Privacy Preserving Collaborative Machine Learning

Dario Pasquini

EPFL

SPRING

SECURITY AND PRIVACY ENGINEERING LABORATORY

Privacy Preserving Collaborative Machine Learning ?

Dario Pasquini

EPFL

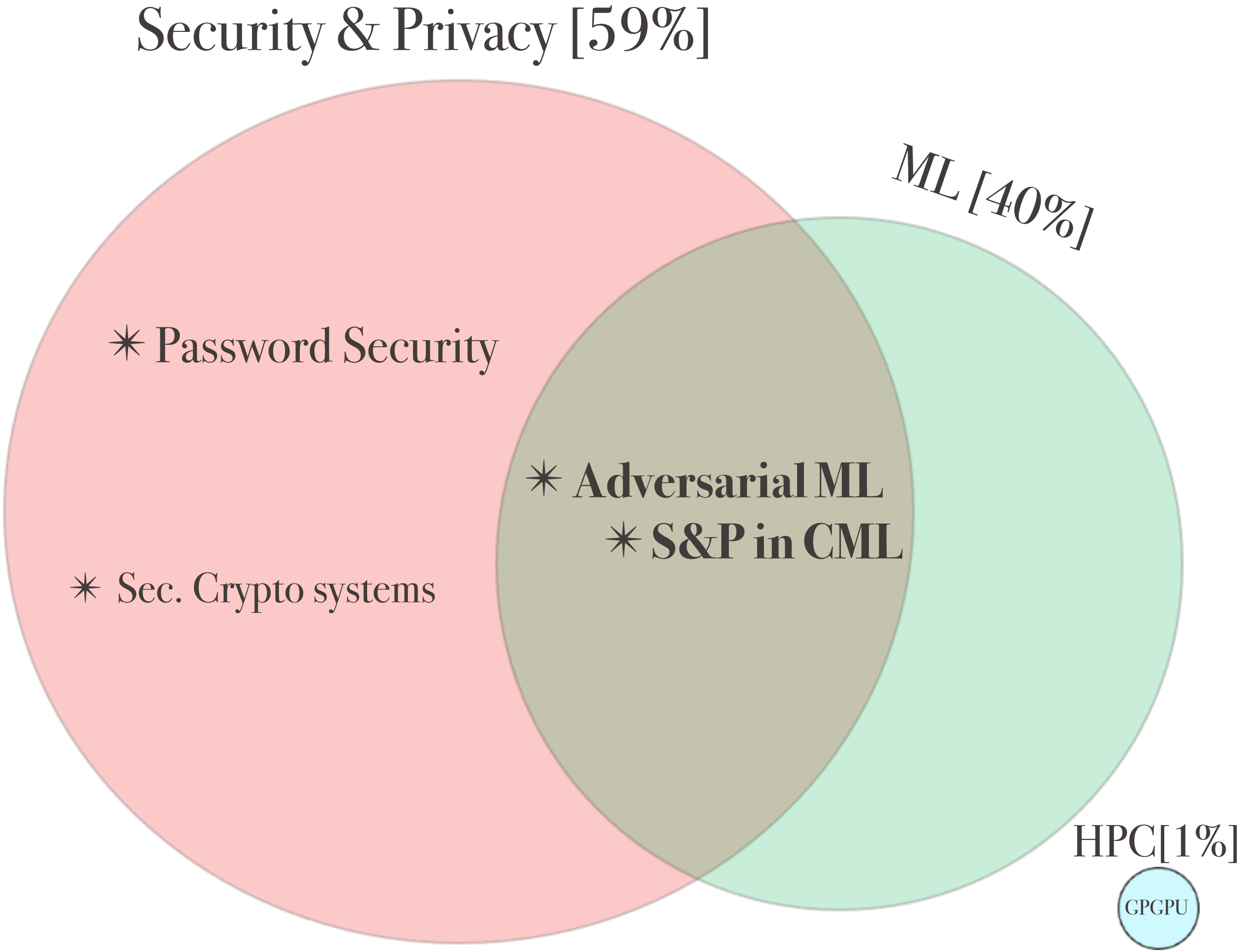
SPRING

SECURITY AND PRIVACY ENGINEERING LABORATORY

About me…

Dario Pasquini, PhD
2nd year Postdoc at EPFL (but leaving soon)

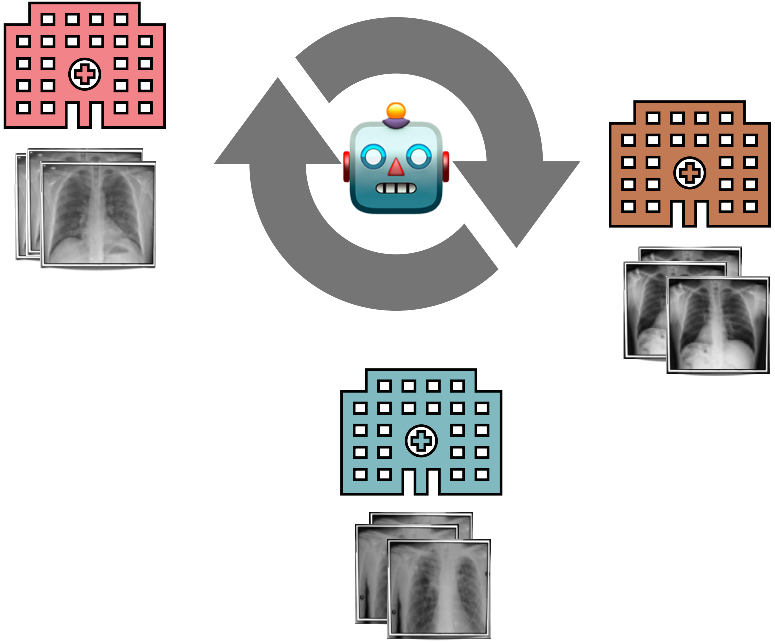
More info at:
https://pasquini-dario.github.io/me/



Background:

- A bit about Collaborative Machine Learning (CML).
- CML is not a private.

The problem we want to solve with CML:



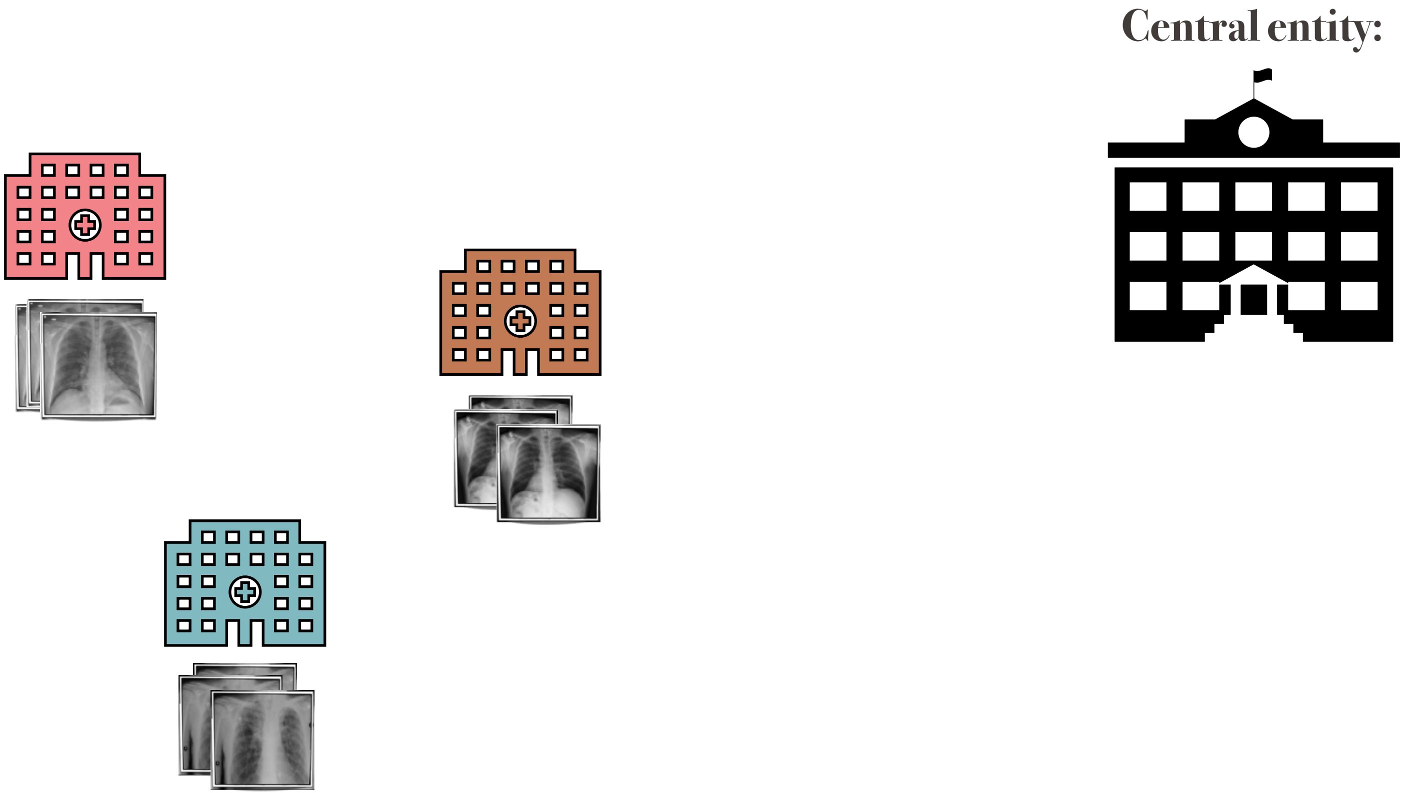
A set of users/organizations (e.g., hospitals):

- Everyone comes with some local data.
    - Not enough to train a ML model locally.
    - Not enough representative.

Let’s collaborate!

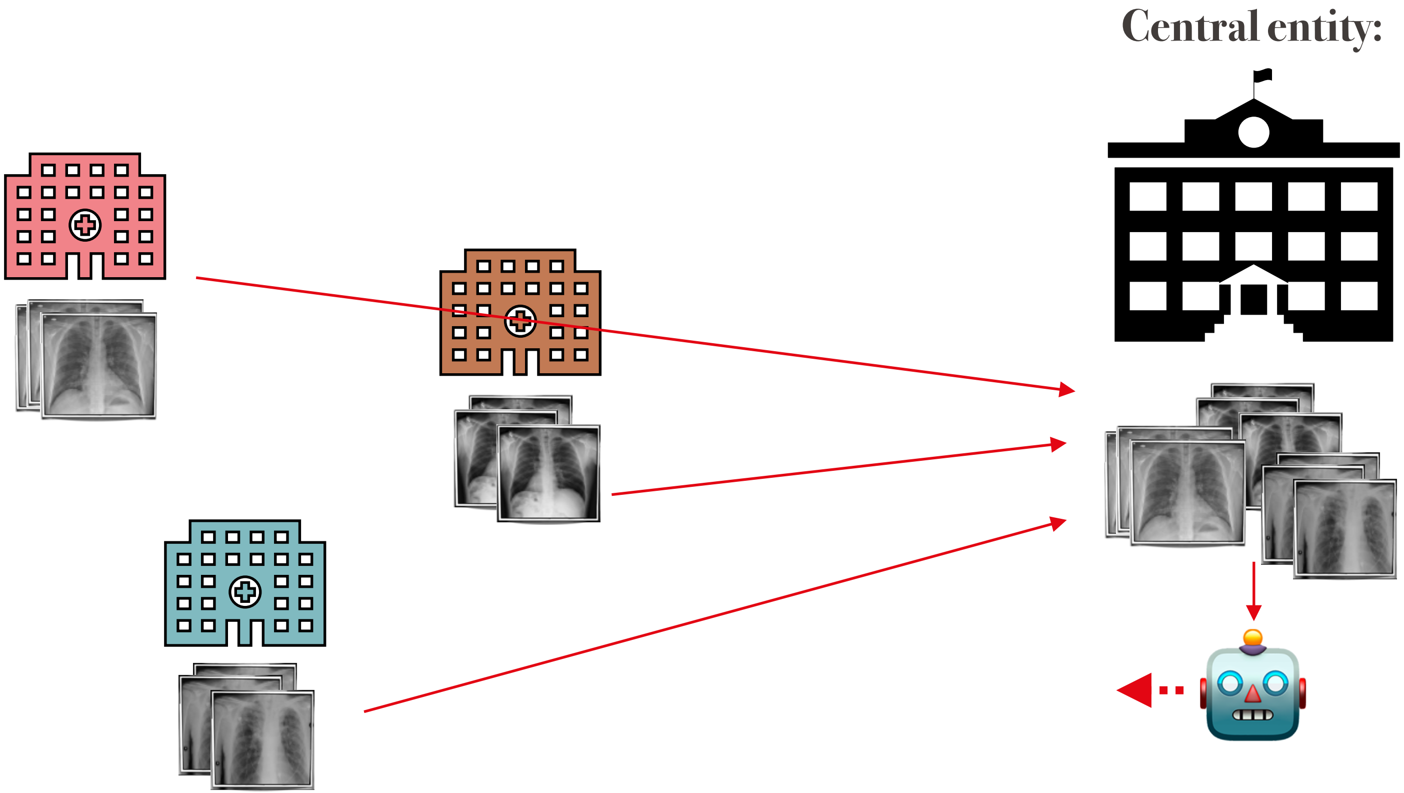
- Train a shared Machine Learning model (🤖) using everyone′s data.

The naive solution: “Centralized Learning”

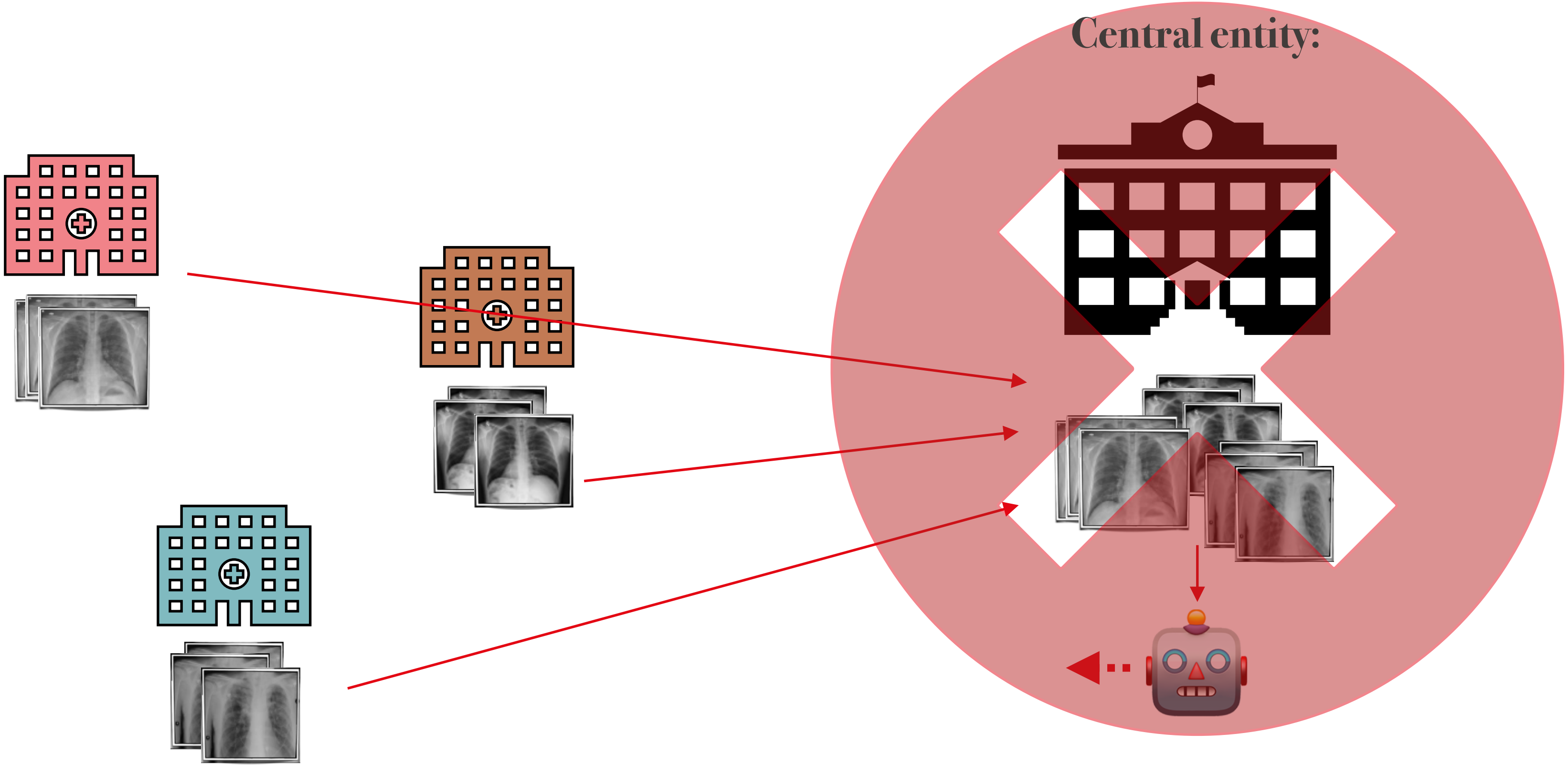


Central entity:

The naive solution: “Centralized Learning”



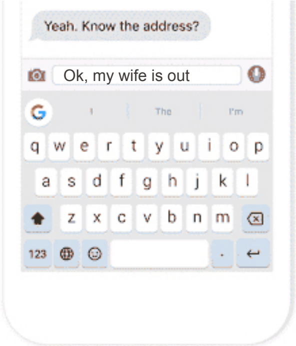
The naive solution: “Centralized Learning”



Why parties can't and shouldn't share their data

★Usually, valuable data is also sensitive:

- e.g., text you write on your phone (google actually did it).

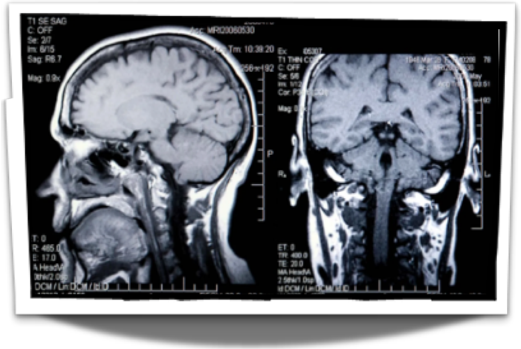


Yeah. Know the address?

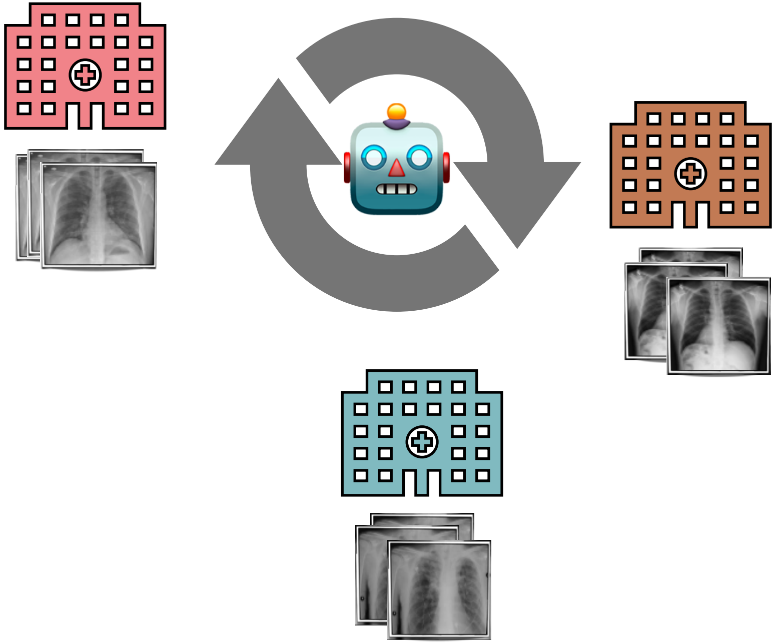
Ok, my wife is out

★Regulations (e.g., GDPR, HIPAA):

- E.g., Hospital's data must not leave the hospital.



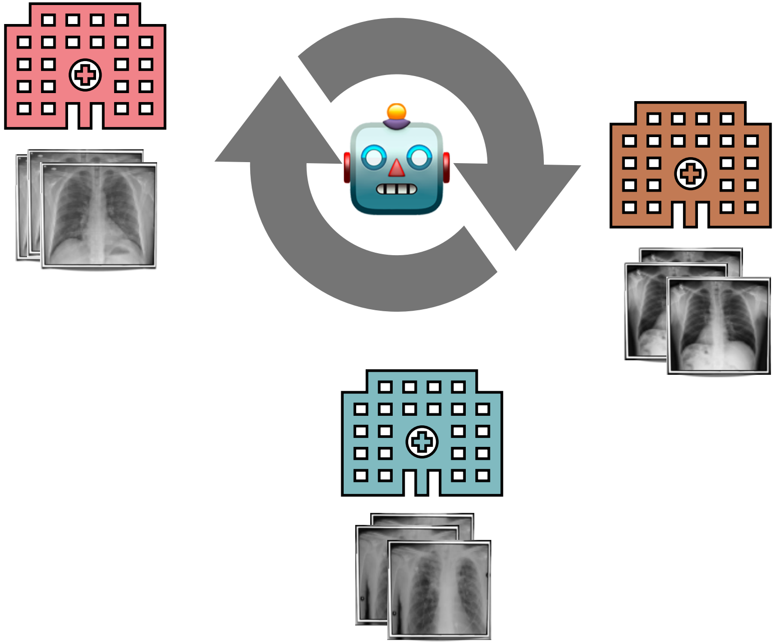
Here comes Collaborative Machine Learning



★Data stays local; data never leaves users' devices 🪄

- <u>Only proxy signals are shared among parties.</u>

Here comes Collaborative Machine Learning



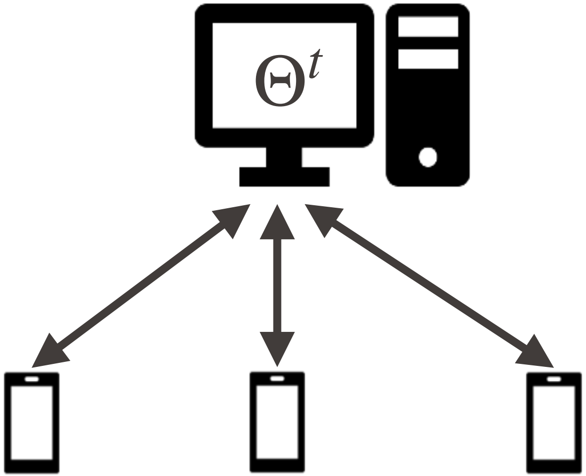
★Data stays local; data never leaves users' devices 🪄

- Only proxy signals are shared among parties.

