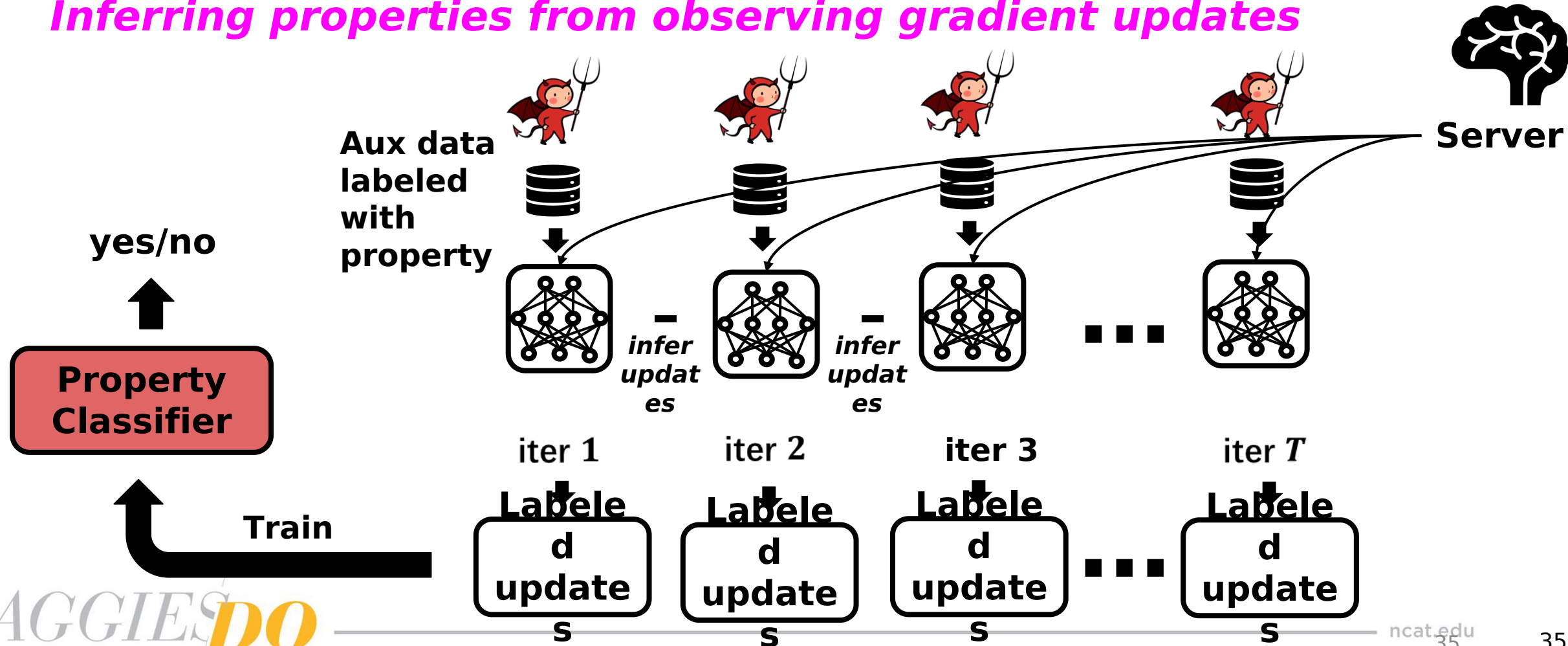
Inferring properties from observing gradient updates





Property Inference Attacks

Inferring properties from observing gradient updates





Property Inference Attacks

Inferring properties from observing gradient 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 $G_{\text{prop}}, G_{\text{nonprop}}$



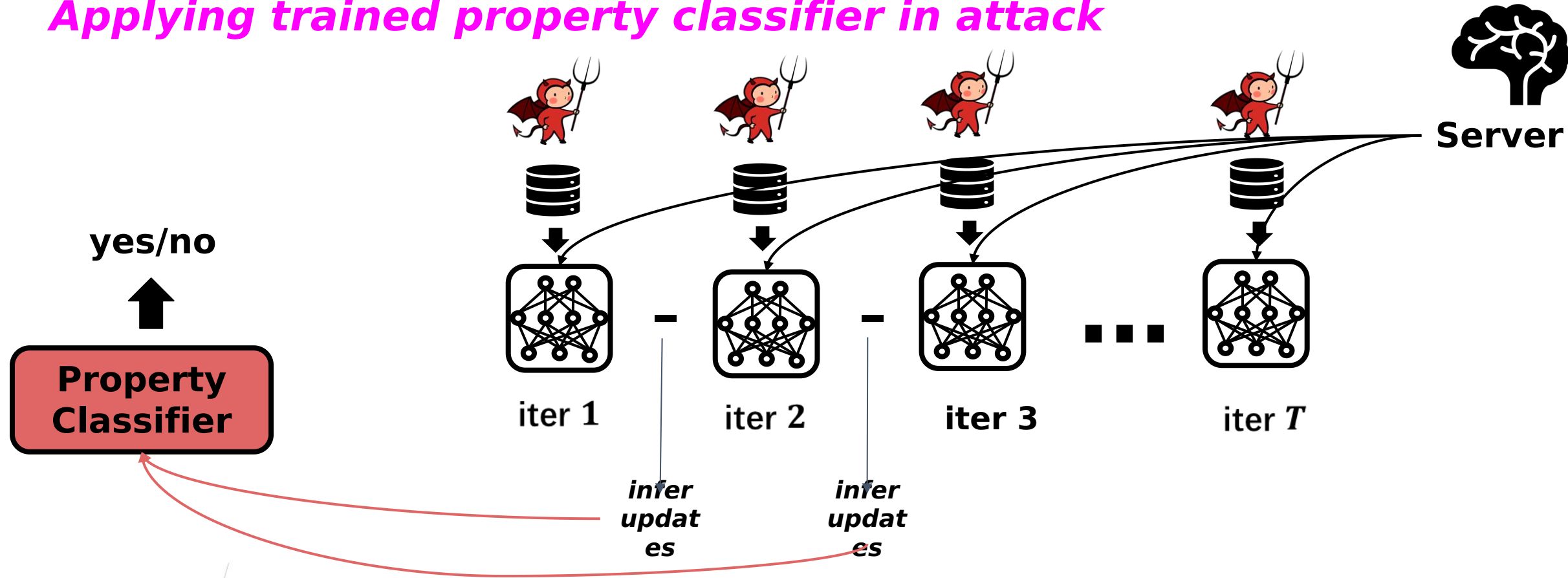
Property Inference Attacks

Applying trained property classifier in attack



Property Inference Attacks

Applying trained property classifier in attack



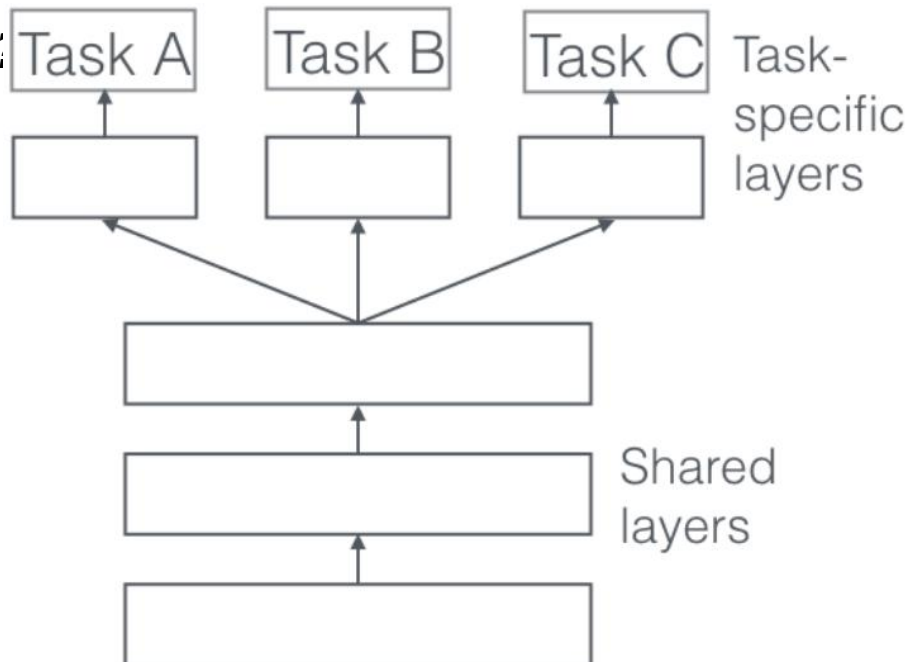


Property Inference Attacks

Active Property Inference

- Uses multi-task learning to make property inference attack more powerful
- A kind of poisoning attack