"That's Privacy Preserving"

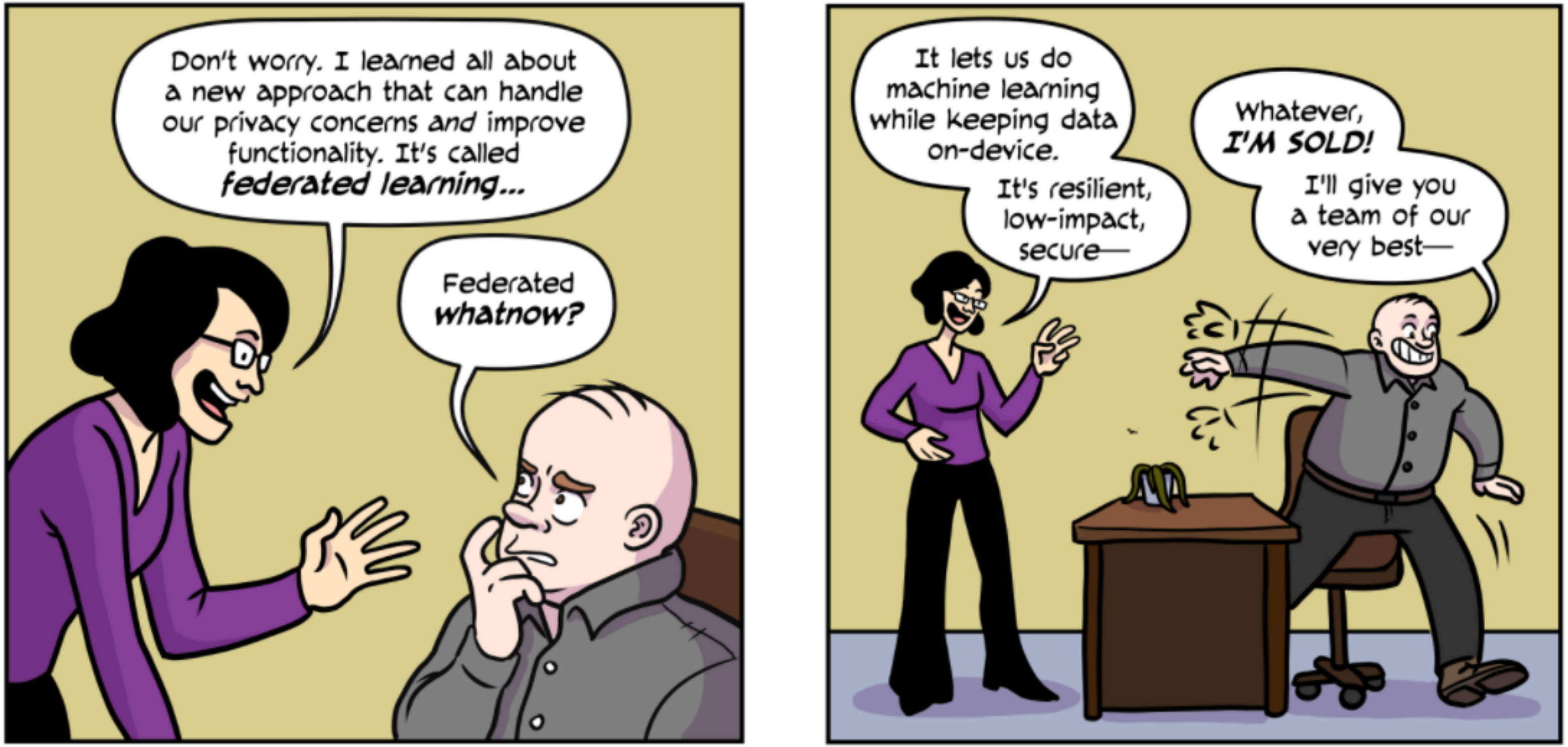
Federated Learning (FL)

Parameter Server:



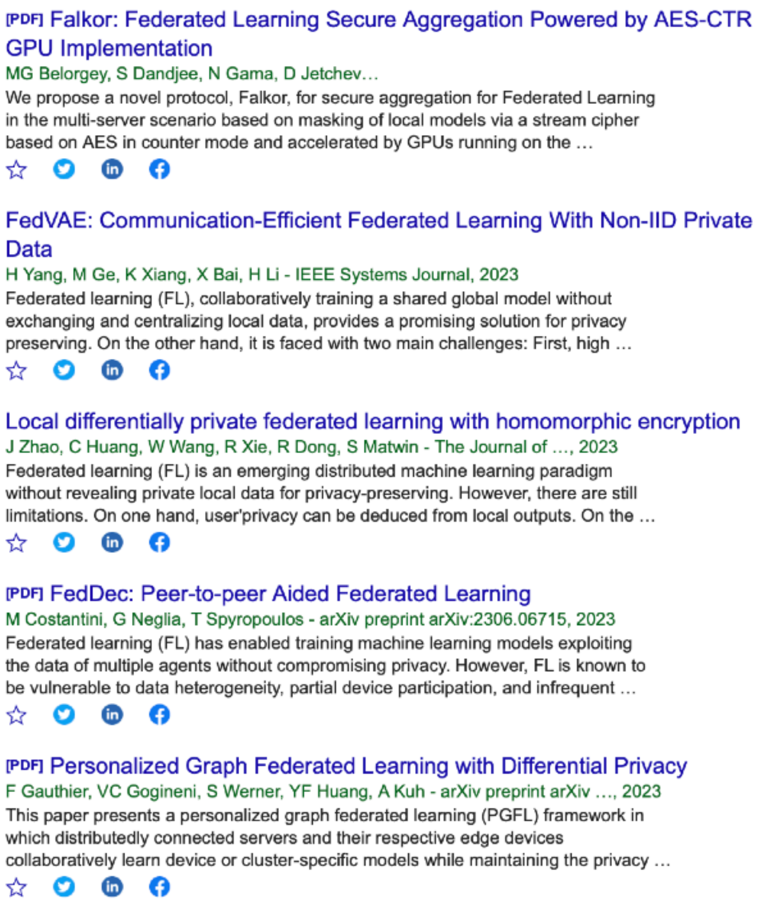
since 2016; by google

FL & the community


Don't worry. I learned all about a new approach that can handle our privacy concerns and improve functionality. It's called ***federated learning...***
Federated ***whatnow?***
It lets us do machine learning while keeping data on-device.
It's resilient, low-impact, secure—
Whatever, ***I'M SOLD!***
I'll give you a team of our very best—

from: https://federated.withgoogle.com/

My weekly google scholar feed…



[PDF] Falkor: Federated Learning Secure Aggregation Powered by AES-CTR GPU Implementation

MG Belorgey, S Dandjee, N Gama, D Jetchev…

We propose a novel protocol, Falkor, for secure aggregation for Federated Learning in the multi-server scenario based on masking of local models via a stream cipher based on AES in counter mode and accelerated by GPUs running on the …

FedVAE: Communication-Efficient Federated Learning With Non-IID Private Data

H Yang, M Ge, K Xiang, X Bai, H Li - IEEE Systems Journal, 2023

Federated learning (FL), collaboratively training a shared global model without exchanging and centralizing local data, provides a promising solution for privacy preserving. On the other hand, it is faced with two main challenges: First, high …

Local differentially private federated learning with homomorphic encryption

J Zhao, C Huang, W Wang, R Xie, R Dong, S Matwin - The Journal of …, 2023

Federated learning (FL) is an emerging distributed machine learning paradigm without revealing private local data for privacy-preserving. However, there are still limitations. On one hand, user'privacy can be deduced from local outputs. On the …

[PDF] FedDec: Peer-to-peer Aided Federated Learning

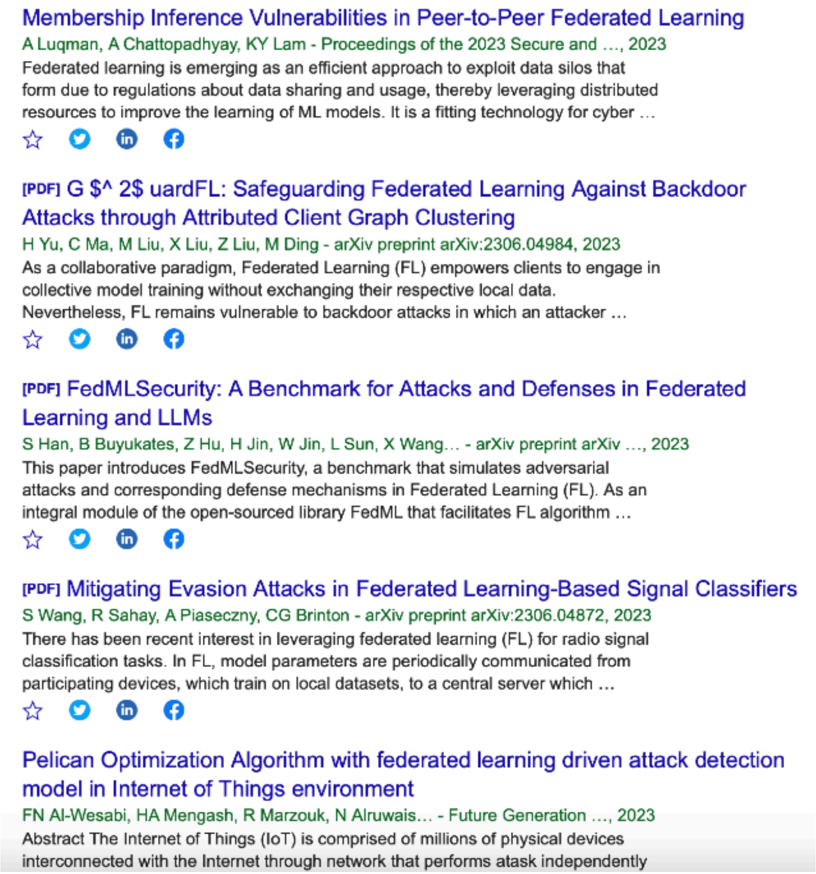
M Costantini, G Neglia, T Spyropoulos - arXiv preprint arXiv:2306.06715, 2023

Federated learning (FL) has enabled training machine learning models exploiting the data of multiple agents without compromising privacy. However, FL is known to be vulnerable to data heterogeneity, partial device participation, and infrequent …

[PDF] Personalized Graph Federated Learning with Differential Privacy

F Gauthier, VC Gogineni, S Werner, YF Huang, A Kuh - arXiv preprint arXiv …, 2023

This paper presents a personalized graph federated learning (PGFL) framework in which distributedly connected servers and their respective edge devices collaboratively learn device or cluster-specific models while maintaining the privacy …



Membership Inference Vulnerabilities in Peer-to-Peer Federated Learning

A Luqman, A Chattopadhyay, KY Lam - Proceedings of the 2023 Secure and …, 2023

Federated learning is emerging as an efficient approach to exploit data silos that form due to regulations about data sharing and usage, thereby leveraging distributed resources to improve the learning of ML models. It is a fitting technology for cyber …

[PDF] G $^ 2$ uardFL: Safeguarding Federated Learning Against Backdoor Attacks through Attributed Client Graph Clustering

H Yu, C Ma, M Liu, X Liu, Z Liu, M Ding - arXiv preprint arXiv:2306.04984, 2023

As a collaborative paradigm, Federated Learning (FL) empowers clients to engage in collective model training without exchanging their respective local data. Nevertheless, FL remains vulnerable to backdoor attacks in which an attacker …

[PDF] FedMLSecurity: A Benchmark for Attacks and Defenses in Federated Learning and LLMs

S Han, B Buyukates, Z Hu, H Jin, W Jin, L Sun, X Wang… - arXiv preprint arXiv …, 2023

This paper introduces FedMLSecurity, a benchmark that simulates adversarial attacks and corresponding defense mechanisms in Federated Learning (FL). As an integral module of the open-sourced library FedML that facilitates FL algorithm …

[PDF] Mitigating Evasion Attacks in Federated Learning-Based Signal Classifiers

S Wang, R Sahay, A Piaseczny, CG Brinton - arXiv preprint arXiv:2306.04872, 2023

There has been recent interest in leveraging federated learning (FL) for radio signal classification tasks. In FL, model parameters are periodically communicated from participating devices, which train on local datasets, to a central server which …

Pelican Optimization Algorithm with federated learning driven attack detection model in Internet of Things environment

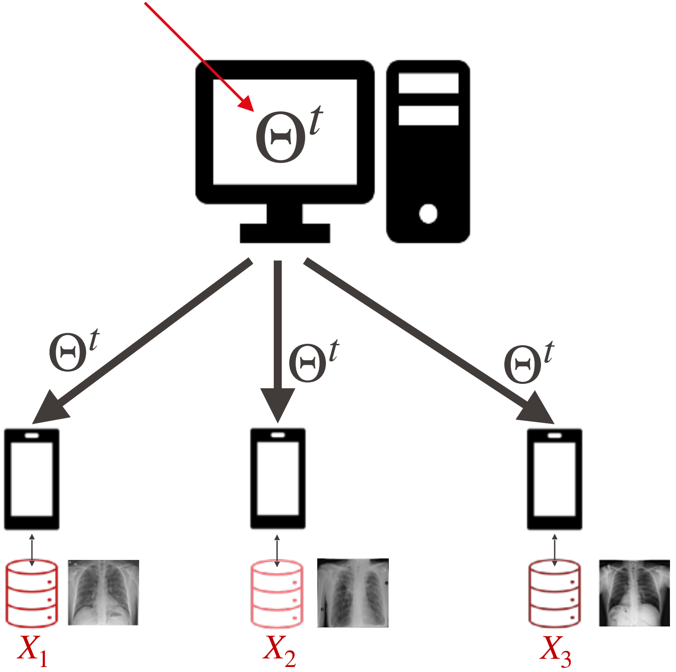
FN Al-Wesabi, HA Mengash, R Marzouk, N Alruwais… - Future Generation …, 2023

Abstract The Internet of Things (IoT) is comprised of millions of physical devices interconnected with the Internet through network that performs atask independently

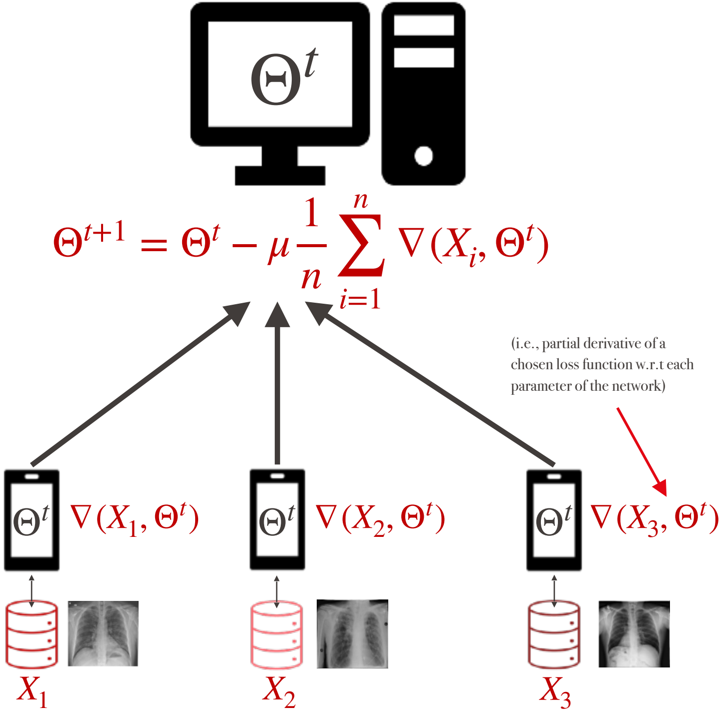
Federated Learning (FL) (FedSGD):

Phase 1: Parameters distribution:

Parameters of the Machine Learning model (a neural network)



Phase 2,3: Local training & model updates aggregation:



$$\Theta^{t+1} = \Theta^t - \mu \frac{1}{n} \sum_{i=1}^{n} \nabla(X_i, \Theta^t)$$

(i.e., partial derivative of a chosen loss function w.r.t each parameter of the network)

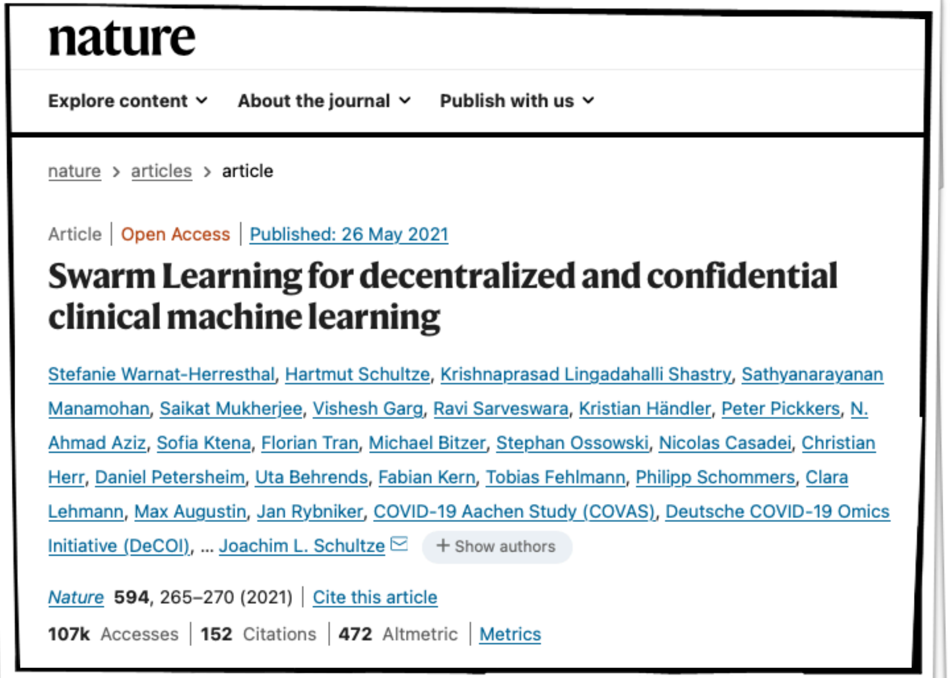
FL is private:

“Model updates” (i.e., gradient from one or more SGD iterations) **are not data**:

$$\nabla(X, \Theta) \neq X$$

Real data remains safely stored on device and it is **never** shared.
What can go wrong?

"CML is private":



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

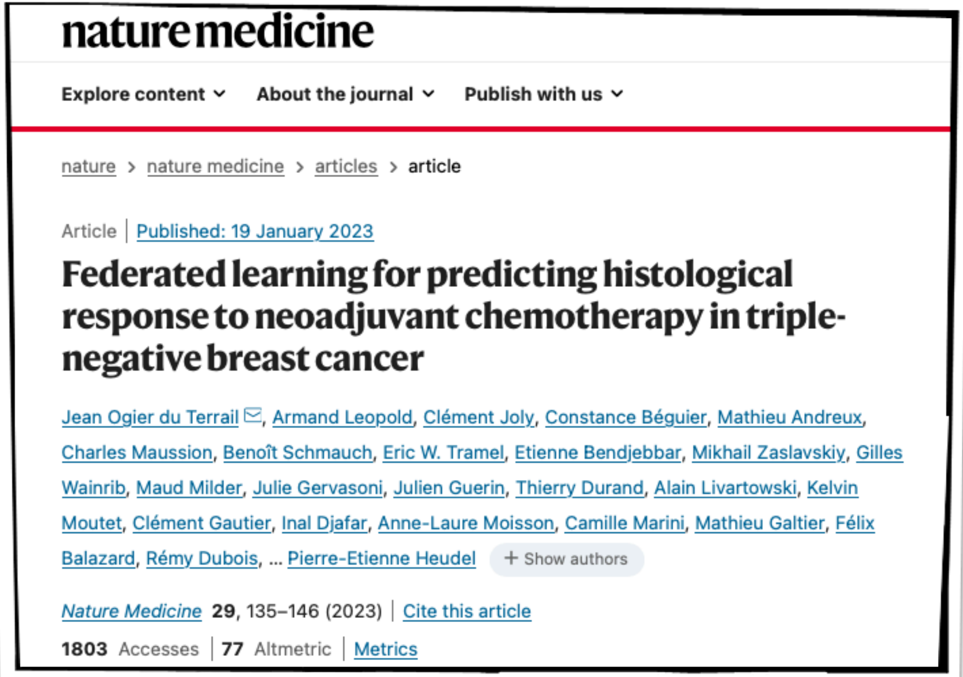
Article | Open Access | Published: 26 May 2021

Swarm Learning for decentralized and confidential clinical machine learning

Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N. Ahmad Aziz, Sofia Ktena, Florian Tran, Michael Bitzer, Stephan Ossowski, Nicolas Casadei, Christian Herr, Daniel Petersheim, Uta Behrends, Fabian Kern, Tobias Fehlmann, Philipp Schommers, Clara Lehmann, Max Augustin, Jan Rybniker, COVID-19 Aachen Study (COVAS), Deutsche COVID-19 Omics Initiative (DeCOI), … Joachim L. Schultze ✉ + Show authors

Nature **594**, 265–270 (2021) | Cite this article

107k Accesses | **152** Citations | **472** Altmetric | Metrics



nature medicine

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › nature medicine › articles › article

Article | Published: 19 January 2023

Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer

Jean Ogier du Terrail ✉, Armand Leopold, Clément Joly, Constance Béguier, Mathieu Andreux, Charles Maussion, Benoît Schmauch, Eric W. Tramel, Etienne Bendjebbar, Mikhail Zaslavskiy, Gilles Wainrib, Maud Milder, Julie Gervasoni, Julien Guerin, Thierry Durand, Alain Livartowski, Kelvin Moutet, Clément Gautier, Inal Djafar, Anne-Laure Moisson, Camille Marini, Mathieu Galtier, Félix Balazard, Rémy Dubois, … Pierre-Etienne Heudel + Show authors

Nature Medicine **29**, 135–146 (2023) | Cite this article

1803 Accesses | **77** Altmetric | Metrics

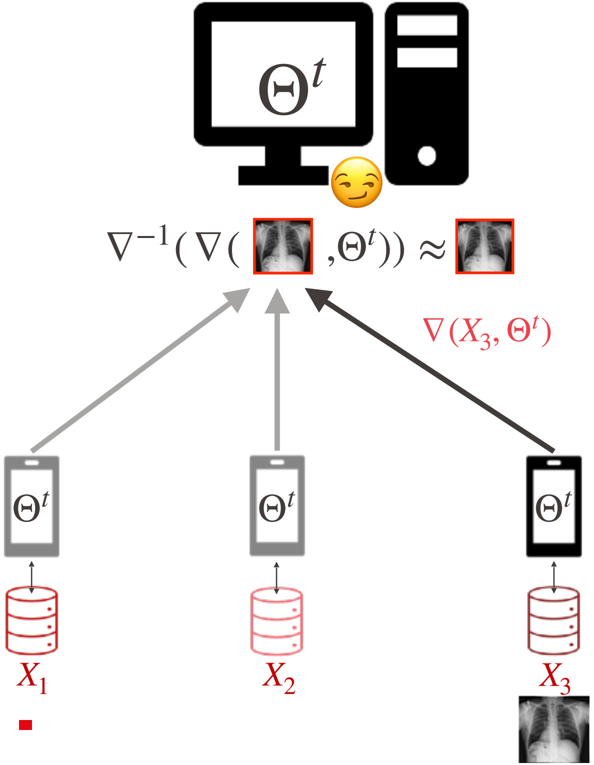
Wait, is FL private?