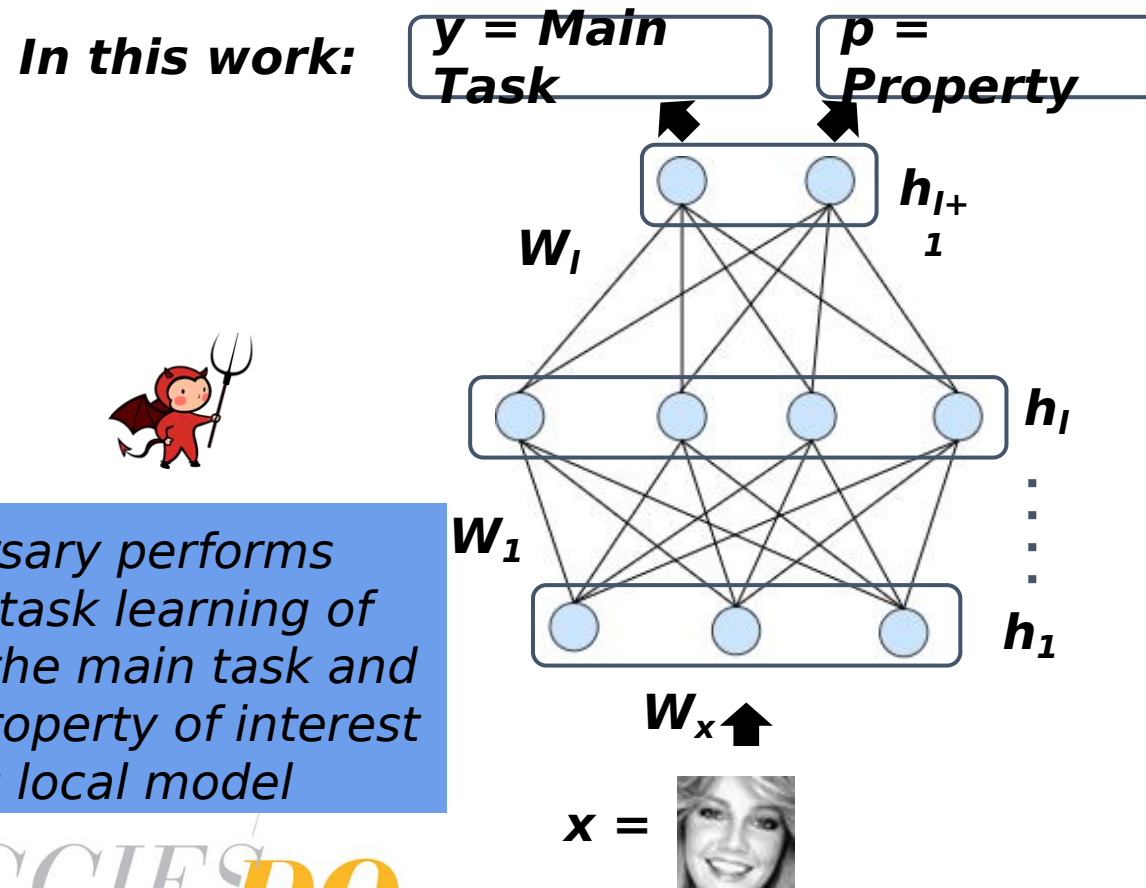
What is multi-task learning:





Property Inference Attacks

Active Property Inference



- Double optimization problem

$$L_{\text{mt}} = \alpha \cdot L(x, y; \theta) + (1 - \alpha) \cdot L(x, p; \theta)$$

- Adversary updates global model with $\nabla_{\theta} L_{\text{mt}}$
- Causes global model to learn separable representations for data with and without the property of interest
- Enhances property inference



Summary of Experiments

Attacks	Model Architecture	
	Two-Party	Multi-Party (4 to 30)
Passive Property Inference	✓	✓
Active Property Inference	✓	
Temporal Inference	✓	✓
Membership Inference	✓	



Datasets

Dataset	Type of Data	# of Records	Main Tasks	Inference Tasks
LFW	images	13.2k	gender/smile/age/ eyewear/race/hair	race/eyewear
FaceScrub	images	18.8k	gender	identity
PIPA	images	18.0k	age	gender
CSI	written essays	1.4k	sentiment	membership/regio n/ gender/veracity
FourSquare	locations	15.5k	gender	membership
Yelp-health	reviews	17.9k	review score	membership/ doctor specialty
Yelp-author	reviews	16.2k	review score	author

Property Classifier Models

- **Conventional ML models** (*not neural networks*)
- **Yelp dataset => Logistic Regression**
- **All other datasets => Random Forest** (*after experimenting with logistic regression, gradient boosting, and random forests*)

Infer Property (Two-Party Labeled Faces in the Wild: Experiments)