★Gradient is just a smooth function of the input data!

★From a **formal security perspective**, sending data or gradient is the same thing:

$$\nabla(X, \Theta) \approx X$$

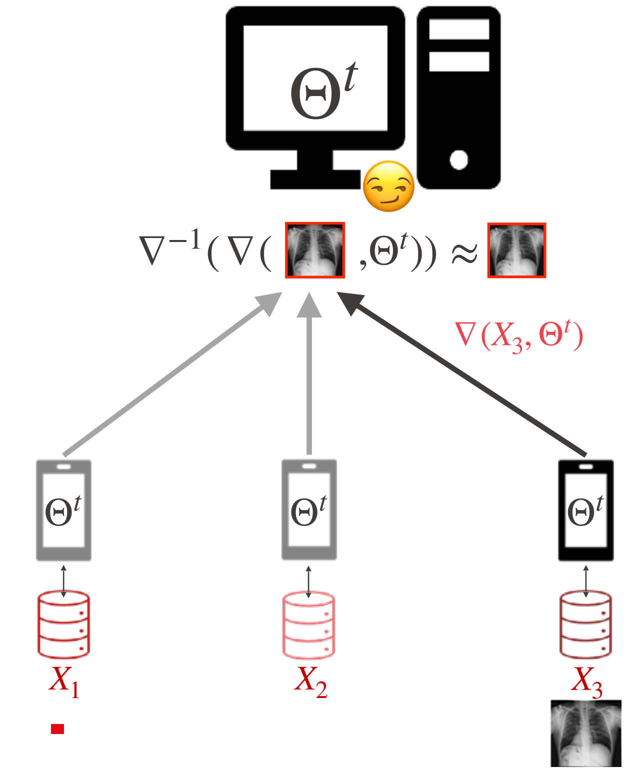
Gradient inversion attack:



- It can be seen as a second order optimization problem:
  - "Find synthetic data (X') such that the gradient generated by X' on Θ^t is similar to the one received from the client":

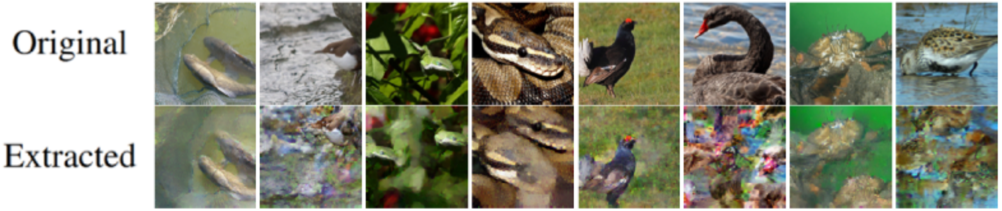
$$argmin_{X'} : \|\nabla(X_3, \Theta^t) - \nabla(X', \Theta^t))\|_2^2$$

Gradient inversion attack:



- It can be seen as a second order optimization problem:
    - “Find synthetic data (X') such that the gradient generated by X' on Θ^t is similar to the one received from the client”:

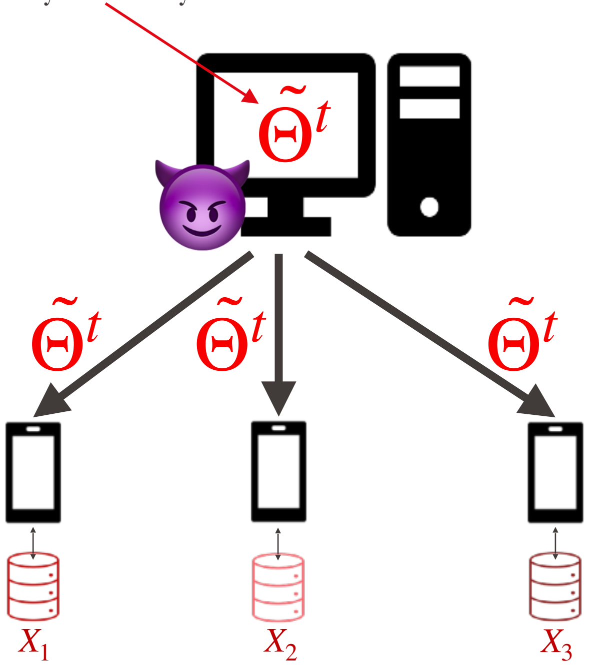
$$argmin_{X'} : \|\nabla(X_3, \Theta^t) - \nabla(X', \Theta^t))\|_2^2$$



from: Geiping et al *“Inverting Gradients - How easy is it to break privacy in federated learning?”, NeurIPS 2020*

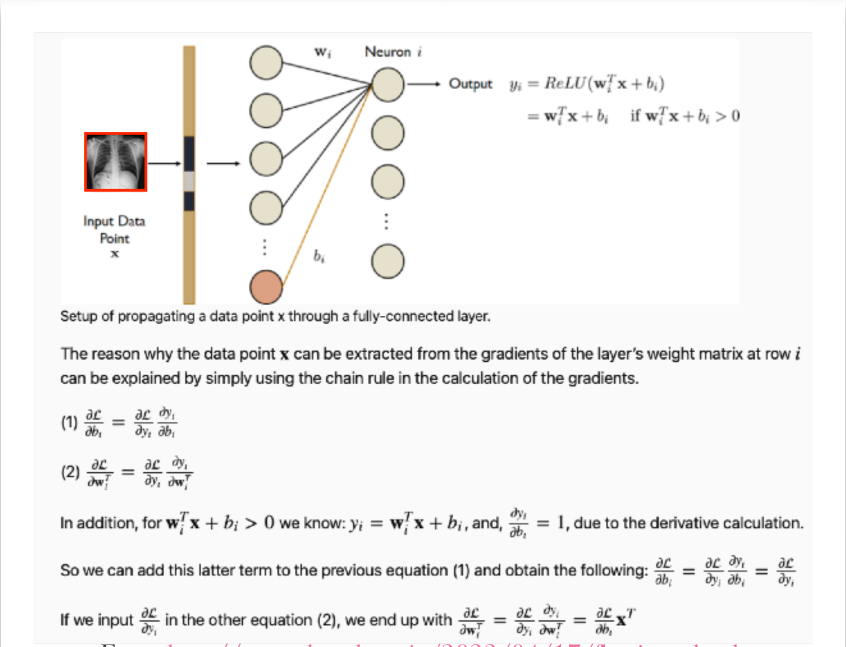
Gradient inversion with a malicious server :

Arbitrarily chosen by the attacker!



The server creates and distributes **malicious parameters**:

- Just an intuition: $\tilde{\Theta}^t$ is forged in a such way that the gradient of the final linear layers "memorizes" the input data:



$\mathbf{w}_i$
Neuron i
Output
$y_i = ReLU(\mathbf{w}_i^T\mathbf{x} + b_i)$
$= \mathbf{w}_i^T\mathbf{x} + b_i \quad \text{if } \mathbf{w}_i^T\mathbf{x} + b_i > 0$
Input Data Point $\mathbf{x}$
b_i

Setup of propagating a data point x through a fully-connected layer.

The reason why the data point $\mathbf{x}$ can be extracted from the gradients of the layer's weight matrix at row i can be explained by simply using the chain rule in the calculation of the gradients.

(1) $\frac{\partial \mathcal{L}}{\partial b_i} = \frac{\partial \mathcal{L}}{\partial y_i} \frac{\partial y_i}{\partial b_i}$

(2) $\frac{\partial \mathcal{L}}{\partial \mathbf{w}_i^T} = \frac{\partial \mathcal{L}}{\partial y_i} \frac{\partial y_i}{\partial \mathbf{w}_i^T}$

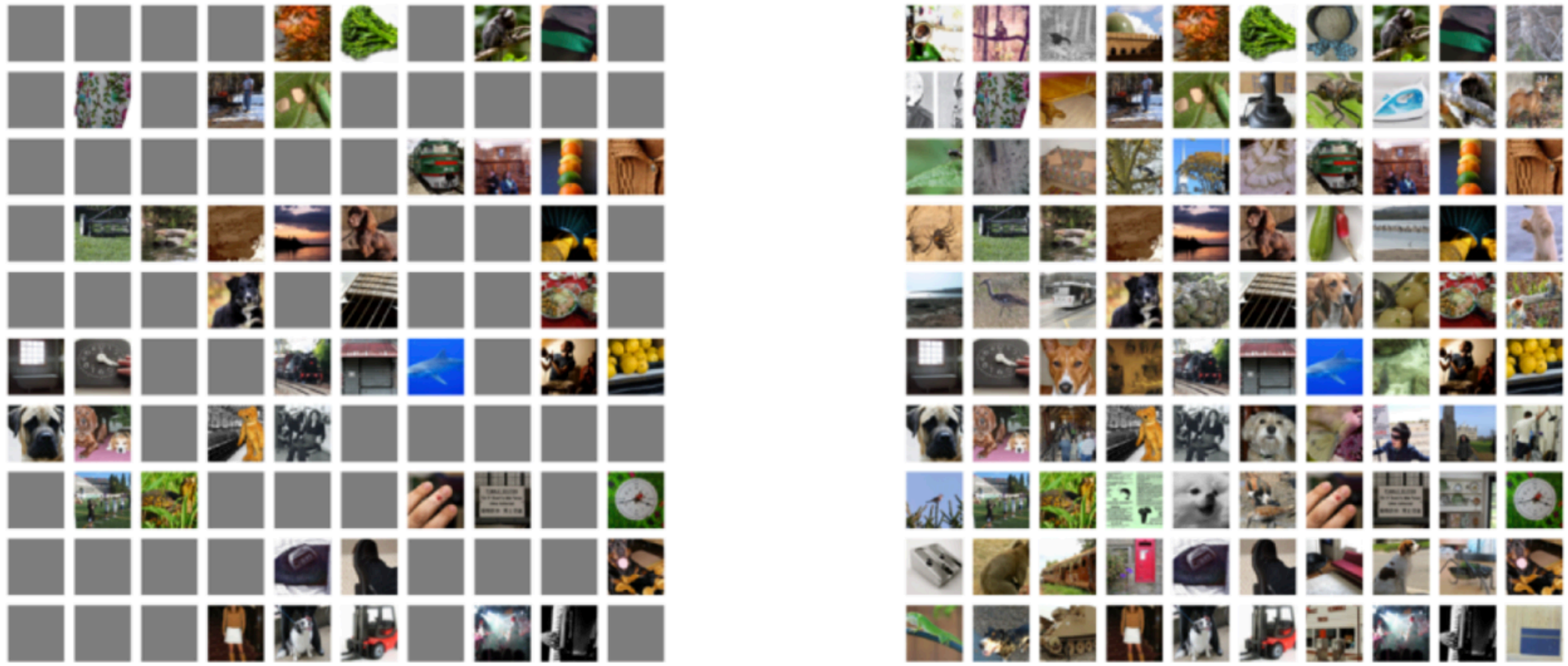
In addition, for $\mathbf{w}_i^T\mathbf{x} + b_i > 0$ we know: $y_i = \mathbf{w}_i^T\mathbf{x} + b_i$, and, $\frac{\partial y_i}{\partial b_i} = 1$, due to the derivative calculation.

So we can add this latter term to the previous equation (1) and obtain the following: $\frac{\partial \mathcal{L}}{\partial b_i} = \frac{\partial \mathcal{L}}{\partial y_i} \frac{\partial y_i}{\partial b_i} = \frac{\partial \mathcal{L}}{\partial y_i}$

If we input $\frac{\partial \mathcal{L}}{\partial y_i}$ in the other equation (2), we end up with $\frac{\partial \mathcal{L}}{\partial \mathbf{w}_i^T} = \frac{\partial \mathcal{L}}{\partial y_i} \frac{\partial y_i}{\partial \mathbf{w}_i^T} = \frac{\partial \mathcal{L}}{\partial b_i}\mathbf{x}^T$

From: http://www.cleverhans.io/2022/04/17/fl-privacy.html

Recovers **exact** copies of **some** of the data in the batch:



(a) Reconstructed data points.

(b) Original data points.

“FL is private”:

- Despite FL is **believed to be a privacy preserving mechanism:**
  - Vanilla FL does not offer any concrete form of protection.

“FL is private”:

- Despite FL is **believed to be a privacy preserving mechanism:**
  - Vanilla FL does not offer any concrete form of protection.

- **“Let’s make it secure, then”:**
  - Secure Aggregation.
  - Differential Privacy.
  - Protocols variations:
    - Peer-to-Peer Federated Learning.
    - Split Learning.

My research is about answering:

Does this stuff actually make CML more private?

[Spoiler Alert]

My research is about answering:

Does this stuff actually make CML more private?

[Spoiler Alert]

Typically, it doesn′t!

Agenda:

- [ACM CCS’22] **“Eluding Secure Aggregation in Federated Learning via Model Inconsistency”**
  Dario Pasquini, Danilo Francati, Giuseppe Ateniese

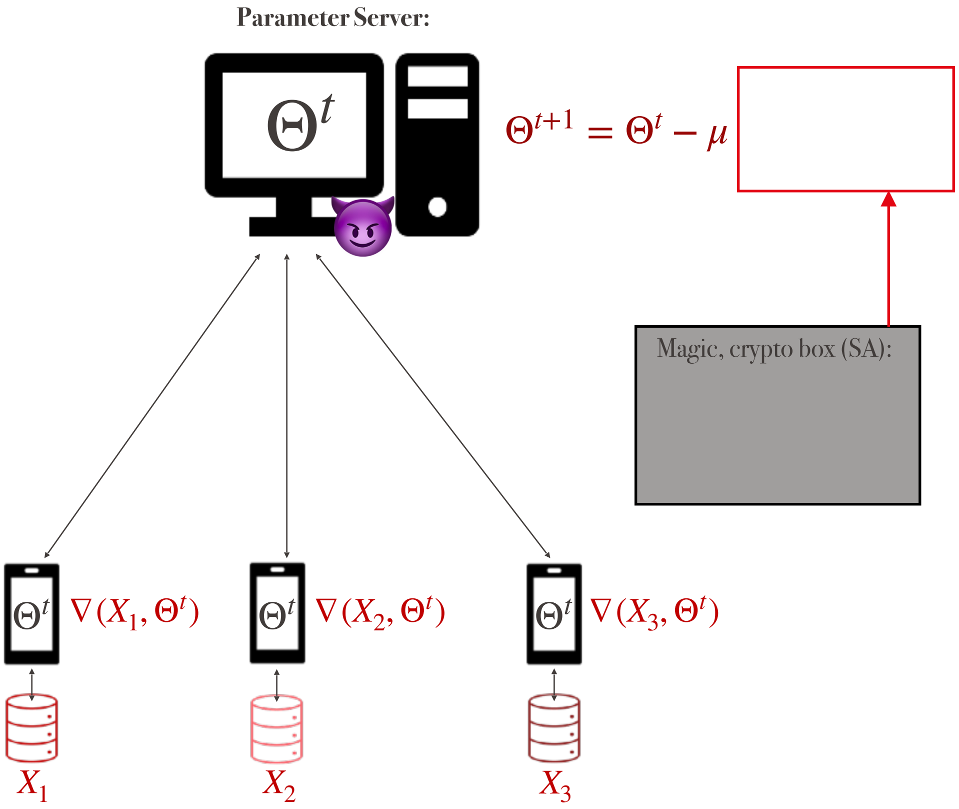
  –

- [IEEE S&P’23] **“On the (In)security of Peer-to-Peer Decentralized Machine Learning”**
  Dario Pasquini, Mathilde Raynal, Carmela Troncoso

  –

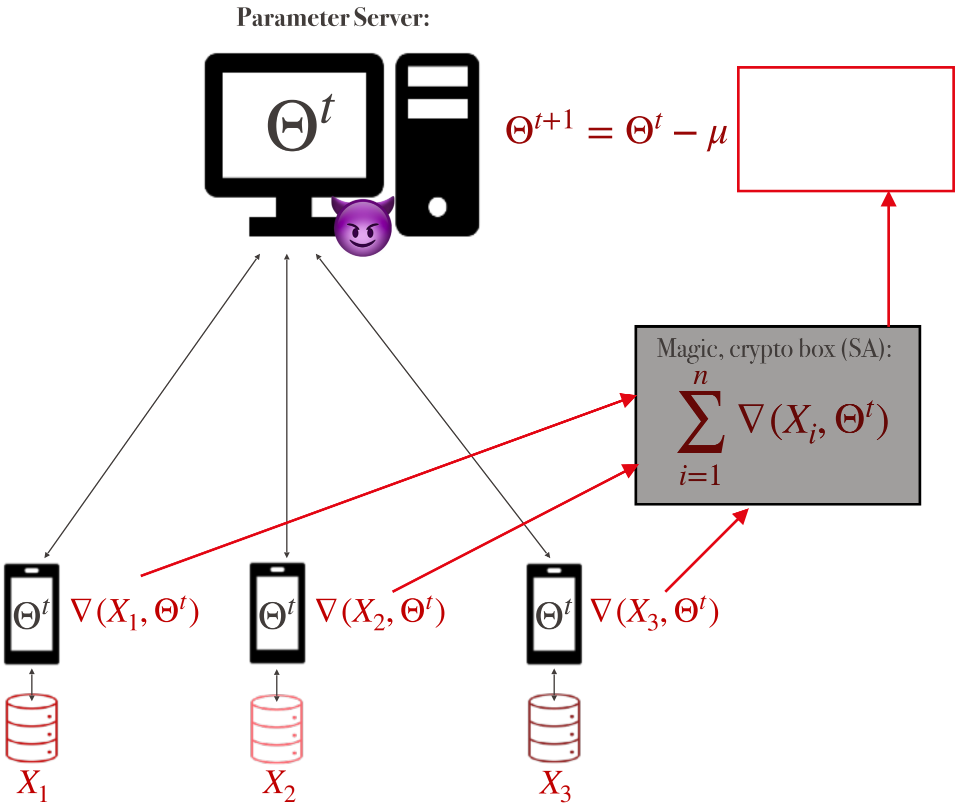
- [ACM CCS’21] **“Unleashing the tiger: Inference attacks on split learning”**
  Dario Pasquini, Giuseppe Ateniese, Massimo Bernaschi

Secure Aggregation (SA) in FL:



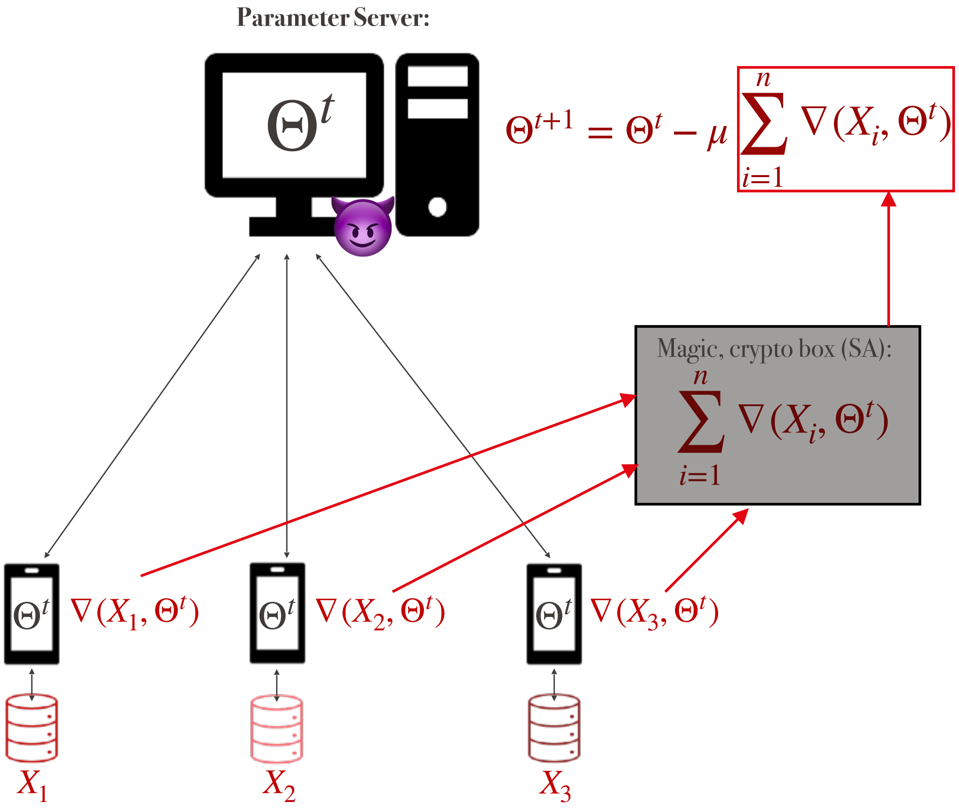
Bonawitz et al. *"Practical secure aggregation for privacy-preserving machine learning"* CCS '17 (>2100 citations in 5 years)

Secure Aggregation (SA) in FL:



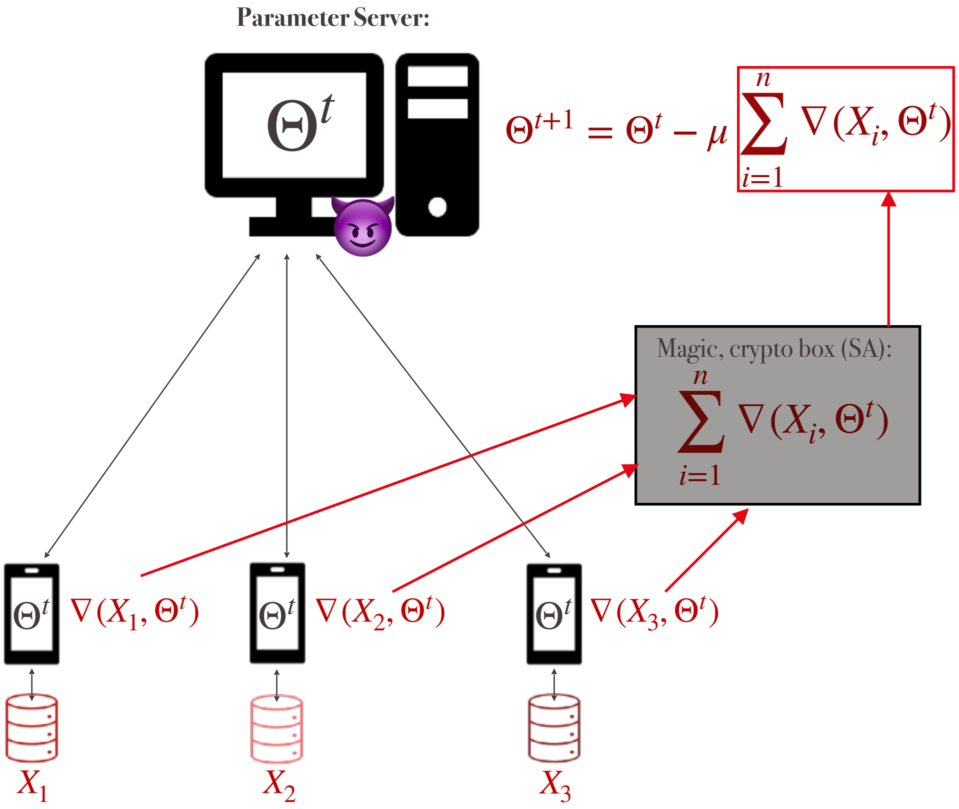
Bonawitz et al. *"Practical secure aggregation for privacy-preserving machine learning"* CCS '17 (>2100 citations in 5 years)

Secure Aggregation (SA) in FL:



Bonawitz et al. *"Practical secure aggregation for privacy-preserving machine learning"* CCS '17 (>2100 citations in 5 years)

Secure Aggregation (SA) in FL:



Parameter Server:

Θ^t

$$\Theta^{t+1} = \Theta^t - \mu \boxed{\sum_{i=1}^{n} \nabla(X_i, \Theta^t)}$$

Magic, crypto box (SA):

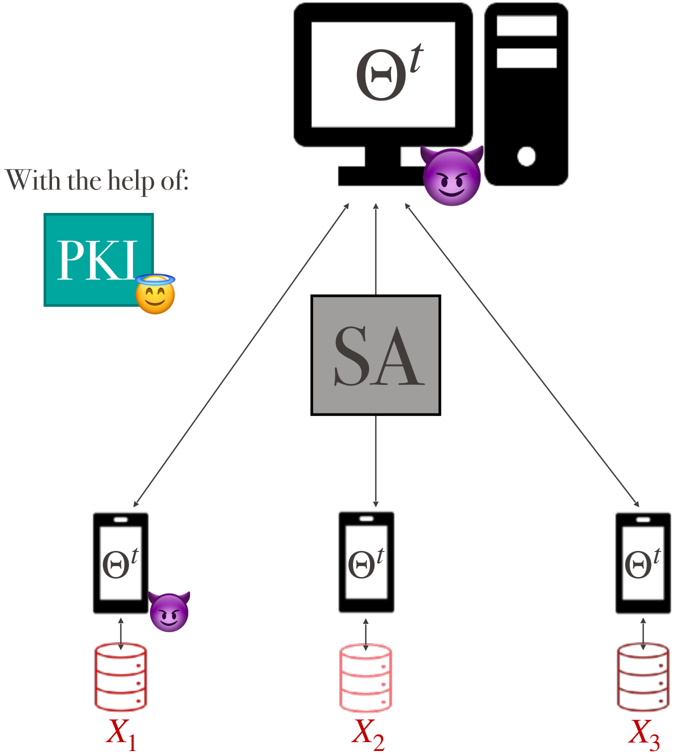
$$\sum_{i=1}^{n} \nabla(X_i, \Theta^t)$$

SA+FL expected privacy:

🔒 "*Privacy by aggregation*":
Aggregating together a suitable number of model updates smooths out the information carried out by individual contributions.

Bonawitz et al. *"Practical secure aggregation for privacy-preserving machine learning"* CCS '17 (>2100 citations in 5 years)

The security of Secure Aggregation:
(adversarial server)



SA's Security definition:

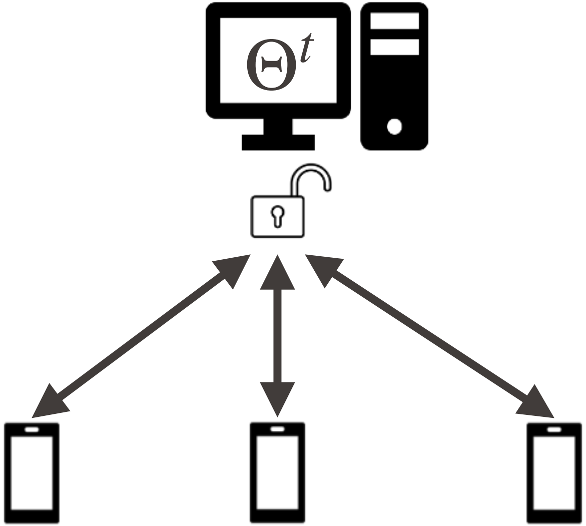
Nothing is learned about the inputs apart from what can be inferred from the final sum.

🔒 **SA is proven secure against a malicious server:**

- that can collude with up to $\frac{n}{3} - 1$ users

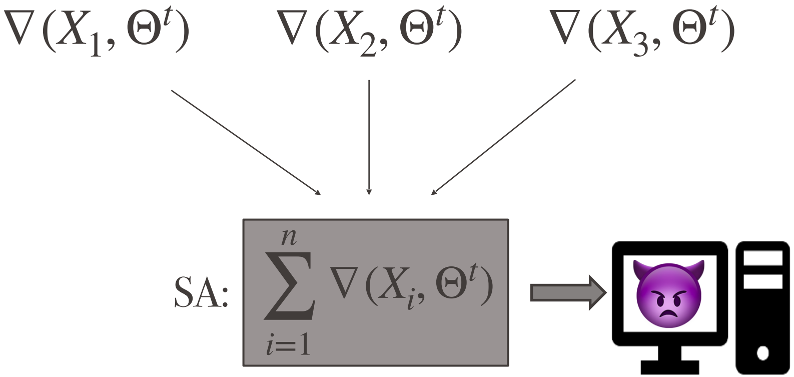
Bonawitz et al. *"Practical secure aggregation for privacy-preserving machine learning"* CCS '17 (>2100 citations in 5 years)

Is Secure Aggregation actually Secure?



Pasquini, Francati, and Ateniese"*Eluding Secure Aggregation in Federated via Model Inconsistency*" CCS'22

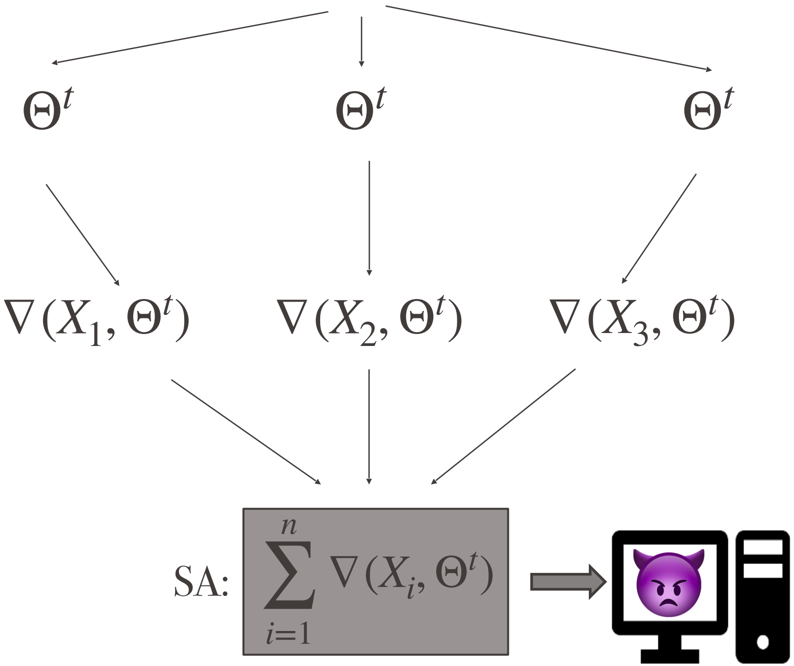
The problem with SA+FL:



$\nabla(X_1, \Theta^t)$ $\nabla(X_2, \Theta^t)$ $\nabla(X_3, \Theta^t)$

SA: $\sum_{i=1}^{n} \nabla(X_i, \Theta^t)$

The problem with SA+FL:

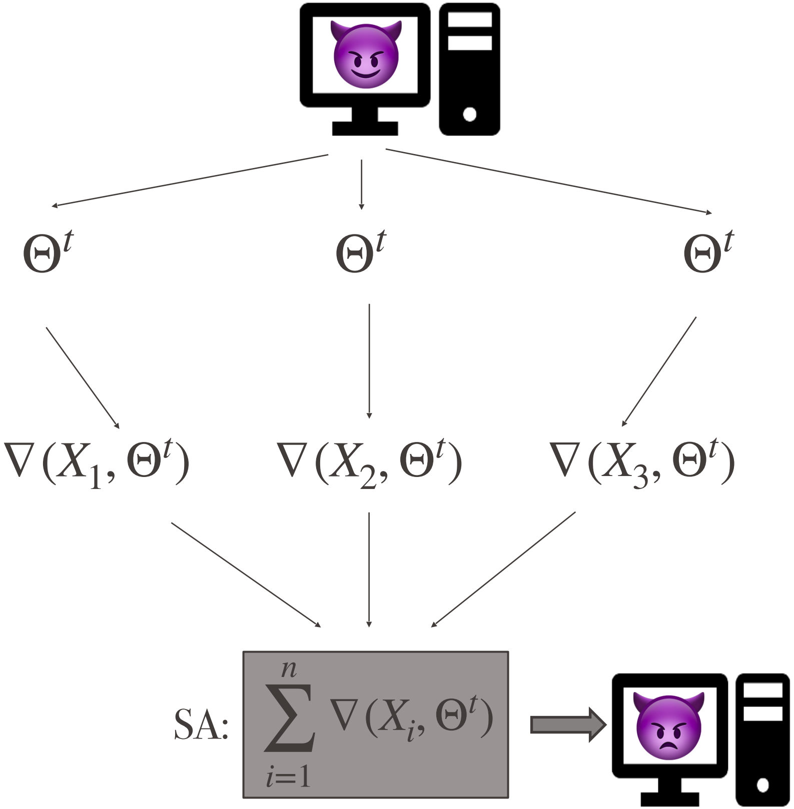


Θ^t Θ^t Θ^t

$\nabla(X_1, \Theta^t)$ $\nabla(X_2, \Theta^t)$ $\nabla(X_3, \Theta^t)$

SA: $\sum_{i=1}^{n} \nabla(X_i, \Theta^t)$

The problem with SA+FL:

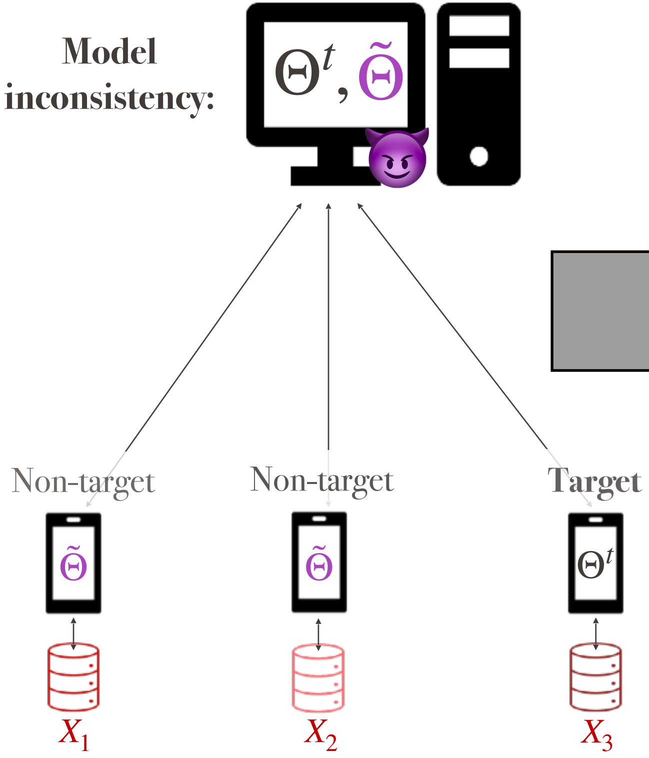


Θ^t Θ^t Θ^t

$\nabla(X_1, \Theta^t)$ $\nabla(X_2, \Theta^t)$ $\nabla(X_3, \Theta^t)$

SA: $\sum_{i=1}^{n} \nabla(X_i, \Theta^t)$

Gradient Suppression attack :

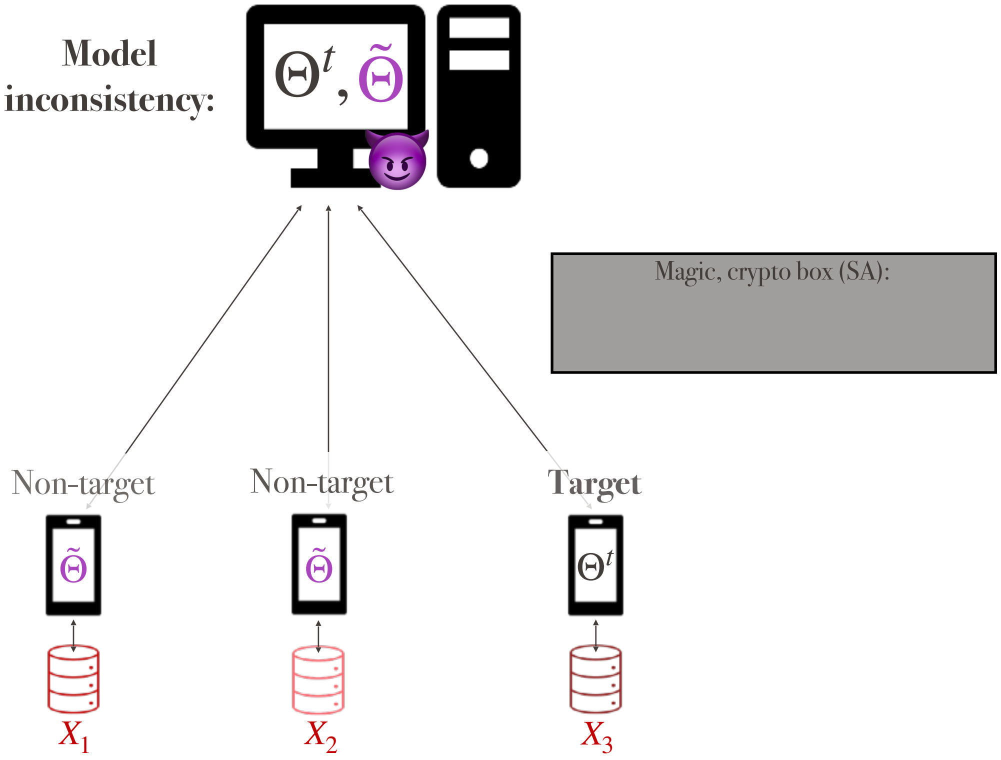


Magic, crypto box (SA):

Attack setup:

1. The server selects a **target** user (all the other users are **non-targets**).
2. The server distributes different parameters to **target** and **non-targets**
   - The **target** gets: Θ^t (as in the honest execution)
   - The **non-targets** get a set of maliciously crafted parameters $\tilde{\Theta}$

Gradient Suppression attack :



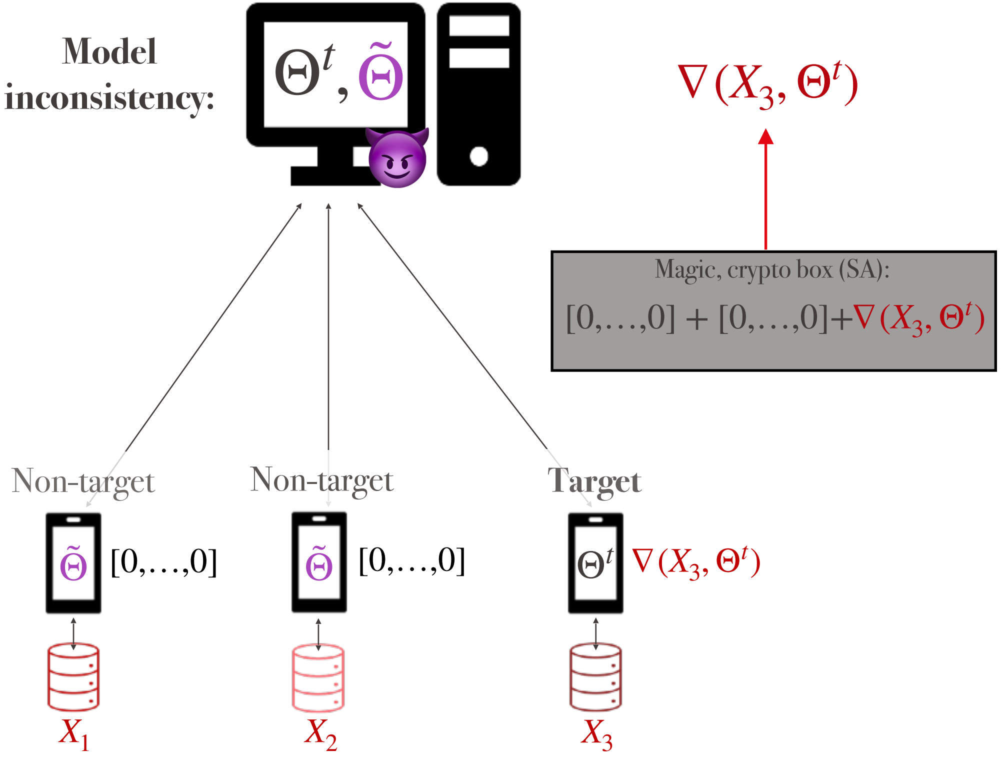
Attack setup:

1. The server selects a **target** user (all the other users are **non-targets**).
2. The server distributes different parameters to **target** and **non-targets**
   - The **target** gets: Θ^t (as in the honest execution)
   - The **non-targets** get a set of maliciously crafted parameters $\tilde{\Theta}$

$\tilde{\Theta}$ is created s.t.:

$$\forall_{X \in \mathbb{X}} \nabla(X, \tilde{\Theta}) = [0,\ldots,0]$$

Gradient Suppression attack :



Attack setup:

1. The server selects a **target** user (all the other users are **non-targets**).
2. The server distributes different parameters to **target** and **non-targets**
   - The **target** gets: Θ^t (as in the honest execution)
   - The **non-targets** get a set of maliciously crafted parameters $\tilde{\Theta}$

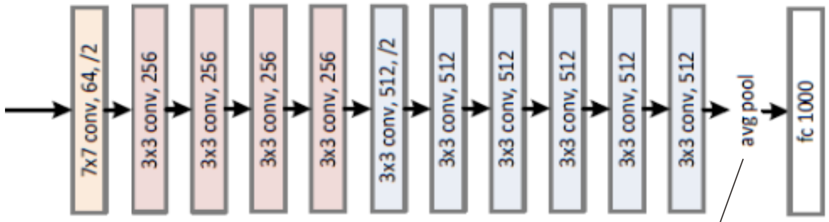
$\tilde{\Theta}$ is created s.t.:

$$\forall_{X \in \mathbb{X}} \nabla(X, \tilde{\Theta}) = [0,\ldots,0]$$

How to kill a neural net:

- The easiest way:
  - Choose $\tilde{\Theta}$ such that $L(y,\ f_{\tilde{\Theta}}(x))$ is a constant function:

$$\begin{array}{c}[0,0,0\ \dots,\ 0]\\ \|\end{array}$$



$$\forall_{x,y\in X}\nabla(X,\tilde{\Theta}) = [0,\dots,0]$$

$$\Uparrow$$

$$g \otimes [0,0,0\ \dots,\ 0] = c \Rightarrow L\big(y,\ c\big)$$

is constant with respect to $\tilde{\Theta}$

Gradient suppression is:

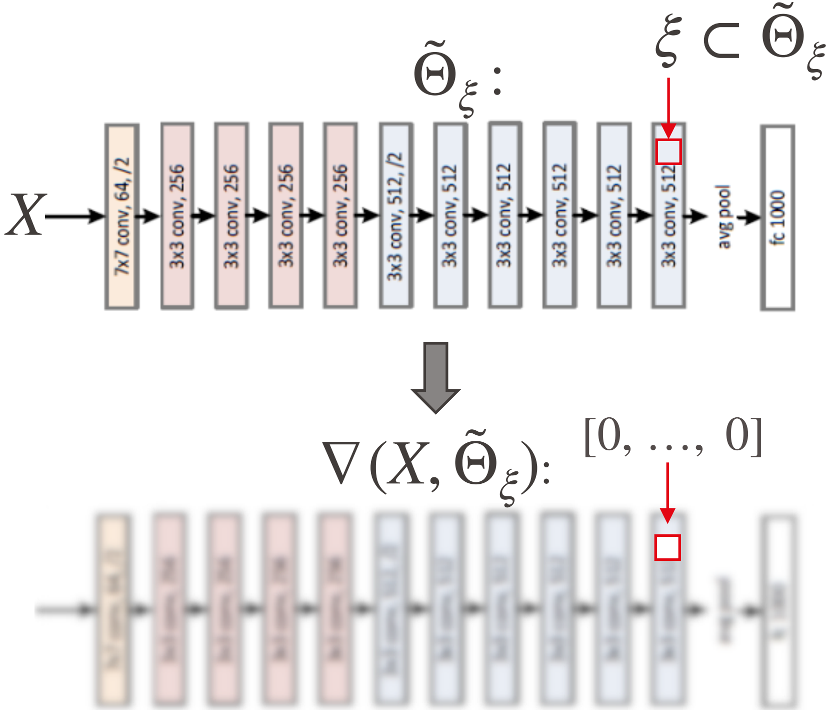
- **Attack properties:**
  - It works even with millions of users (e.g., real-world cross-device FL)
  - It is task/network-agnostic (it would work for any NN/task)
  - It does not require any auxiliary information on users