Participant trains on facial images with certain attributes

target
label

Property

Correlati
on

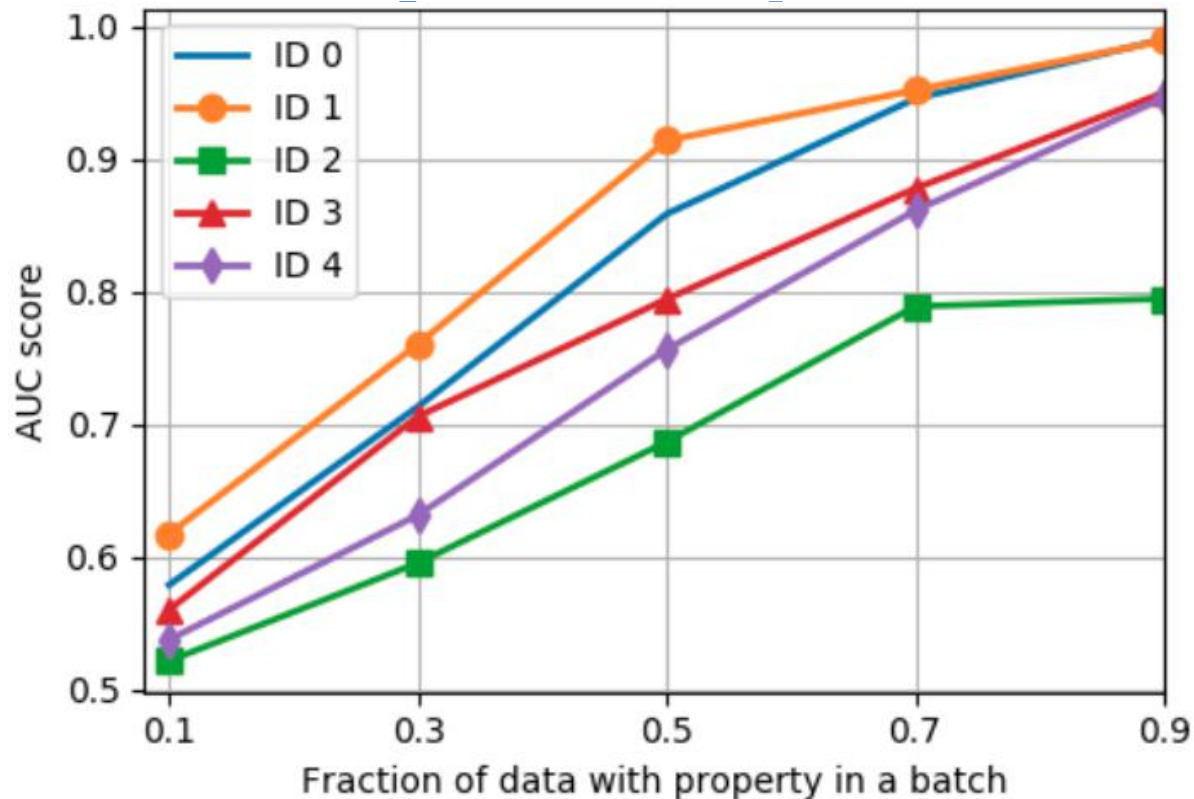
Attack
AUC

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

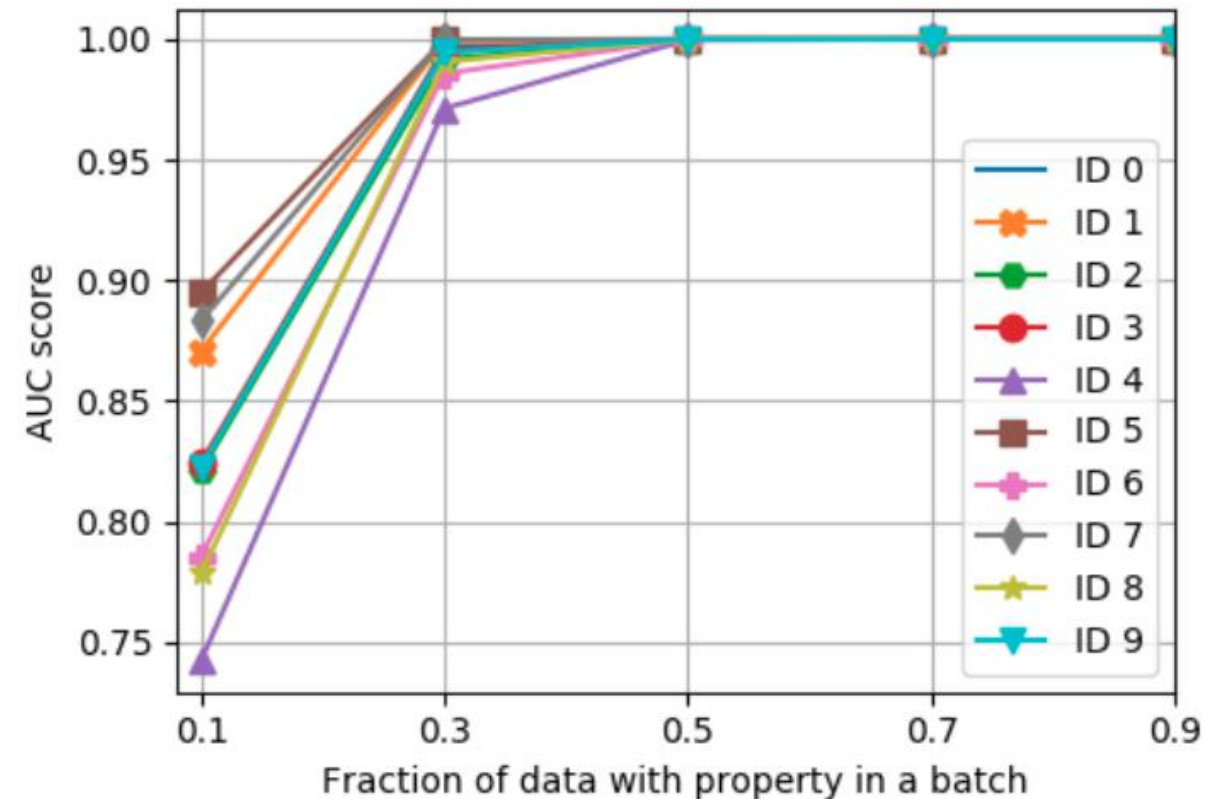
**Main task
and
property
are not
correlated!**



Fractional Property Inference (Two-Party)



(a) FaceScrub



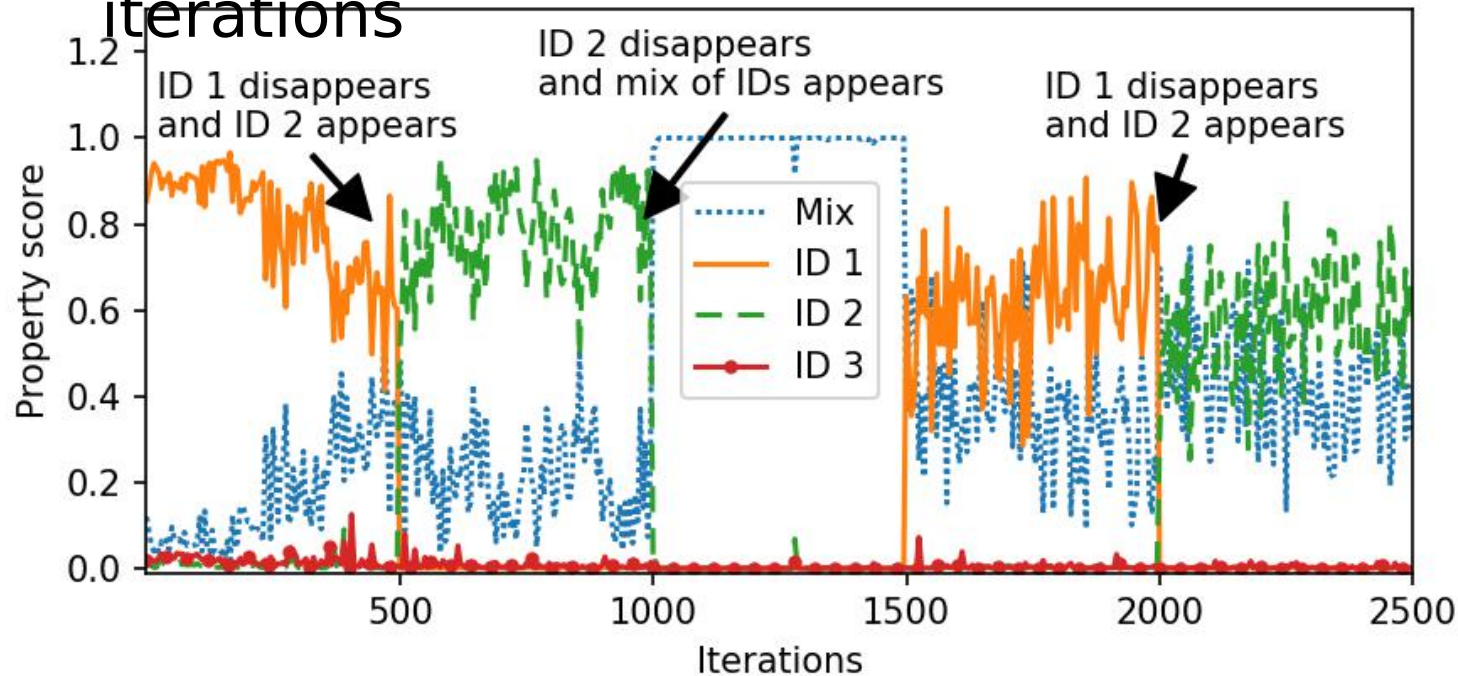
(b) Yelp-author



Infer Occurrence (Two-Party Experiments)

FaceScrub: target=gender, property=facial IDs

Participant trains on faces of different people in different iterations

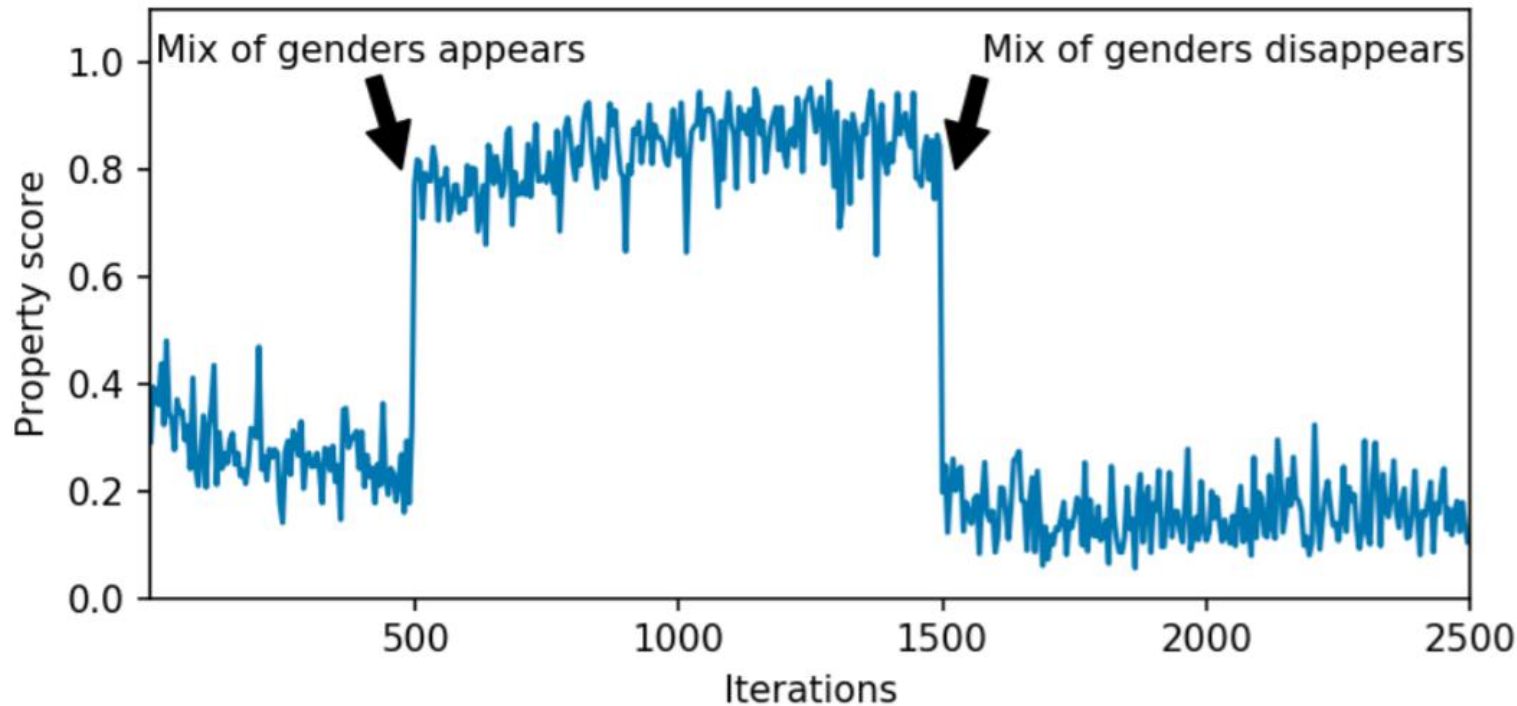


Infer when images of a certain person appear and disappear in training data



Infer Occurrence (Two-Party Experiments)

PIPA: target=age, property=same gender or mixed gender
Participant trains on images with groups of people in different



Infer when images contain people of the same gender or mixed gender

Active Attack (Two-Party Experiments)

FaceScrub: target=gender,
property=facial IDs

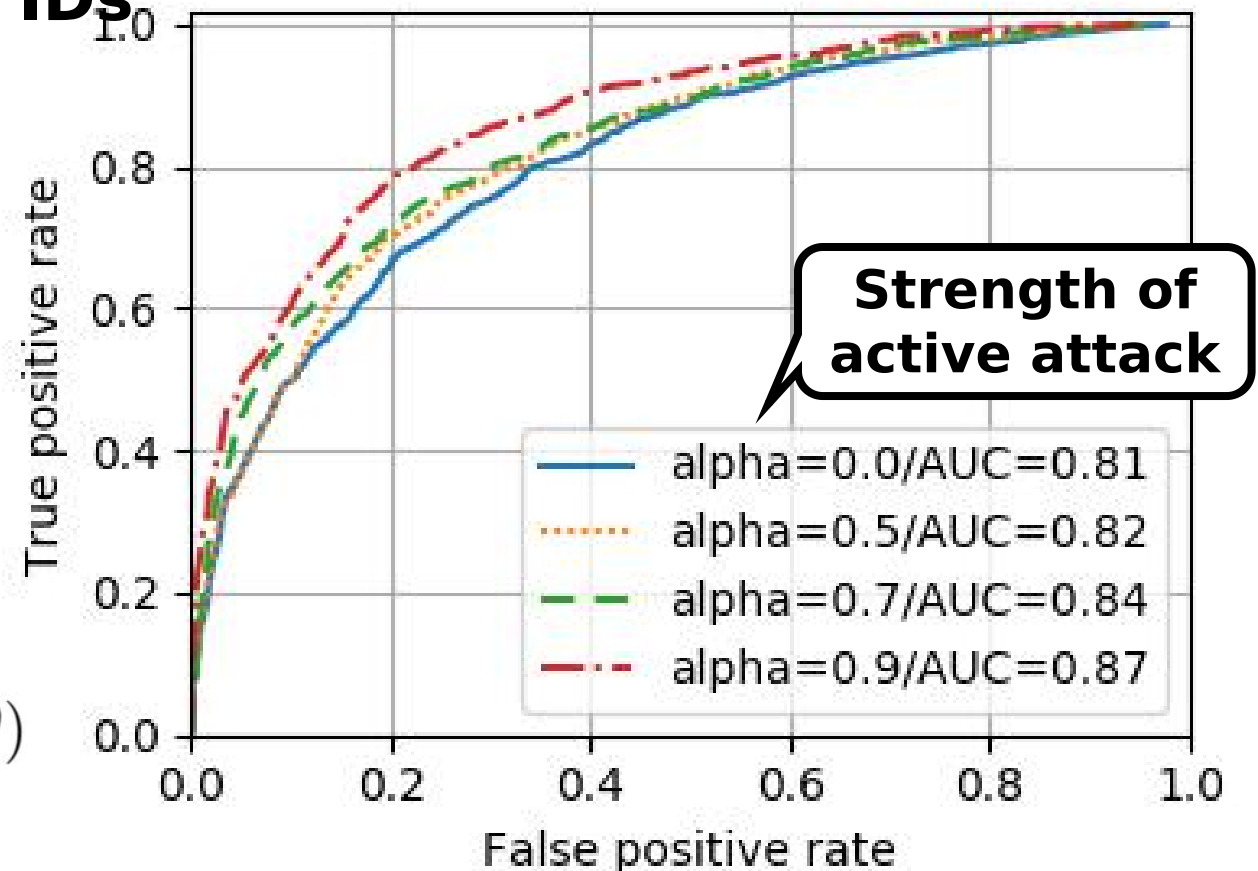
Adversary uses multi-task learning to create a model that

- Predicts task label
- **Predicts property**

Adversary can actively bias the model to leak property by sending crafted updates!