- **However:**
  - It is trivially detectable by aware non-target users

- We introduce a more sophisticated attack called "**Canary Gradient Attack**":

Partial Gradient Suppression:

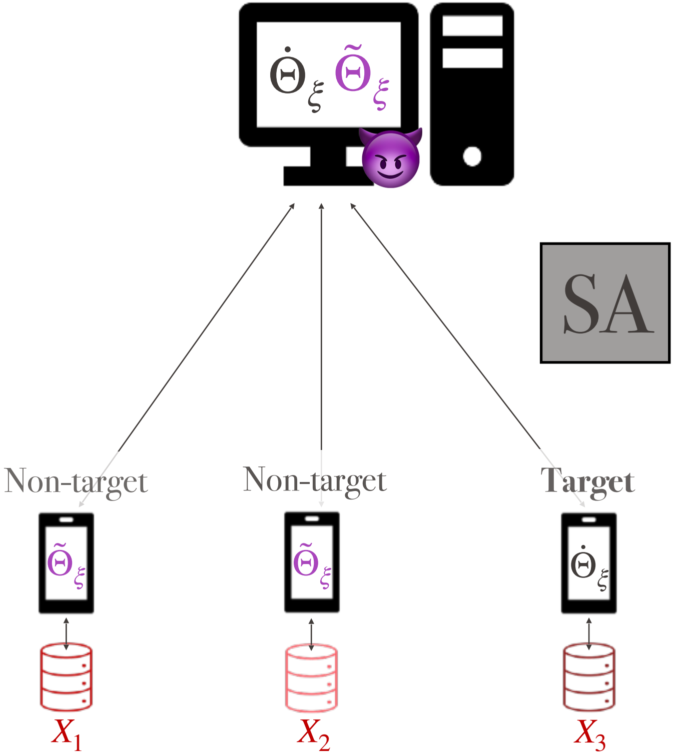
Idea: we forge $\tilde{\Theta}_\xi$ s.t. only a small subset of parameters ξ gets zero-gradient:



e.g., ratio of zeroed gradients for a ResNet18:

$$\frac{|\xi|}{|\tilde{\Theta}_\xi|} = \frac{2}{3{,}439{,}332} = 5 \cdot 10^{-7}\%$$

Canary-Gradient attack (setup):

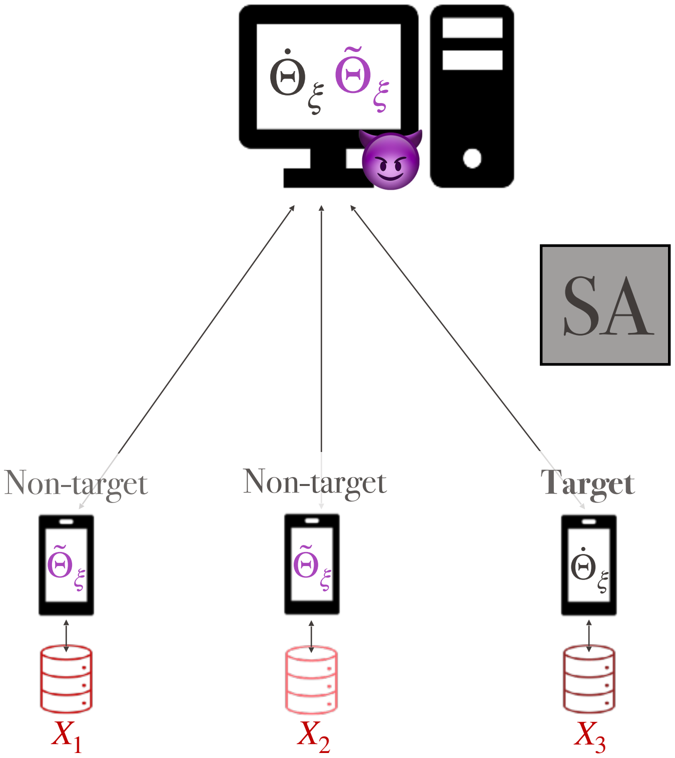


Attack setup:

- The **non-targets** get $\tilde{\Theta}_\xi$; **that is, gradient for** ξ **is always zero.**
- The **target** gets $\dot{\Theta}_\xi$; that is, gradient for ξ can be either non-zero or zero **conditionally to the input** X **used to compute the gradient by the target.**
  - E.g., Membership Inference Attack: $x_t \in X$?

$\nabla(X, \dot{\Theta}_\xi)$:

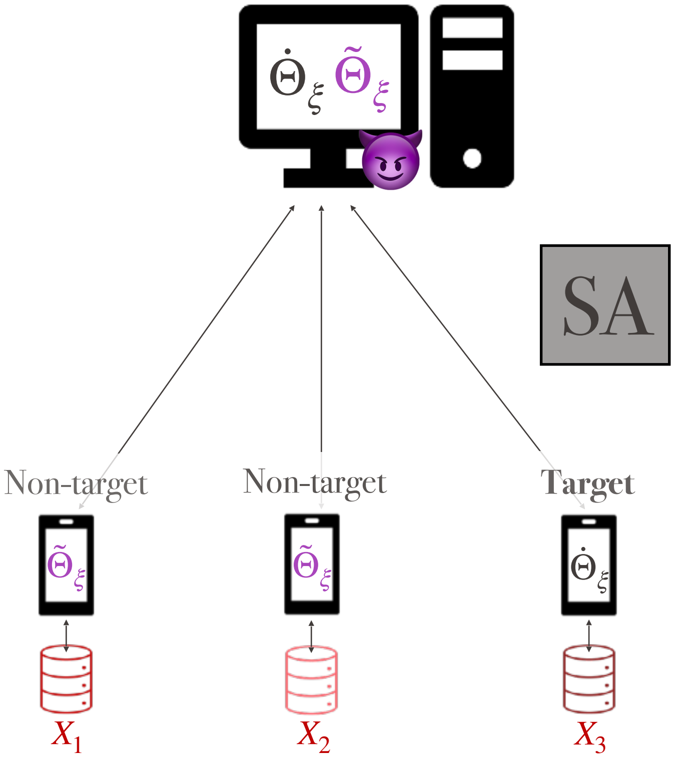
Canary-Gradient attack (setup):



Attack setup:

- The **non-targets** get $\tilde{\Theta}_\xi$; that is, **gradient for ξ is always zero.**
- The **target** gets $\dot{\Theta}_\xi$; that is, gradient for ξ can be either non-zero 💡 or zero 💡 **conditionally to the input X used to compute the gradient by the target.**
  - E.g., Membership Inference Attack: $x_t \in X$?

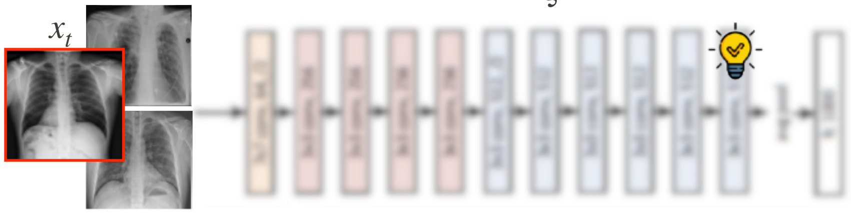
Canary-Gradient attack (setup):



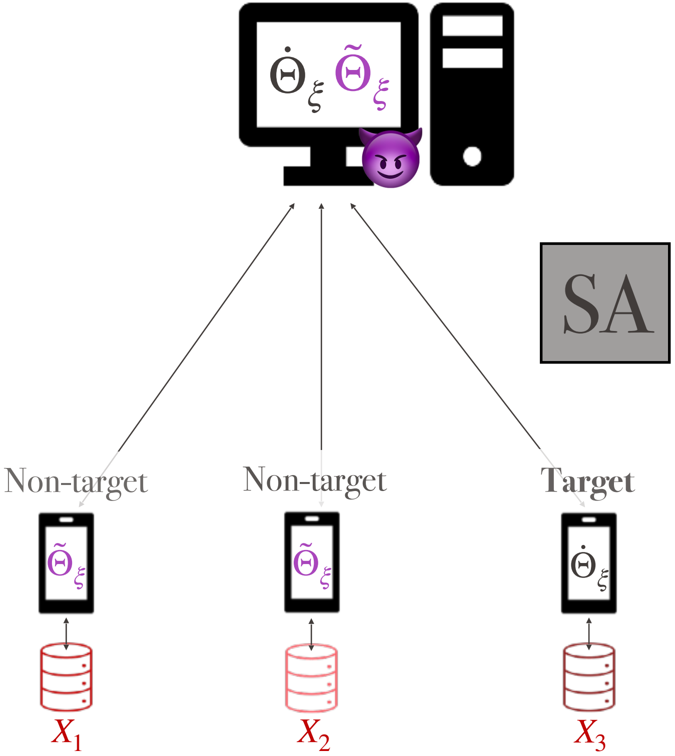
Attack setup:

- The **non-targets** get $\tilde{\Theta}_\xi$; that is, **gradient for ξ is always zero.**
- The **target** gets $\dot{\Theta}_\xi$; that is, gradient for ξ can be either non-zero 💡 or zero 💡 **conditionally to the input X used to compute the gradient by the target.**
  - E.g., Membership Inference Attack: $x_t \in X$?

$\nabla(X, \dot{\Theta}_\xi)$:



Canary-Gradient attack (setup):

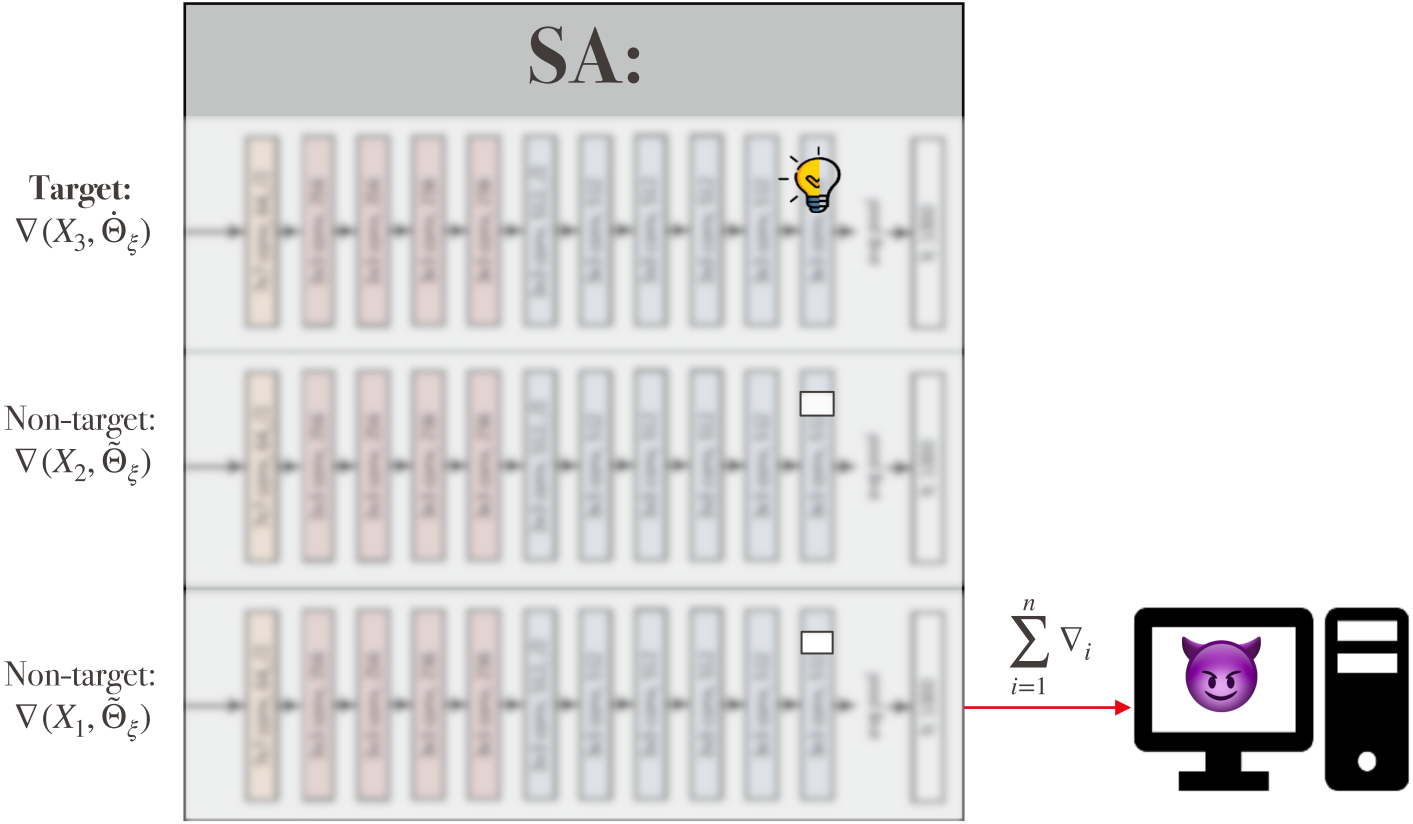


Attack setup:

- The **non-targets** get $\tilde{\Theta}_\xi$; that is, **gradient for ξ is always zero.**
- The **target** gets $\dot{\Theta}_\xi$; that is, gradient for ξ can be either non-zero 💡 or zero 💡 **conditionally to the input X used to compute the gradient by the target.**
  - E.g., Membership Inference Attack: $x_t \in X$?

$$\nabla(X, \dot{\Theta}_\xi):$$

MIAs via Canary-Gradient attack (aftermath):



Target:
$\nabla(X_3, \dot{\Theta}_\xi)$

Non-target:
$\nabla(X_2, \tilde{\Theta}_\xi)$

Non-target:
$\nabla(X_1, \tilde{\Theta}_\xi)$

$$\sum_{i=1}^{n} \nabla_i$$

MIAs via Canary-Gradient attack (aftermath):

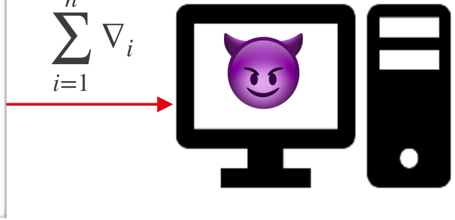
SA:

Target:
$\nabla(X_3, \dot{\Theta}_\xi)$

Non-target:
$\nabla(X_2, \tilde{\Theta}_\xi)$

Non-target:
$\nabla(X_1, \tilde{\Theta}_\xi)$

$\sum_{i=1}^{n} \nabla_i$

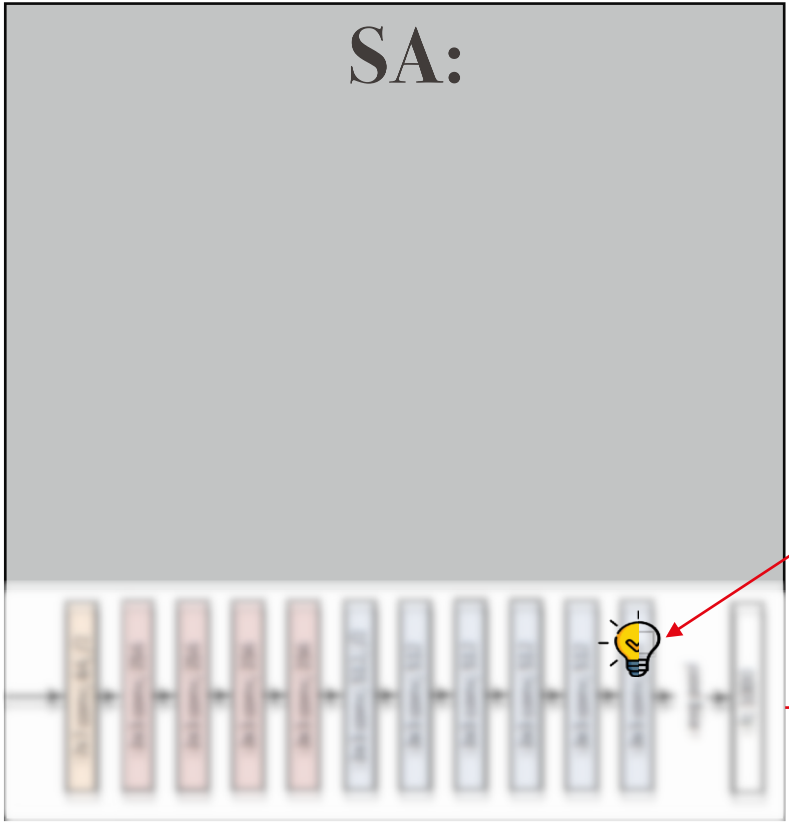


MIAs via Canary-Gradient attack (aftermath):

Target:
$\nabla(X_3, \dot{\Theta}_\xi)$

Non-target:
$\nabla(X_2, \tilde{\Theta}_\xi)$

Non-target:
$\nabla(X_1, \tilde{\Theta}_\xi)$



The attacker recovers the exact gradient ξ for the **target**, then:

💡 $\rightarrow x_t \in X_3$

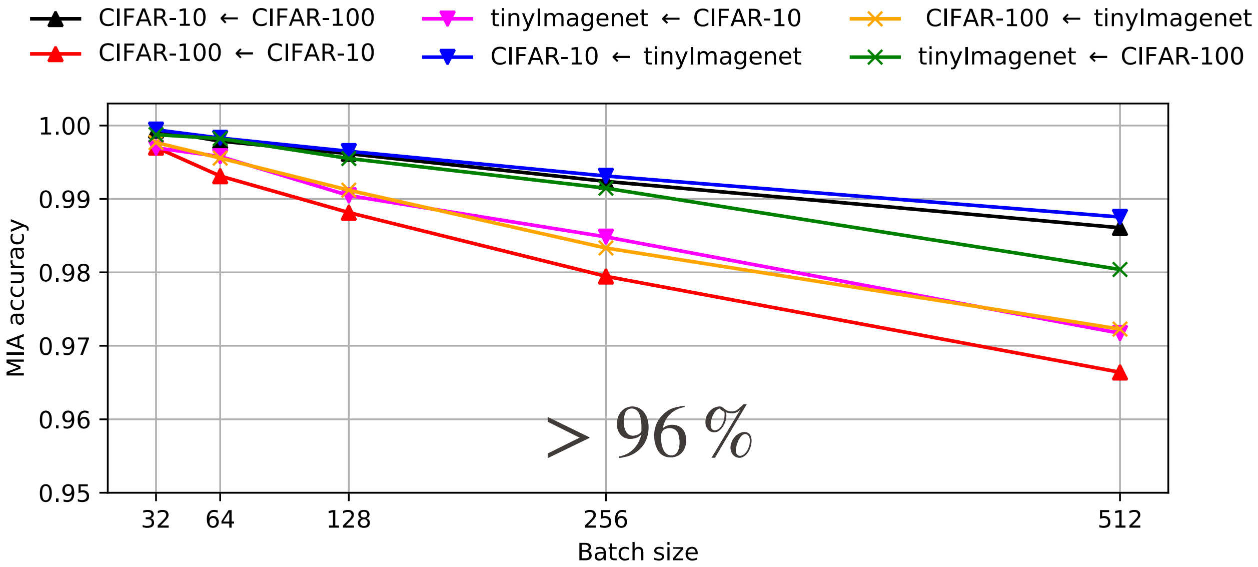
○ $\rightarrow x_t \notin X_3$

$\sum_{i=1}^{n} \nabla_i$

MIAs via Canary-Gradient attack on FedSGD:

New state-of-the-art MIA in FL (malicious server) that works under SA:

- Canary-Gradient injected in a ResNet18
- Only 2 parameters for ξ:
  - The scale and shift parameters of a single channel in a normalization layer
    - i.e., $5 \cdot 10^{-7}\%$ of the total parameters in the network

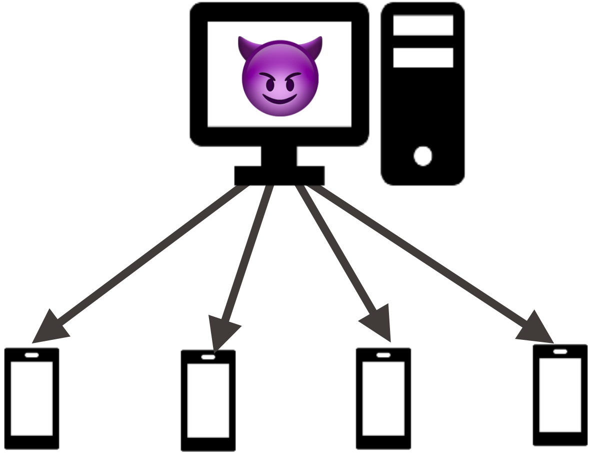


$> 96\,\%$

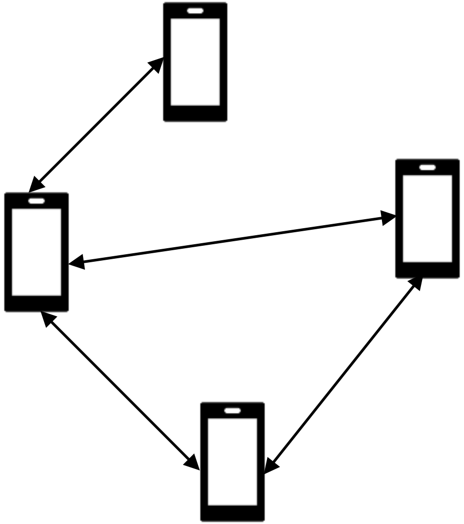
P2P Federated Learning

“Ok, it’s clear now; the problem is the server!”