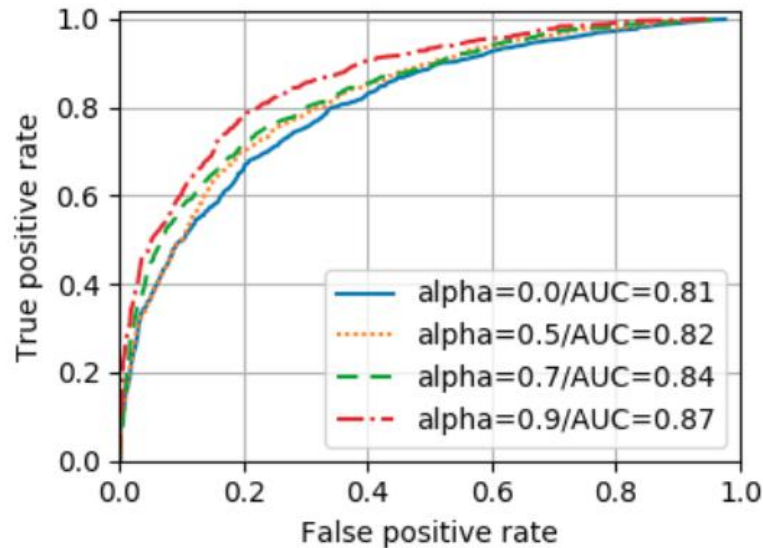
$$L_{mt} = \alpha \cdot L(x, y; \theta) + (1 - \alpha) \cdot L(x, p; \theta)$$

AGGIES **DO**

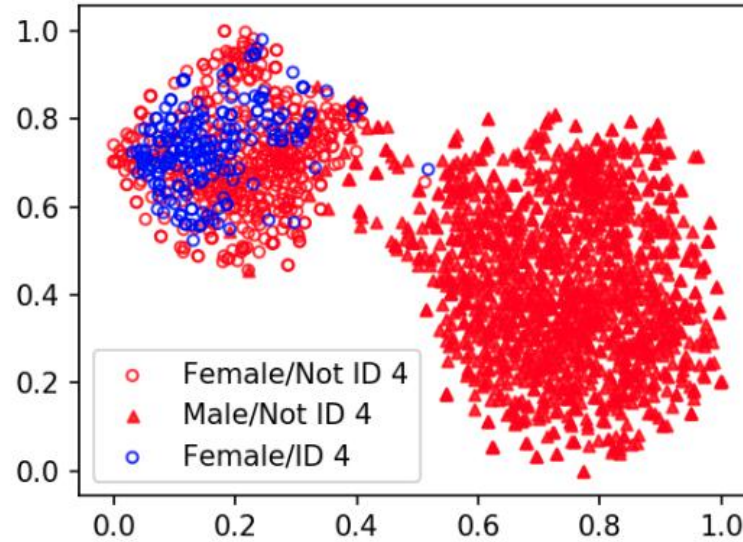


Active Attack (Two-Party Experiments)

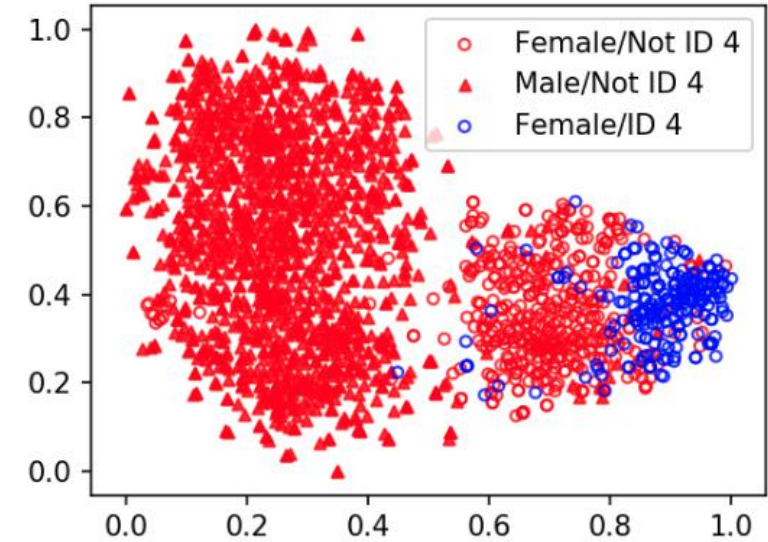
FaceScrub: target=gender,
property=facial IDs



(a) ROC for different α



(b) t-SNE of the final layer for $\alpha = 0$



(c) t-SNE of the final layer for $\alpha = 0.7$.



Membership Inference (Two-Party Experiments)

Yelp-health: target=review score,

- Create a unique test consisting of a group of features that represents unique sample/member you want to infer (test bag of words (BoW)).

Example:



Sample: {1, 0, 0, 0, ..., 1, 1,

1}

- After each batch of training infer all the features present (non-zero gradients in embedding layer) in the batch (batch BoW)
- If test BoW is a subset of batch BoW, then the test BoW has been inferred as a member used in the current batch of training