We go fully-decentralized; Welcome to **Decentralized Machine Learning**:



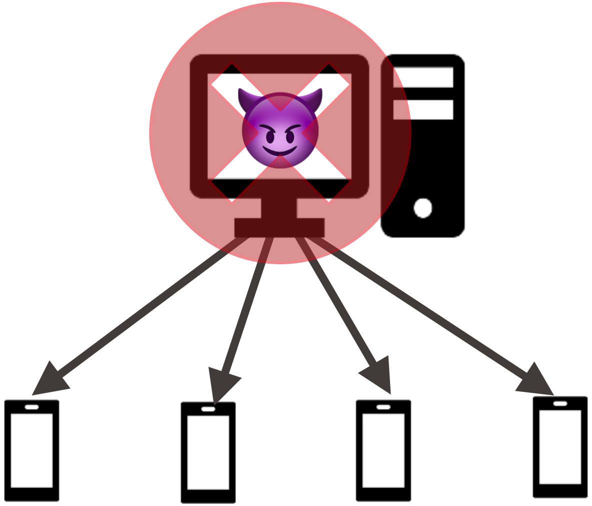
Peer-to-Peer: Communication:



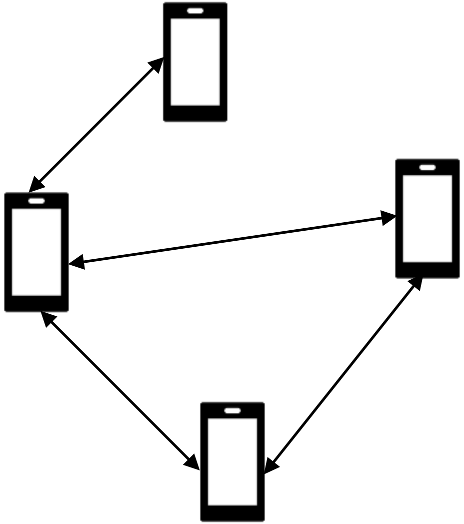
Lalitha et al “Peer-to-peer Federated Learning on Graphs” 2019
Guha Roy et al “BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learnig” 2019
….. and many others ….

"Ok, it's clear now; the problem is the server!"

We go fully-decentralized; Welcome to **Decentralized Machine Learning**:

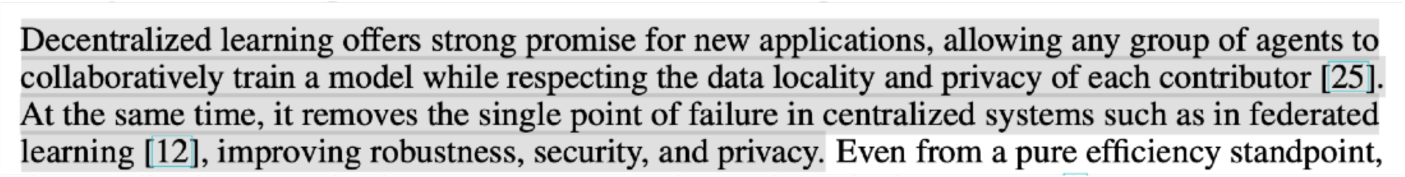


Peer-to-Peer: Communication:



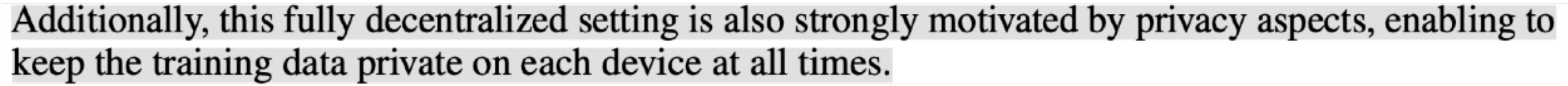
Lalitha et al "Peer-to-peer Federated Learning on Graphs" 2019
Guha Roy et al "BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learnig" 2019
….. and many others ….

"Ok, it's clear now; the problem is the server!"



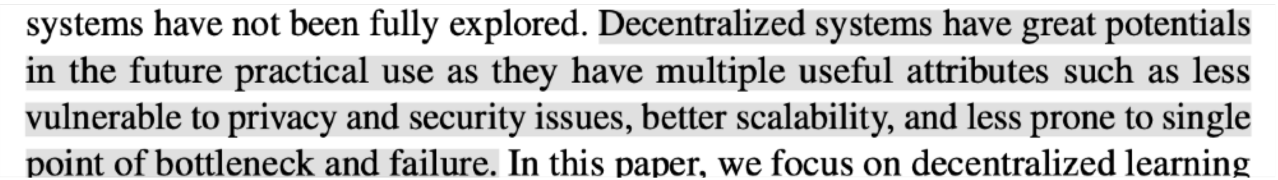
Decentralized learning offers strong promise for new applications, allowing any group of agents to collaboratively train a model while respecting the data locality and privacy of each contributor [25]. At the same time, it removes the single point of failure in centralized systems such as in federated learning [12], improving robustness, security, and privacy. Even from a pure efficiency standpoint,

[RelaySum for Decentralized Deep Learning on Heterogeneous Data, NeurIPS 2021]



Additionally, this fully decentralized setting is also strongly motivated by privacy aspects, enabling to keep the training data private on each device at all times.

[Decentralized Deep Learning With Arbitrary Communication Compression ICLR 2020]

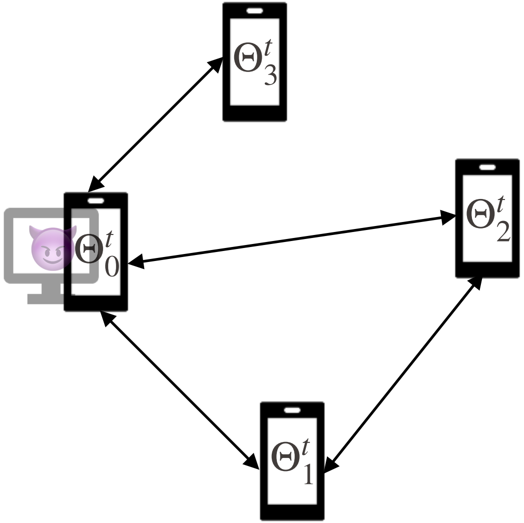


systems have not been fully explored. Decentralized systems have great potentials in the future practical use as they have multiple useful attributes such as less vulnerable to privacy and security issues, better scalability, and less prone to single point of bottleneck and failure. In this paper, we focus on decentralized learning

[Towards Decentralized Deep Learning with Differential Privacy CLOUD 2019]

Lalitha et al "Peer-to-peer Federated Learning on Graphs" 2019
Guha Roy et al "BrainTorrent: A Peer-to-Peer Environment for Decentralized Federated Learnig" 2019
….. and many others ….

Does decentralization make things better?



Pasquini, Raynal, and Troncoso "*On the (In)security of Peer-to-Peer Decentralized Machine Learning*" IEEE S&P'23

Is it the case?

- 🔍We perform a thorough security (privacy, mainly) analysis of the protocol:
  - Both Semi-honest (😏) & Malicious security (😈).

    (we introduce 6 new attacks)

Is it the case?

- 🔍We perform a thorough security (privacy, mainly) analysis of the protocol:
  - Both Semi-honest (😏) & Malicious security (😈).
    (we introduce 6 new attacks)
- ⚠️No! **DL protocols inherently boost adversaries' capabilities, resulting in less privacy for the users.**

Is it the case?

- 🔍We perform a thorough security (privacy, mainly) analysis of the protocol:
  - Both Semi-honest (😏) & Malicious security (😈). (we introduce 6 new attacks)
- ⚠️ **No! DL protocols inherently boost adversaries' capabilities, resulting in less privacy for the users.**
- 🔬We characterize the main factors responsible for DL insecurity:

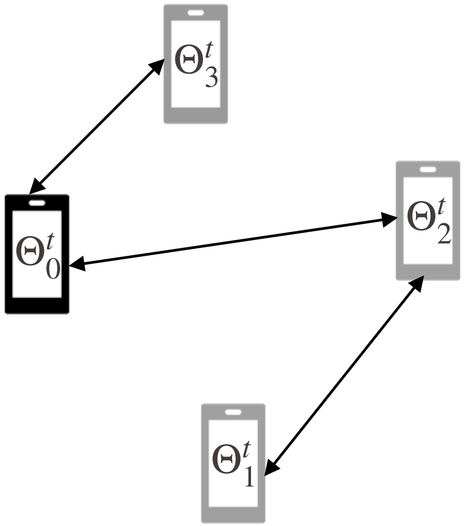
Is it the case?

- 🔍We perform a thorough security (privacy, mainly) analysis of the protocol:
  - Both Semi-honest (😏) & Malicious security (😈).
    (we introduce 6 new attacks)
- ⚠️ **No! DL protocols inherently boost adversaries' capabilities, resulting in less privacy for the users.**
- 🔬We characterize the main factors responsible for DL insecurity:

Local Generalization🔪 🗡️Adv. System Knowledge

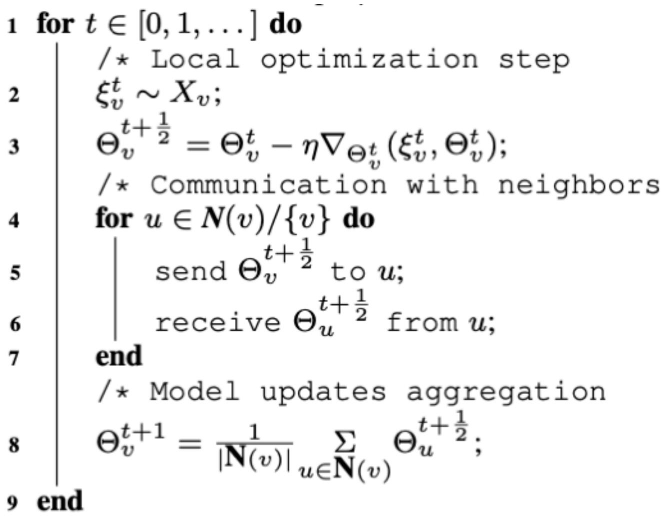
Decentralized Learning:

- Every user picks a set of neighbors users.
- Then, every node simultaneously:

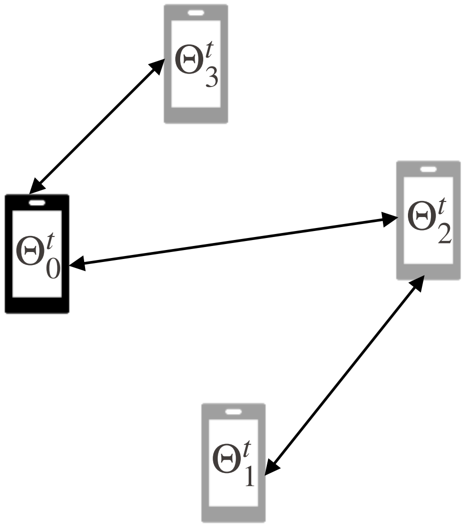


Decentralized Learning:

- Every user picks a set of neighbors users.
- Then, every node simultaneously:



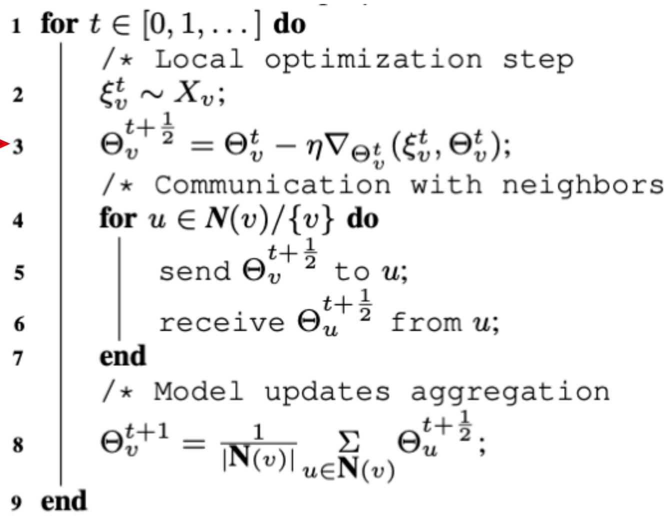
```
for $t \in [0, 1, \dots]$ do
    /* Local optimization step
    $\xi_v^t \sim X_v$;
    $\Theta_v^{t+\frac{1}{2}} = \Theta_v^t - \eta \nabla_{\Theta_v^t}(\xi_v^t, \Theta_v^t)$;
    /* Communication with neighbors
    for $u \in N(v)/\{v\}$ do
        send $\Theta_v^{t+\frac{1}{2}}$ to $u$;
        receive $\Theta_u^{t+\frac{1}{2}}$ from $u$;
    end
    /* Model updates aggregation
    $\Theta_v^{t+1} = \frac{1}{|\mathbf{N}(v)|} \sum_{u \in \mathbf{N}(v)} \Theta_u^{t+\frac{1}{2}}$;
end
```



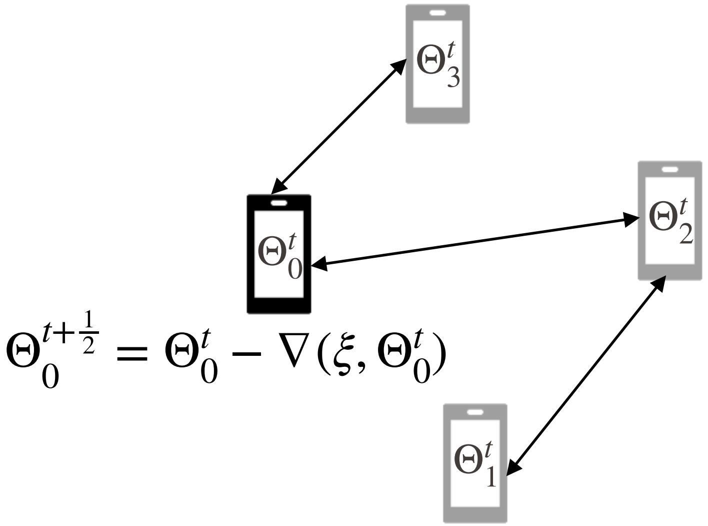
Note: Every node may have a different set of parameters.

Decentralized Learning:

- Every user picks a set of neighbors users.
- Then, every node simultaneously:



```
for $t \in [0, 1, \dots]$ do
    /* Local optimization step
    $\xi_v^t \sim X_v$;
    $\Theta_v^{t+\frac{1}{2}} = \Theta_v^t - \eta \nabla_{\Theta_v^t} (\xi_v^t, \Theta_v^t)$;
    /* Communication with neighbors
    for $u \in N(v)/\{v\}$ do
        send $\Theta_v^{t+\frac{1}{2}}$ to $u$;
        receive $\Theta_u^{t+\frac{1}{2}}$ from $u$;
    end
    /* Model updates aggregation
    $\Theta_v^{t+1} = \frac{1}{|\mathbf{N}(v)|} \sum_{u \in \mathbf{N}(v)} \Theta_u^{t+\frac{1}{2}}$;
end
```

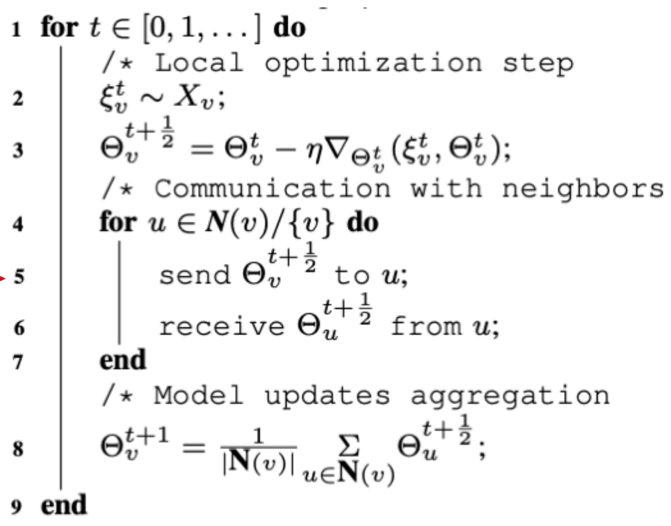


$$\Theta_0^{t+\frac{1}{2}} = \Theta_0^t - \nabla(\xi, \Theta_0^t)$$

Note: Every node may have a different set of parameters.

Decentralized Learning:

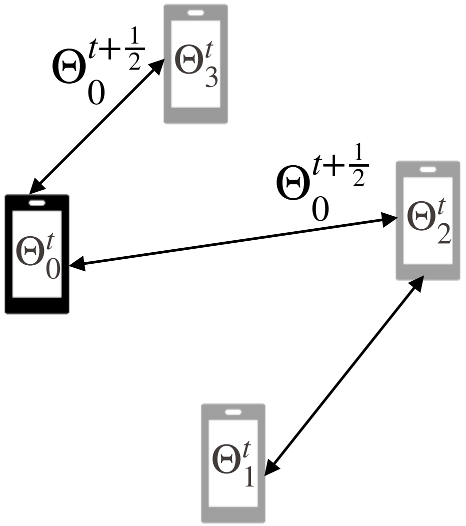
- Every user picks a set of neighbors users.
- Then, every node simultaneously:



```
for $t \in [0, 1, \ldots]$ do
    /* Local optimization step
    $\xi_v^t \sim X_v$;
    $\Theta_v^{t+\frac{1}{2}} = \Theta_v^t - \eta\nabla_{\Theta_v^t}(\xi_v^t, \Theta_v^t)$;
    /* Communication with neighbors
    for $u \in N(v)/\{v\}$ do
        send $\Theta_v^{t+\frac{1}{2}}$ to $u$;
        receive $\Theta_u^{t+\frac{1}{2}}$ from $u$;
    end
    /* Model updates aggregation
    $\Theta_v^{t+1} = \frac{1}{|\mathbf{N}(v)|} \sum_{u \in \mathbf{N}(v)} \Theta_u^{t+\frac{1}{2}}$;
end
```

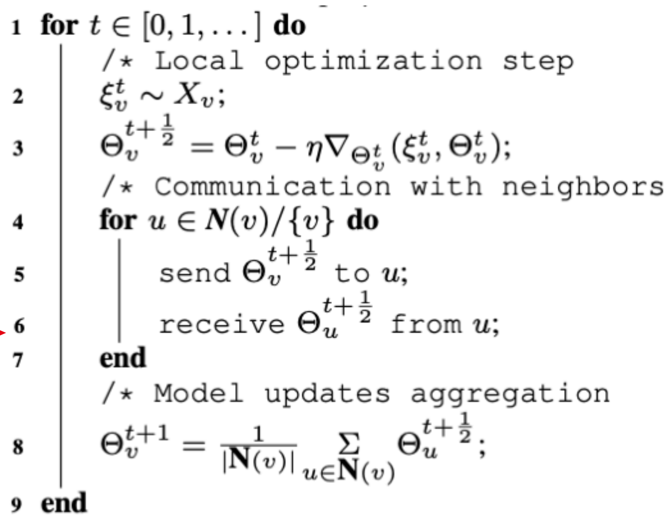
Note: Every node may have a different set of parameters.

You can also see it as a "Gossip protocol":



Decentralized Learning:

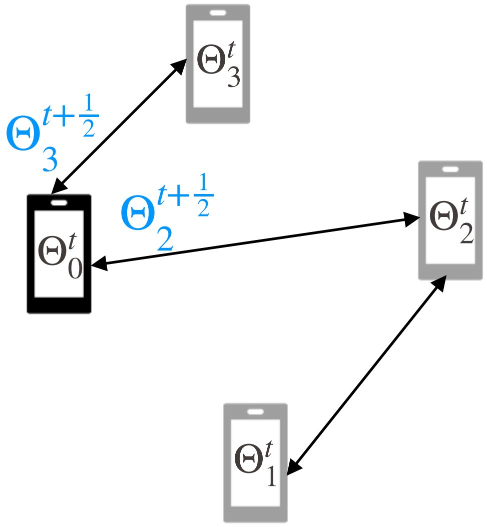
- Every user picks a set of neighbors users.
- Then, every node simultaneously:



```
for $t \in [0, 1, \dots]$ do
    /* Local optimization step
    $\xi_v^t \sim X_v$;
    $\Theta_v^{t+\frac{1}{2}} = \Theta_v^t - \eta \nabla_{\Theta_v^t}(\xi_v^t, \Theta_v^t)$;
    /* Communication with neighbors
    for $u \in N(v)/\{v\}$ do
        send $\Theta_v^{t+\frac{1}{2}}$ to $u$;
        receive $\Theta_u^{t+\frac{1}{2}}$ from $u$;
    end
    /* Model updates aggregation
    $\Theta_v^{t+1} = \frac{1}{|\mathbf{N}(v)|} \sum_{u \in \mathbf{N}(v)} \Theta_u^{t+\frac{1}{2}}$;
end
```

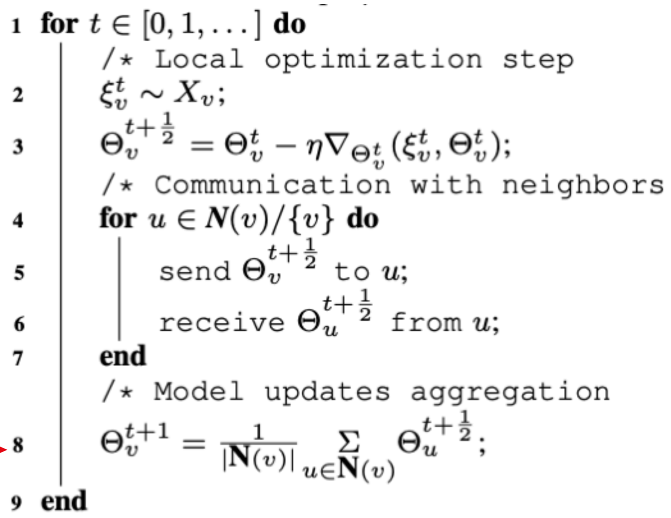
Note: Every node may have a different set of parameters.

You can also see it as a “Gossip protocol”:



Decentralized Learning:

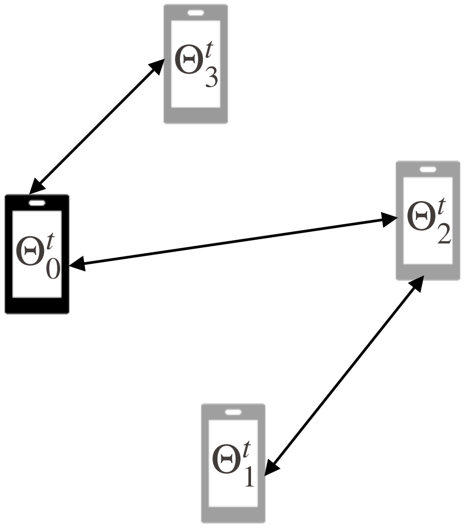
- Every user picks a set of neighbors users.
- Then, every node simultaneously:



```
for $t \in [0, 1, \dots]$ do
    /* Local optimization step
    $\xi_v^t \sim X_v$;
    $\Theta_v^{t+\frac{1}{2}} = \Theta_v^t - \eta \nabla_{\Theta_v^t}(\xi_v^t, \Theta_v^t)$;
    /* Communication with neighbors
    for $u \in N(v)/\{v\}$ do
        send $\Theta_v^{t+\frac{1}{2}}$ to $u$;
        receive $\Theta_u^{t+\frac{1}{2}}$ from $u$;
    end
    /* Model updates aggregation
    $\Theta_v^{t+1} = \frac{1}{|\mathbf{N}(v)|} \sum_{u \in \mathbf{N}(v)} \Theta_u^{t+\frac{1}{2}}$;
end
```

Note: Every node may have a different set of parameters.