Yelp-health	
Batch Size	Precision
32	0.92
64	0.84
128	0.75
256	0.66
512	0.62



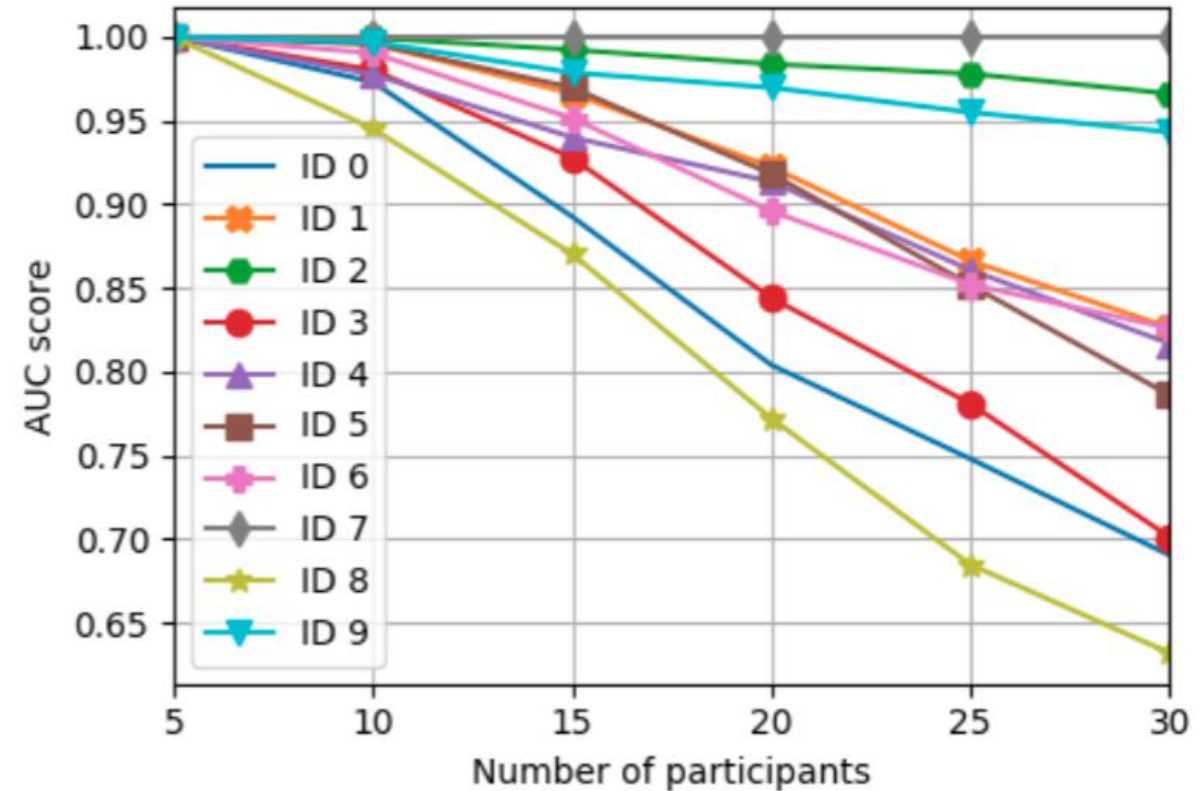
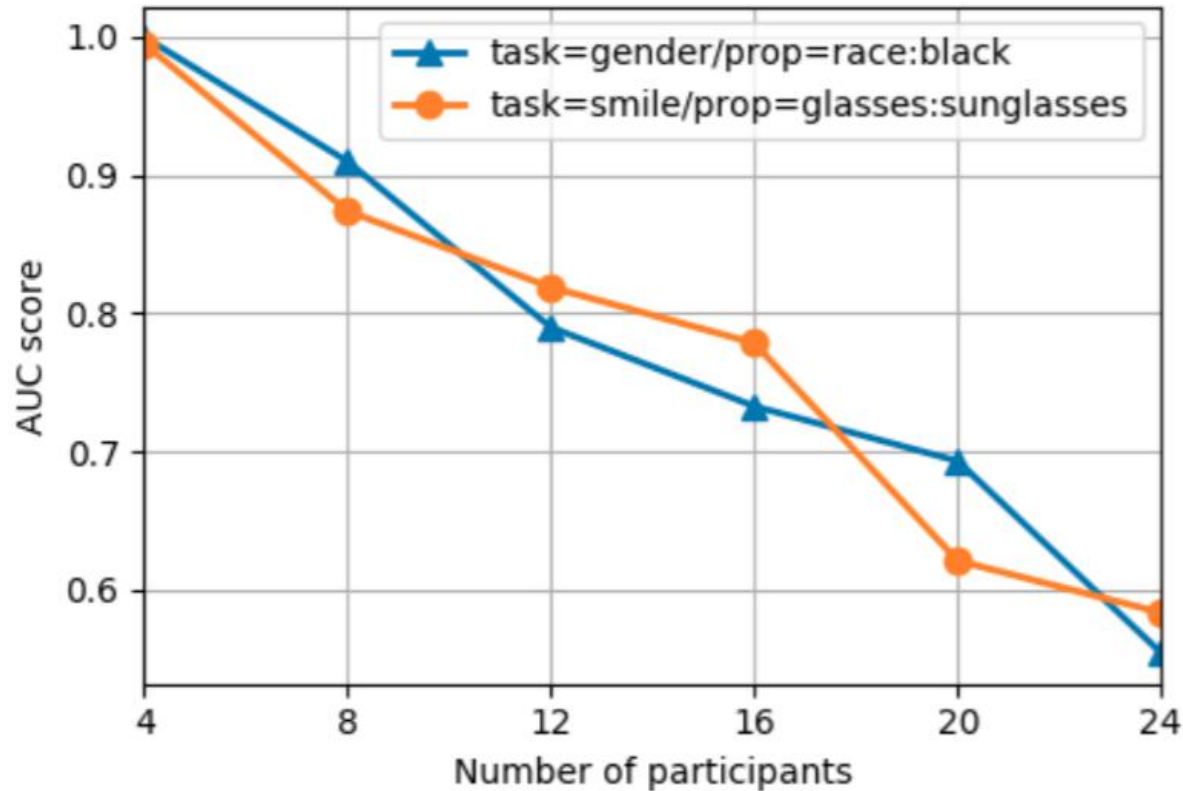
Membership Inference (Two-Party Experiments)

FourSquare: target=gender,
inference=r

FourSquare	
Batch Size	Precision
100	0.99
200	0.98
500	0.91
1,000	0.76
2,000	0.62

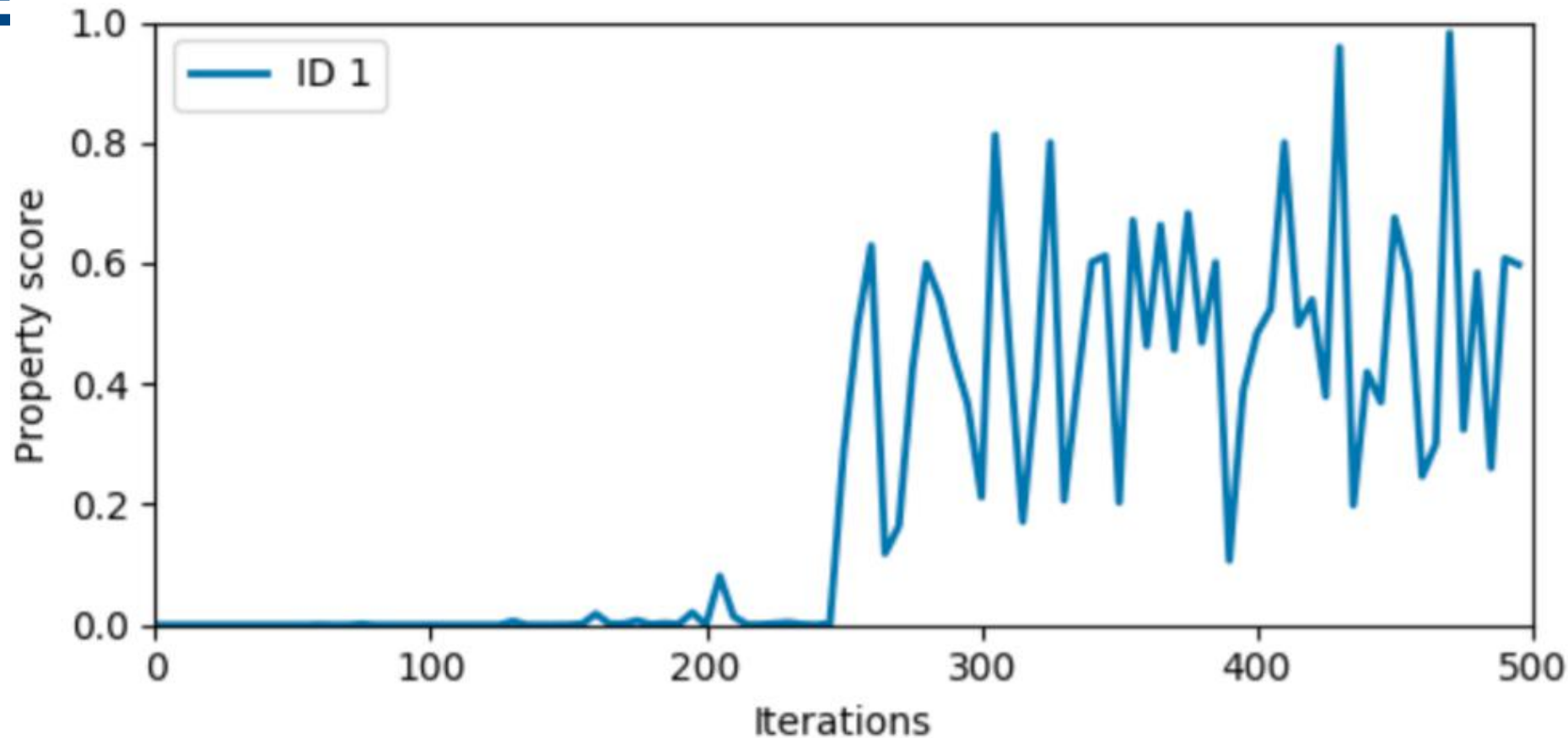


Property Inference (Multi-Party Experiments)





Infer Occurrence (Multi-Party E





Code Demonstration

Code Structure

Load Data

Split Data among participants

Perform collaborative learning
with adversary using aux data
to label updates

Use labeled updates to train
and test property classifier

Libraries

- Theano for deep learning
- Lasagne for deep learning
- Scikit_Learn for conventional ML models (Property classifier)

Dataset and Model Architecture

- LFW
- Two-Party
- iterations: 3000
- learn rate: 0.01


Code Demonstration

[Code](#) [Issues 6](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security 0](#) [Insights](#)

Code for Exploiting Unintended Feature Leakage in Collaborative Learning (in Oakland 2019)

[5 commits](#) [1 branch](#) [0 packages](#) [0 releases](#) [2 contributors](#)

[Branch: master](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

 **csong27** Update README.md

Latest commit 7600988 on May 28, 2019

.gitignore	update	17 months ago
README.md	Update README.md	11 months ago
__init__.py	push	17 months ago
distributed_sgd.py	update	17 months ago
inference_attack.py	update	17 months ago
load_lfw.py	update	17 months ago
split_data.py	update	17 months ago



Code Demonstration

1 # LFW Attribute Values v1.2 - lfw_attributes.txt - <http://www.cs.columbia.edu/CAVE/projects/faceverification>
2 # person imagenum Male Asian White Black Baby Child Youth Middle Aged Senior Black Hair Blond Hair Brown Hair Bald No Eyewear
Eyeglasses Sunglasses Mustache Smiling Frowning Chubby Blurry Harsh Lighting Flash Soft Lighting Outdoor Curly Hair Wavy Hair Straight Hair Receding
Hairline Bangs Sideburns Fully Visible Forehead Partially Visible Forehead Obstructed Forehead Bushy Eyebrows Arched Eyebrows Narrow Eyes Eyes Open Big
Nose Pointy Nose Big Lips Mouth Closed Mouth Slightly Open Mouth Wide Open Teeth Not Visible No Beard Goatee Round Jaw Double Chin Wearing Hat Oval
Face Square Face Round Face Color Photo Posed Photo Attractive Man Attractive Woman Indian Gray Hair Bags Under Eyes Heavy Makeup Rosy Cheeks Shiny
Skin Pale Skin 5 o' Clock Shadow Strong Nose-Mouth Lines Wearing Lipstick Flushed Face High Cheekbones Brown Eyes Wearing Earrings Wearing Necktie Wearing
Necklace

3 Aaron Eckhart 1 1.56834639173 -1.88904271738
1.73720324618 -0.929728671614 -1.4717994909 -0.195580416696 -0.835609388667 -0.351468332141 -1.01253348522 -0.719593319061 -0.632400663502 0.464839153939 -0.973528328799
1.56518551138 -1.29670421719 -1.54271878921 -0.684671060805 -0.864989670524 0.76688573774 -0.218952102857 -1.65566546684 -0.787043915291 -0.599664927461 0.458518580099 0.1897596683
0.851554669872 -0.385720388897 -0.497719222187 -0.161149044729 -0.25751432601 -0.0888388089788 0.455468790136 -0.839211431403 -0.0229481172569 -0.922567662796 -0.114538586108
1.46122165091 1.75848095942 0.0688935153644 1.26785977543 -1.12024419679 0.917616524376 -1.30795658065 -1.50041333023 1.02922066901 0.832362932111 -0.498656998427
0.251364993624 -0.705281306212 -0.515715482239 0.374239188976 -0.168674595709 -0.614143271487 3.09770263624 1.52385816838
0.779277999669 -0.0714539213939 -1.24648342154 -0.76928347674 -0.725596699772 -1.82061027862 -2.07297656641 -0.960758740847 0.361737685257
1.16611821063 -1.16491625494 -1.13999038432 -2.37174572455 -1.29993198905 -0.414681760268 -1.1449020909 0.694007237055 -0.826608788807

4 Aaron Guil 1 0.169850615079 -0.9824078298
0.422709344724 -1.28218444066 -1.36005999796 -0.867001510546 -0.45229265405 -0.197520738279 -0.956073046658 -0.802106525403 -0.736883019349 0.294554304216 -1.27764713376
0.95477088461 -0.990991854047 -1.16735850503 -0.83514604497 0.798544268921 -0.971678536001 0.342825883931 -1.32256184017 0.962937279485 -1.19936329824 -0.157306858225
0.443223864039 -0.00288155806399 -0.0211583900293 -0.226562576584 -0.0810385892617 -0.827201916484 -0.106624294025 1.22759371328 -0.812223054077 -1.24125787846
0.0962724582428 -0.4045435044 0.32591852997 0.474452358604 1.13535949237 0.0587247173045 0.611175959505 -1.17251028284 0.428512003215 -0.874235053954 -1.19156451444
0.192359075355 -0.204165914866 0.342347000895 0.239512219774 -1.47469040233 0.236057105309 -0.565208399216 -0.712541538523
2.99707627363 -0.273305759119 -0.187721706702 -0.604608482776 -1.32170093568 -0.938558989147 0.494294491446 -0.659043168968 -1.14374681565 -0.775721833113 -0.832036380098 -0.397680027246
0.874160103001 -0.945431057978 -0.268648623951 -0.00624408064799 -0.0304056925377 -0.480128381674 0.666759772228 -0.496558800435
5 Aaron Patterson 1 0.997748978625 -1.36419463748 -0.157376927297 -0.756447251994 -1.89182505036 -0.87152602607 -0.862893308853
0.0314446531456 -1.34152295494 -0.0900374885122 -1.20072546722 -0.332460195946 -0.537006417334 1.29836399866 -1.49847119745 -1.28582334941 1.14174165431 0.172817484002
0.106412089631 -0.788843002003 0.349295353266 -1.64371594869 0.454287433947 1.18945756037 -0.688414064597 -0.590574328946 -0.266672886189 0.467224077346 0.567348333084 -1.71910100907
0.124666691797 -1.60274145814 -0.659399105992 -1.7537616053 1.20447343826 0.0221883756845 -1.13544276804 1.70285701537 -0.422143801135 0.587859199153 0.414362867222
0.344447478056 -1.26045130216 -0.577746346924 0.4055670773834 -1.91654528379 0.921260295304 0.247436704738 -0.4284513959 -0.77227338357 0.370673075034 -0.509596298842 -0.768481995141
1.70689701019 0.126523976738 -0.497001028198 -0.393041978339 -0.178306935845 -1.1802267439 -0.596914490833 -1.80538247092 -0.951643406892 -0.838087417542 1.54974268112
1.88474515371 -0.999765023736 -1.3598581042 -1.91210796401 -1.09563421851 0.915125965207 -0.572332382954 0.144261972973 -0.84123127649
6 Aaron Peirsol 1 1.12271853446 -1.99779909564 1.91614437179 -2.51421429402 -2.58007139867 -1.40423935631 0.057551079477
0.000195881567807 -1.27351176256 -1.43146224608 -0.0705187622747 -0.33923864402 -2.00414944689 0.665694950342 -0.775940385642 -1.47162908339 -1.17907991578
0.563327280416 -0.664428541937 -1.40792813233 0.435594119792 -0.589987923681 -1.60349837846 1.17074082314 0.760103210296 0.211497967133 -0.516180321472 -1.33114623282 0.202839683836
0.149644776334 -0.046429612354 0.640885073388 -0.107615735848 -0.831271054612 -0.827004573553 -0.588724809097 0.429254705025 1.58766441509
0.499086141956 -0.0568691636225 -0.866642909756 -0.95968887997 0.350729937273 -1.33535414862 -0.427889677878 0.826817153515 -0.256779421679
0.149751104659 -1.20153137751 -1.08391687413 0.255363468848 -0.650423019316 -0.506292732963 1.10159231815 0.64078323912 1.57502827776 -0.484396724644 -1.55968231935 -1.43712369413
0.37936300285 -0.648233451093 -2.25735171759 -1.07561269561 0.567822023847 -0.176088957461 1.10812479108 -1.60094409268 3.26461275672 0.813418335935 0.30863081628 -0.848693270575
0.475941175723 -0.447025051151

7 Aaron Peirsol 2 1.07821423781 -2.00809831161 1.67621103655 -2.2780559446 -2.65184543714 -1.34840776272 0.649089348664
0.0176564027753 -1.88911117008 -1.85721274169 -0.568056876713 0.840375172105 -1.98126920929 1.66671001116 -0.910723410402 -1.99350933975 -0.871334969156
0.507786460915 -0.488946636787 -0.886489888727 -0.990131832196 -0.75081327101 -0.378478666175 0.583085632685 -1.47960267898
0.250184727343 -0.38112304548 -0.61199103242 -0.143090577304 -1.07275988807 0.43209396266 1.08919285242 -0.470928661825 -1.17712410289 -0.111312751227 -0.154602526203 -1.0315082548
2.39245818882 -0.19157645761 1.2279032724 -1.38179742097 -1.52885140882 0.90796350999 -1.32428788117 -0.934644359205 0.686994692207 -0.149300693668
0.0336262689499 -0.911137660105 -1.24109244095 0.904176737805 -0.309967224321 -1.03889207091 3.75811845298 1.05836521801
1.50214861782 -0.649714863778 -1.07294353889 -1.7783162834 -0.743271427817 -3.30070920476 -0.7792193783 -1.46147431539 -0.955282684761
0.119113361315 -1.12817571921 -3.16104814248 0.0826804119045 -0.439613851533 -0.359859014792 -0.760773908824 -0.410151910322

Countermeasures

Sharing fewer gradients

- Theory: the fewer the parameters/gradients shared by participants, the less information can be leaked
- Demonstrated in [1]
- Comes with the cost of reduced accuracy of the global model

**Property inference attacks against CSI dataset.
Main Task: sentiment**

Property / % parameters update	10%	50%	100%
Top region (Antwerpen)	0.84	0.86	0.93
Gender	0.90	0.91	0.93
Veracity	0.94	0.99	0.99

**** does not show accuracy of actual model for the main task***

Countermeasures

Dimensionality Reduction

- Theory: for sparse input space, use a smaller subset of the larger input space (e.g the most frequent features). Smaller input space should leak less information
- Comes with the cost of reduced accuracy of the global model

Membership inference attacks against CSI and FourSquare datasets.

Main Task: gender

CSI			FourSquare		
Top N words	Attack Precision	Model AUC	Top N locations	Attack Precision	Model AUC
4,000	0.94	0.91	30,000	0.91	0.64
2,000	0.92	0.87	10,000	0.86	0.59
1,000	0.92	0.85	3,000	0.65	0.51
500	0.82	0.84	1,000	0.52	0.50

Countermeasures

Dropout

Property inference attacks against CSI dataset.
Main Task: sentiment

- Theory: regularization technique to reduce overfitting by randomly deactivating links between neurons
- Reduces number of gradients
- Comes with the cost of reduced accuracy of the global model

Dropout Prob.	Attack AUC	Model AUC
0.1	0.94	0.87
0.3	0.97	0.87
0.5	0.98	0.87
0.7	0.99	0.86
0.9	0.99	0.84

Countermeasures

Participant or Record Level Differential Privacy (DP)

- Theory: add random noise to participant datasets/individual samples to make reverse engineering difficult
- Comes with the cost of reduce accuracy of joint model
- Record level DP can prevent membership inference, but not individual property inference
- Participant level DP can limit the success of all attacks, but it needs a large number of participants for training to converge and get a reasonable accuracy [2]

Countermeasures

Artificial Multiplication of Participants

- Theory: multi-party property inference attack accuracy reduces as number of participants increase.
- Multiply number of participants even if not needed;
- Split data across participants where possible

Limitations

- Availability of auxiliary data (some may be easier to find than others)
- Number of participants
- Undetectable properties
- Attribution of inferred properties (trivial in two-party; not possible in multi-party)

Related Work

- Orekondy, T., Schiele, B., & Fritz, M. Gradient-Leaks: Understanding and Controlling Deanononymization in Federated Learning.
- B. Hitaj, G. Ateniese, and F. P´erez-Cruz. Deep models under the GAN: Information leakage from collaborative deep learning. In CCS , 2017.

Related Work

B. Hitaj, G. Ateniese, and F. Pérez-Cruz. Deep models under the GAN: Information leakage from collaborative deep learning.

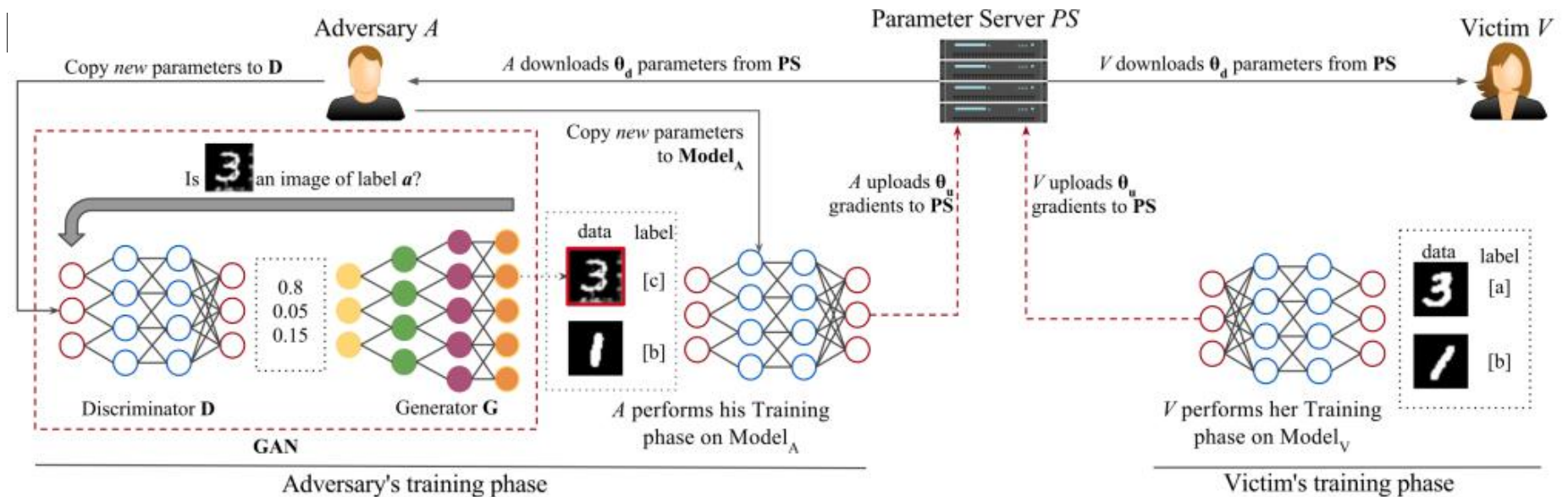


Figure 4: GAN Attack on collaborative deep learning. The victim on the right trains the model with images of 3s (class *a*) and images of 1s (class *b*). The adversary only has images of class *b* (1s) and uses its label *c* and a GAN to fool the victim into releasing information about class *a*. The attack can be easily generalized to several classes and users. The adversary does not even need to start with any true samples.

Conclusion

- Property and membership attacks on collaborative learning have been demonstrated
- Uncorrelated properties to main task can be leaked through model/gradient updates
- Active attacks can make property inference even more powerful
- Countermeasures such as fewer gradient sharing, dimensionality reduction, dropout are not very effective
- Collaborative learning has security vulnerabilities; more research is required to come up with effective countermeasures



Q & A