You can also see it as a "Gossip protocol":



Update local parameters:

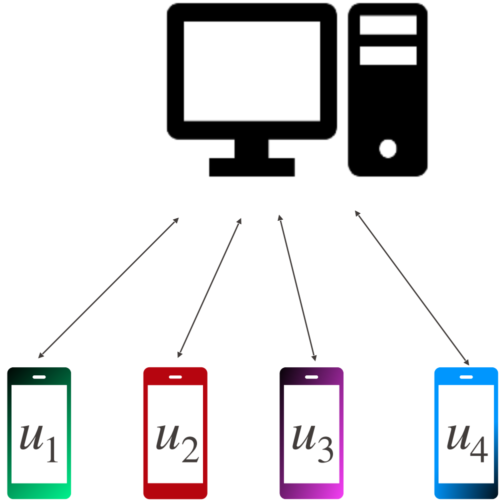
$$\Theta_0^{t+1} = (\Theta_0^{t+\frac{1}{2}} + \Theta_2^{t+\frac{1}{2}} + \Theta_3^{t+\frac{1}{2}})/3$$

Local Generalization 🔪

The Local Generalization phenomenon:

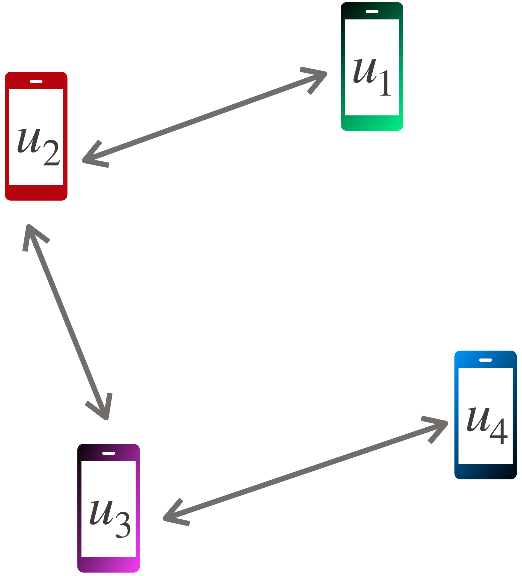
Federated Learning (FL):

- Every user shares the **same** model.



Decentralized Learning (DL):

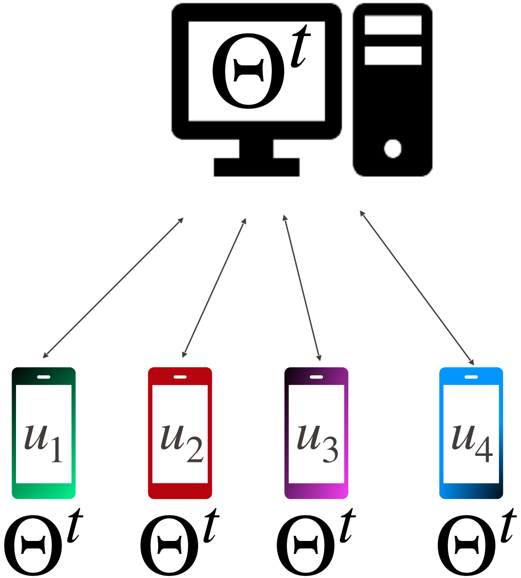
- Every user has **different** local parameters.



The Local Generalization phenomenon:

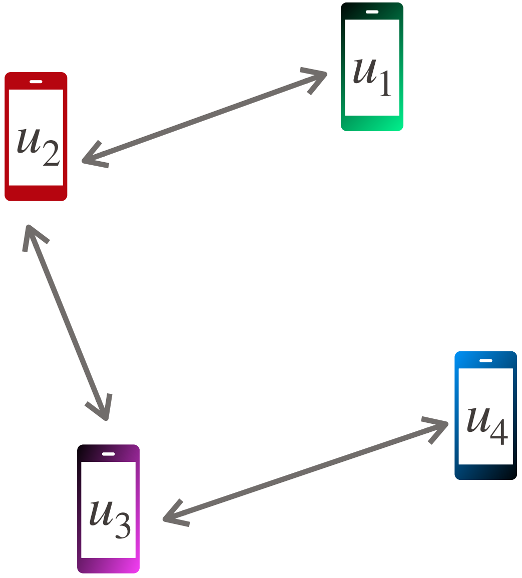
Federated Learning (FL):

- Every user shares the **same** model.



Decentralized Learning (DL):

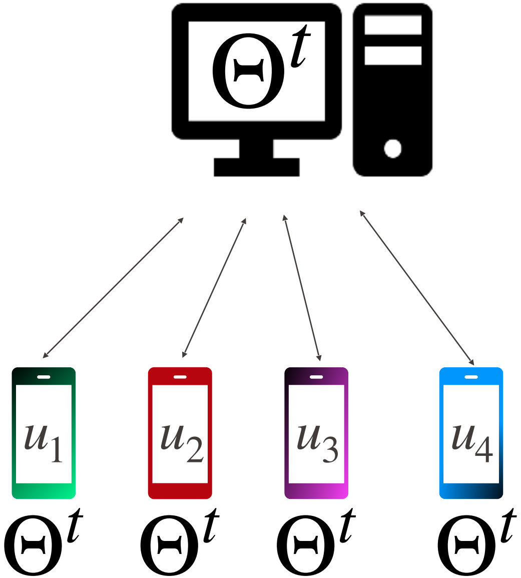
- Every user has **different** local parameters.



The Local Generalization phenomenon:

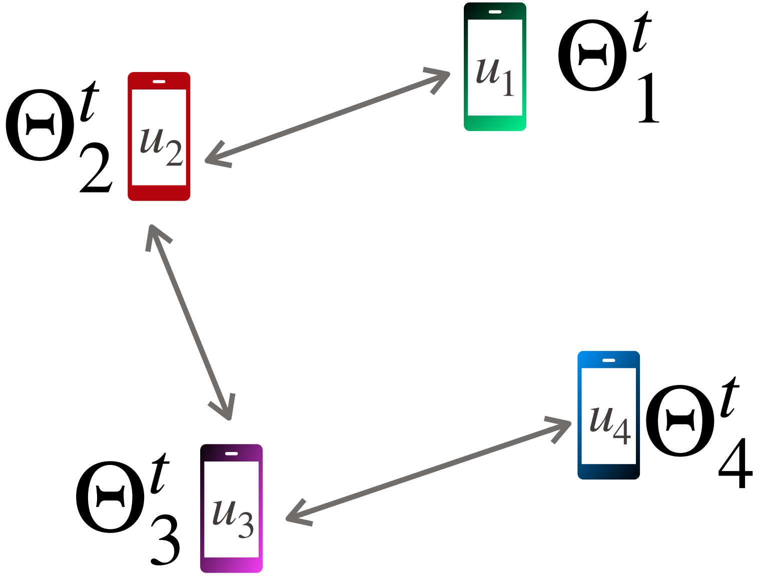
Federated Learning (FL):

- Every user shares the **same** model.



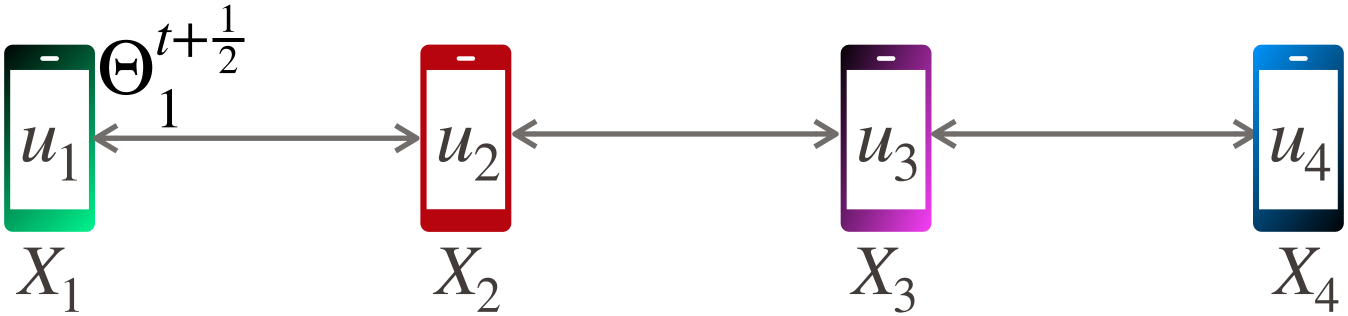
Decentralized Learning (DL):

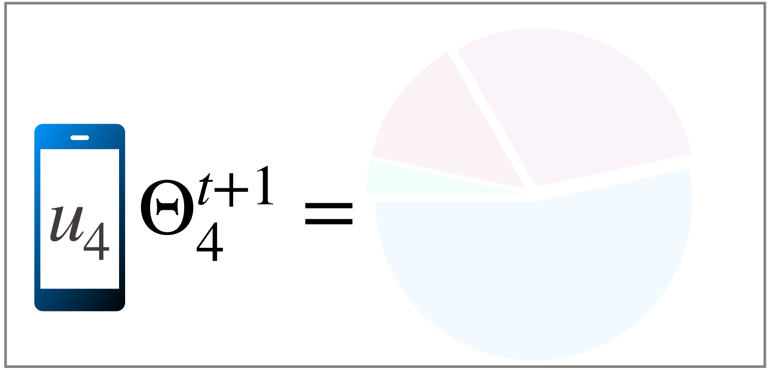
- Every user has **different** local parameters.



Gossip communication and Generalization:

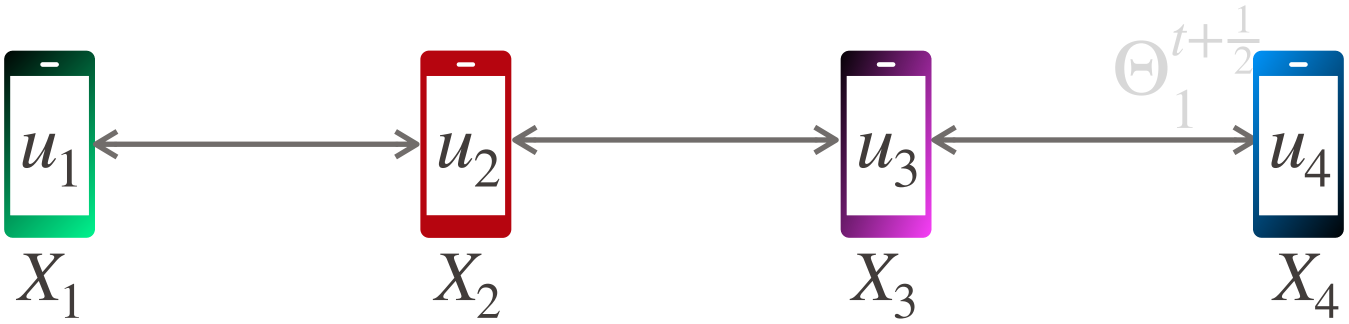
- Gossip communication induces uneven generalization:

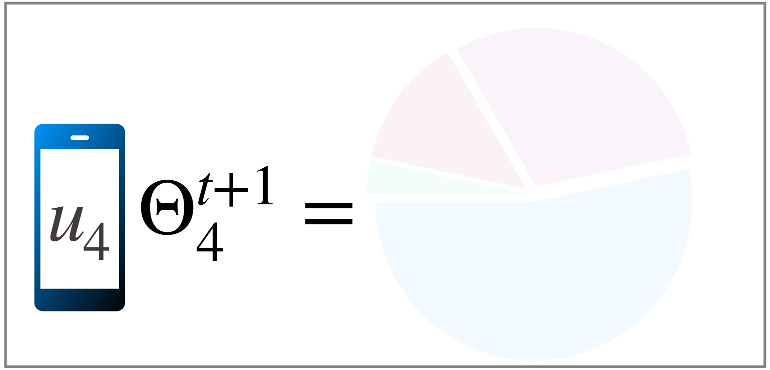




Gossip communication and Generalization:

- Gossip communication induces uneven generalization:

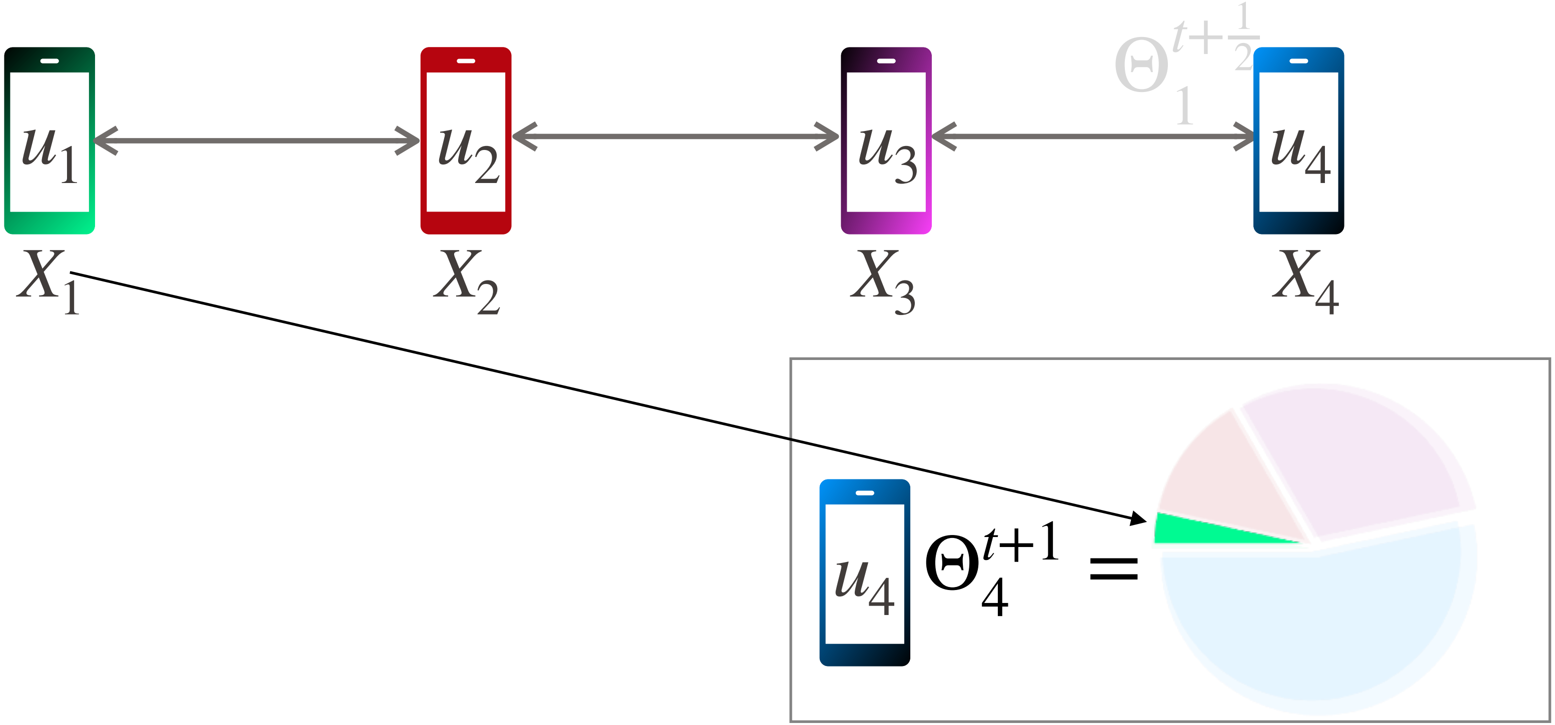




$$\Theta_4^{t+1} =$$

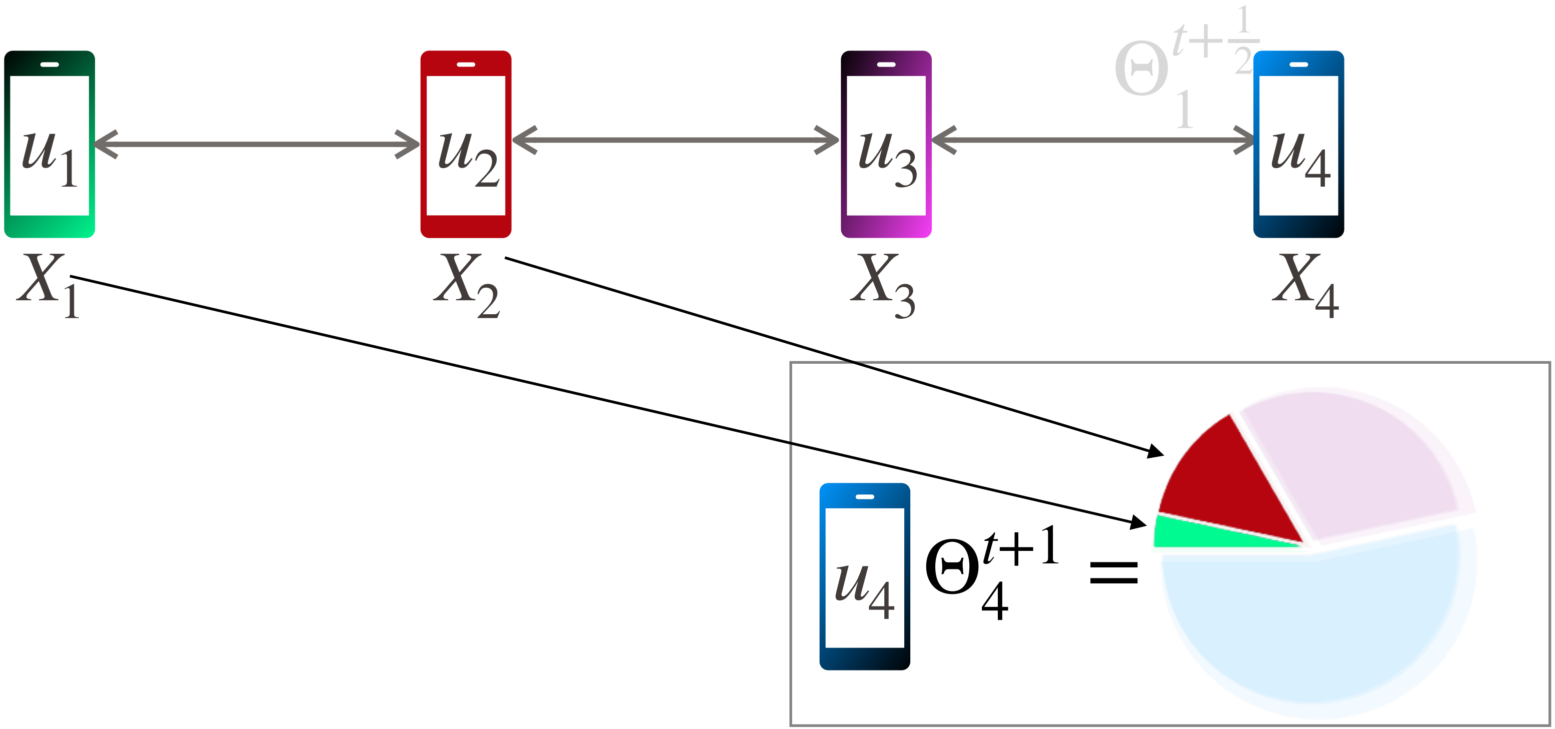
Gossip communication and Generalization:

- Gossip communication induces uneven generalization:



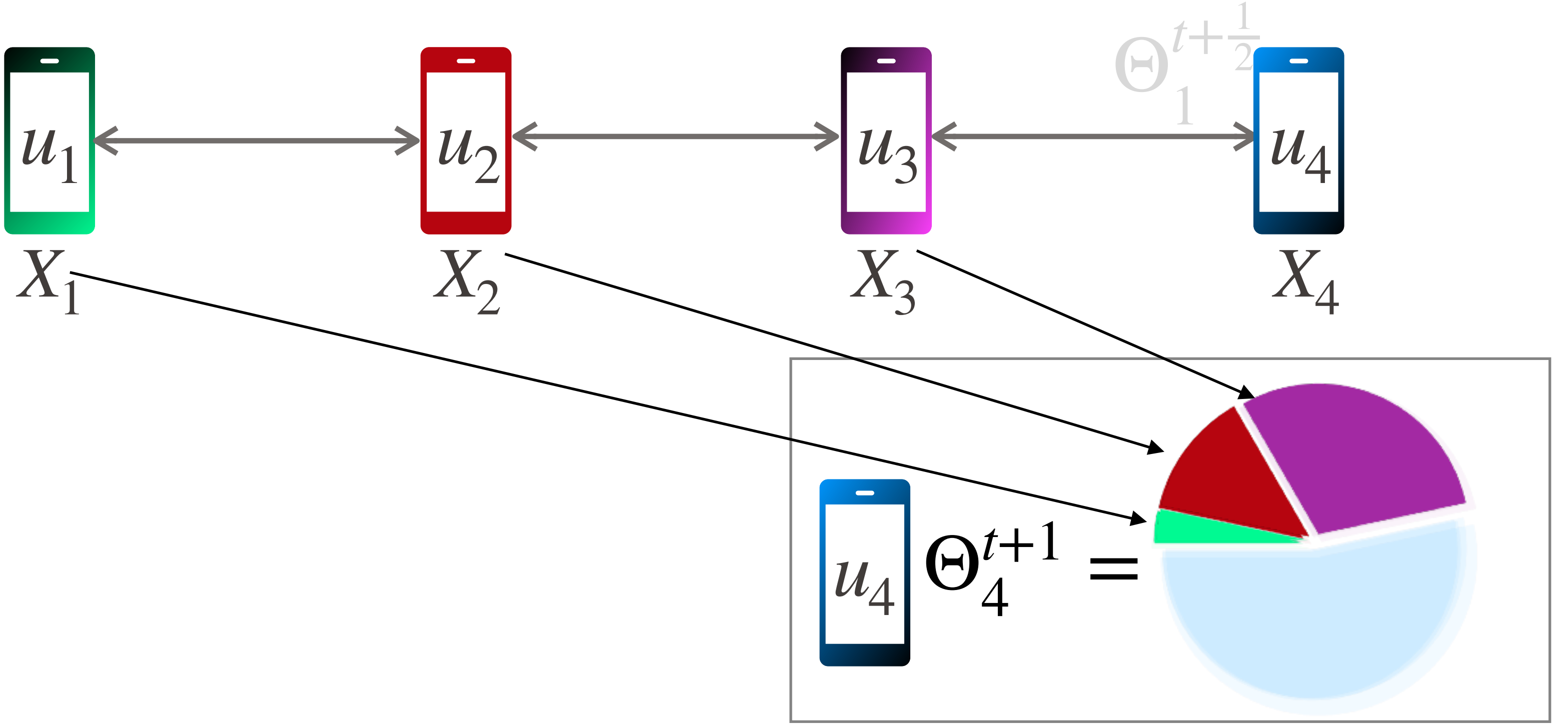
Gossip communication and Generalization:

- Gossip communication induces uneven generalization:



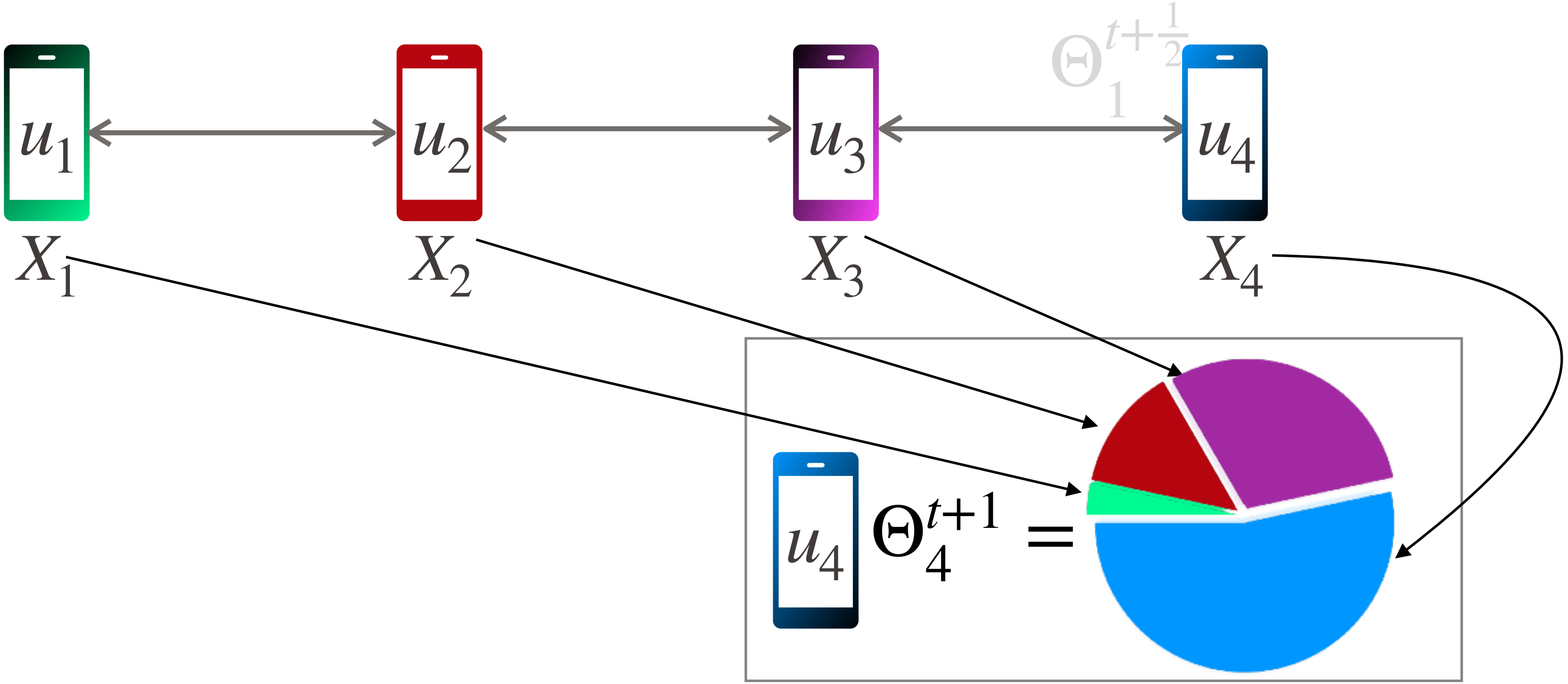
Gossip communication and Generalization:

- Gossip communication induces uneven generalization:



Gossip communication and Generalization:

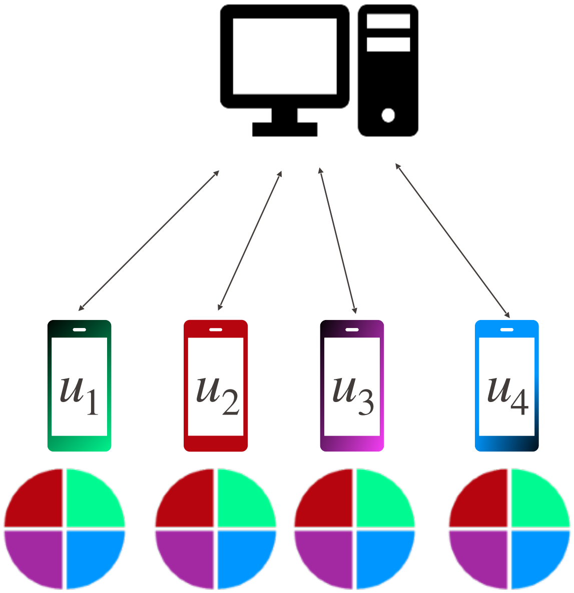
- Gossip communication induces uneven generalization:



Local Generalization:

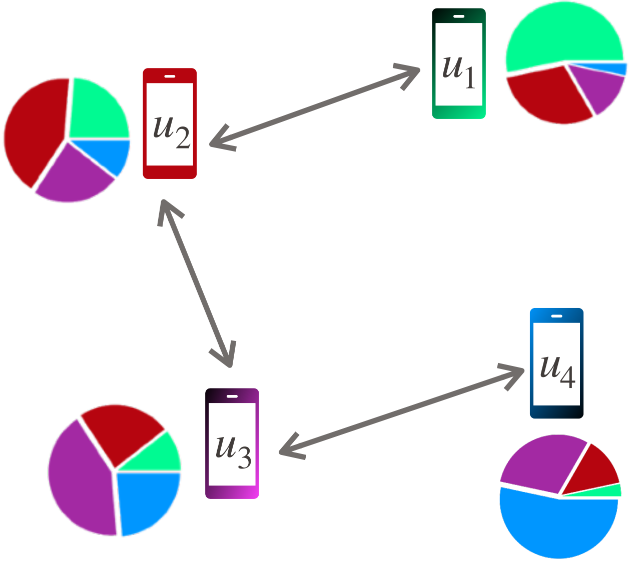
Federated Learning (FL):

- Every node contributes equally to the global model.



Decentralized Learning (DL):

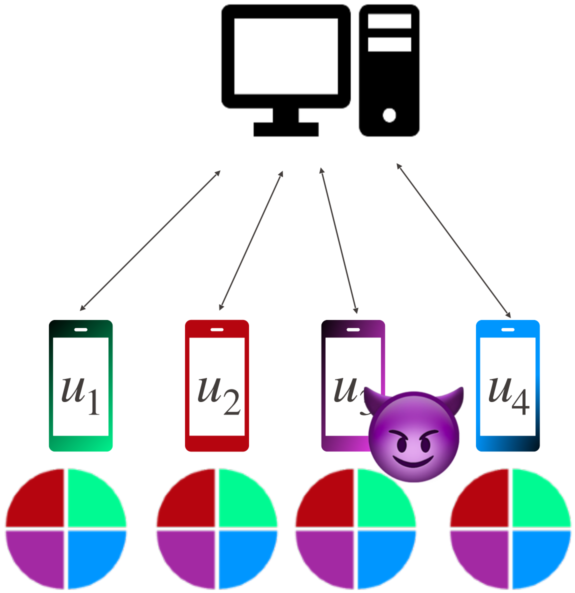
- Nodes' local models are **dominated** by their own **local data**.



Local Generalization:

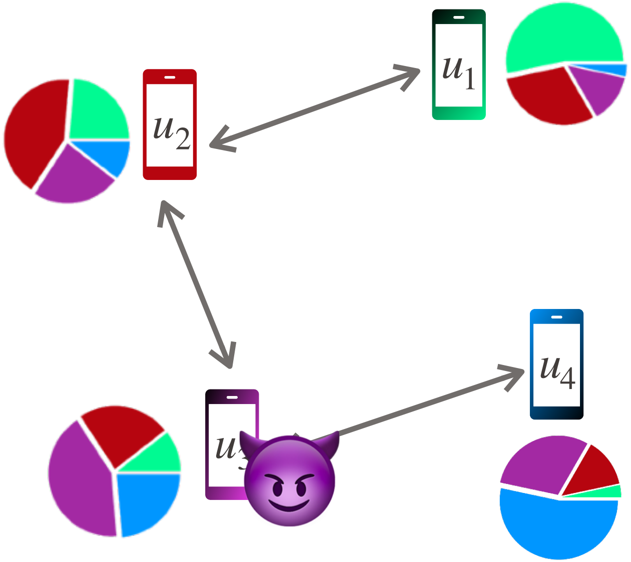
Federated Learning (FL):

- Every node contributes equally to the global model.



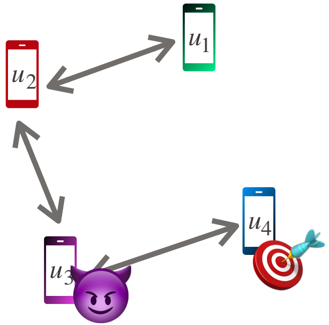
Decentralized Learning (DL):

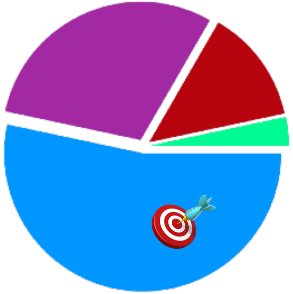
- Nodes' local models are **dominated** by their own **local data**.



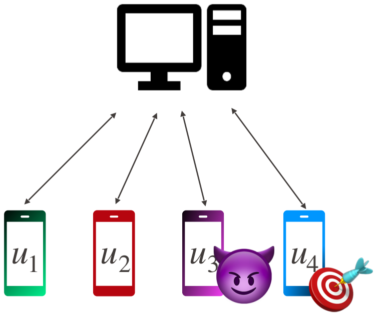
Generalization is Privacy [MIA]:

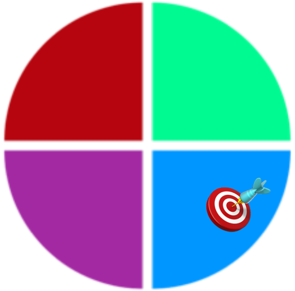
- Attack on **DL** model update:





- Attack on FL model update:
  (Global model)



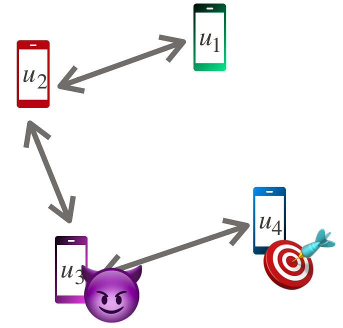


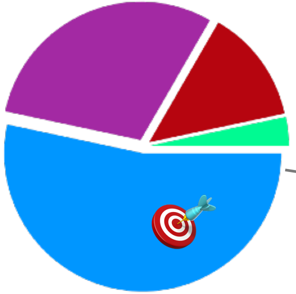
Membership attack success:

Higher means less privacy

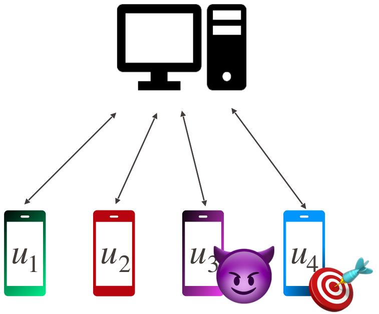
Generalization is Privacy [MIA]:

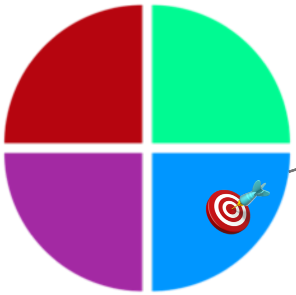
- Attack on **DL** model update:



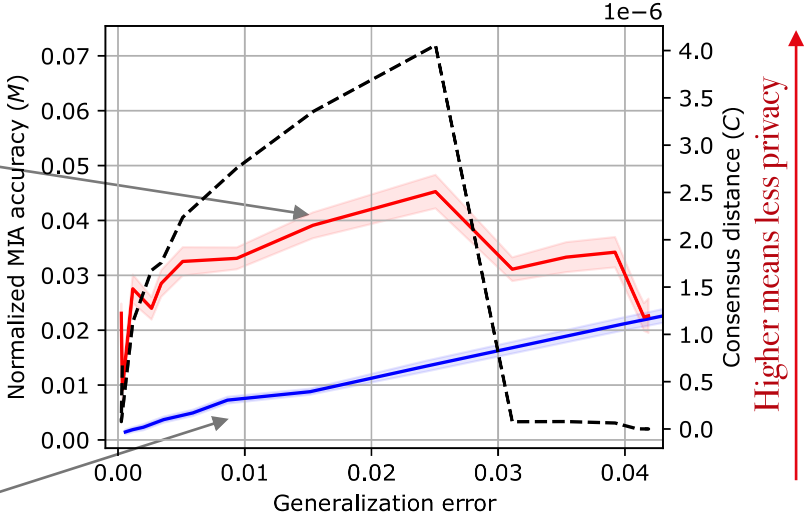


- Attack on FL model update:
  (Global model)



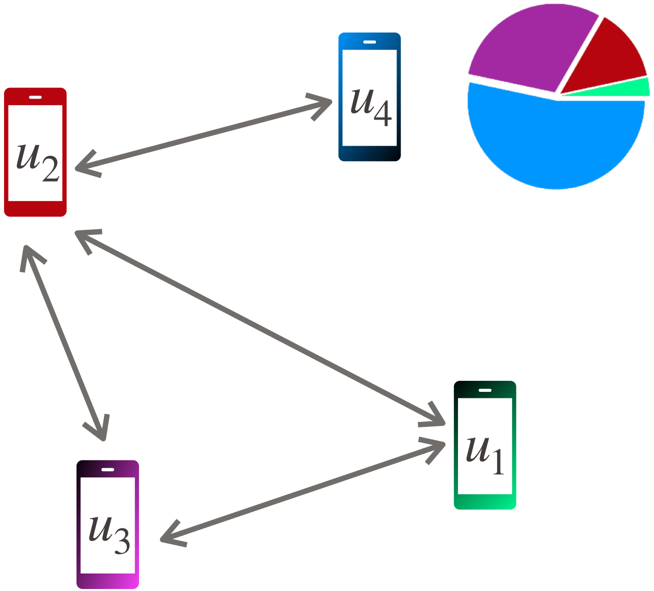


Membership attack success:



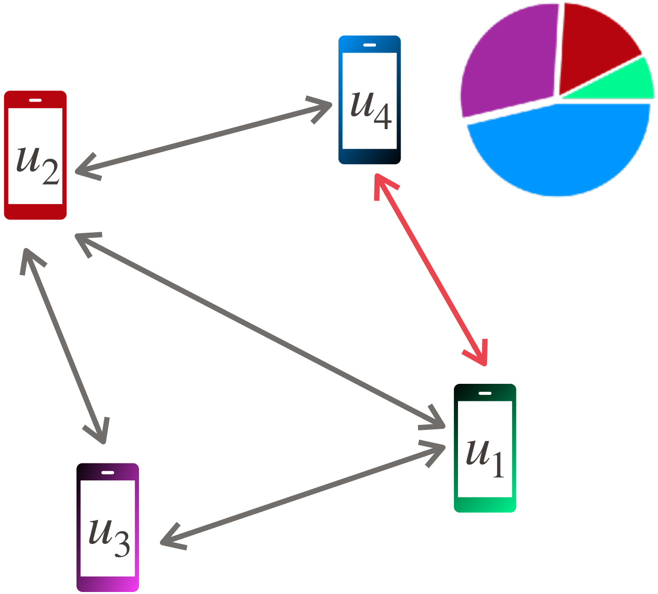
Setup: Torus-36, CIFAR-100, ResNet-20

Reducing Local Generalization:



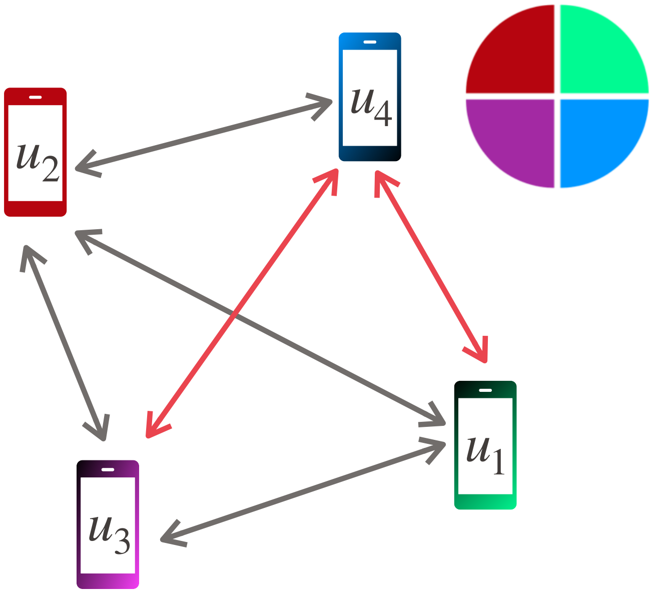
- Dense topologies reduce local generalization.

Reducing Local Generalization:



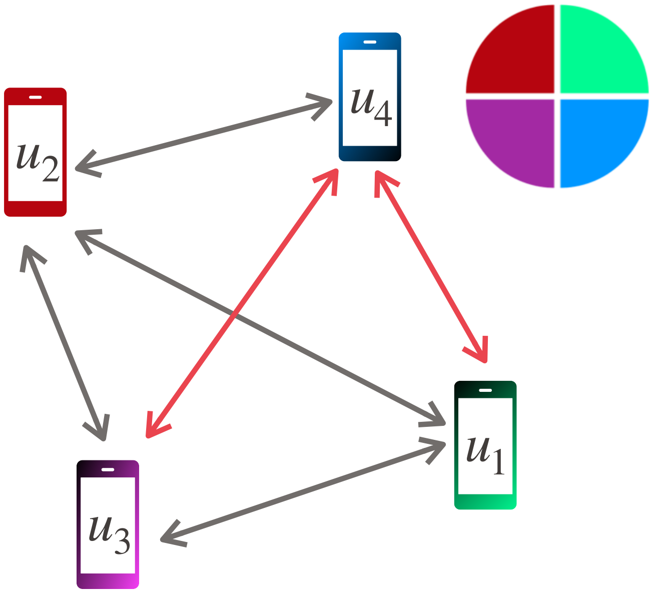
- Dense topologies reduce local generalization.

Reducing Local Generalization:



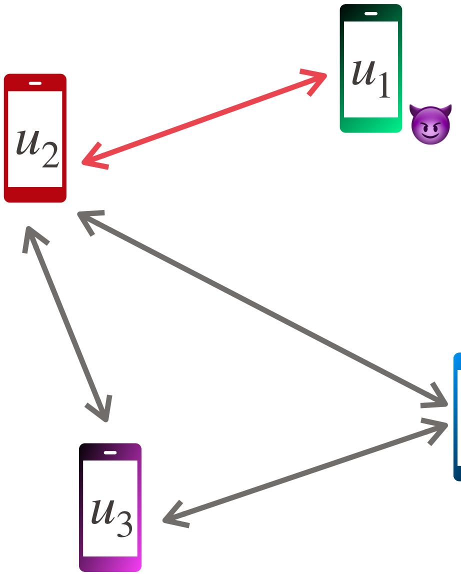
- **Dense topologies reduce local generalization.**
- When the topology is fully-connected:
  - No more local generalization phenomenon! (DL becomes equivalent to FL)

Reducing Local Generalization:



- **Dense topologies reduce local generalization.**
- When the topology is fully-connected:
  - No more local generalization phenomenon! (DL becomes equivalent to FL)
- ⚖️ However:
  - 👔 **Efficiency:** Every node is now a communication bottleneck.
  - 🛂 **Security:** The attacker gains: "**System knowledge**".

Adversarial "System-Knowledge" :

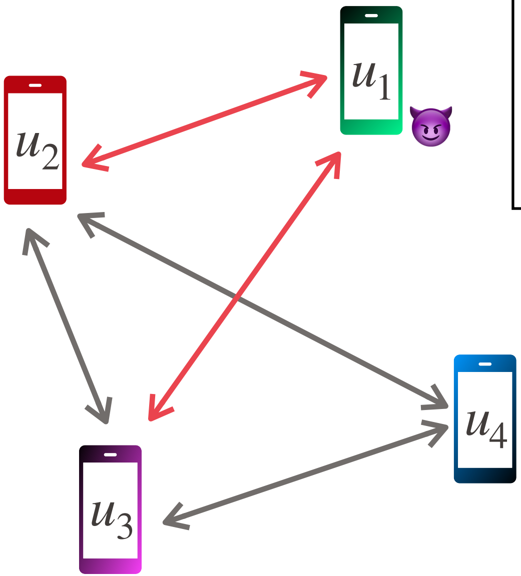


Adversarial Knowledge:

$\Theta_2^{t+1/2}$

- Every new neighbor grants the adversary with a **new and different view** of the state of the underlying system.

Adversarial "System-Knowledge" :



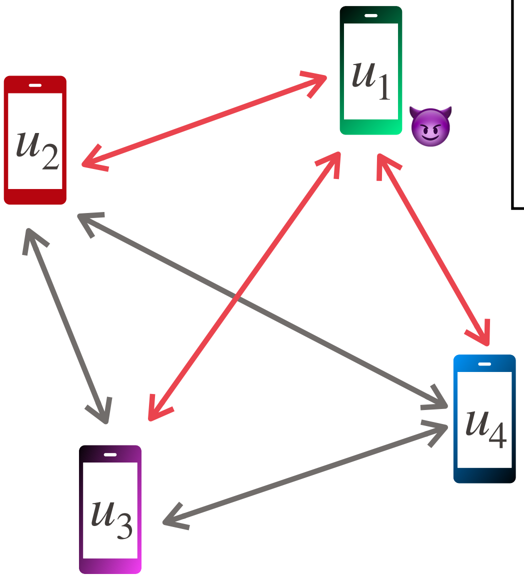
Adversarial Knowledge:

$\Theta_2^{t+1/2}$

$\Theta_3^{t+1/2}$

- Every new neighbor grants the adversary with a **new and different view** of the state of the underlying system.

Adversarial "System-Knowledge" :

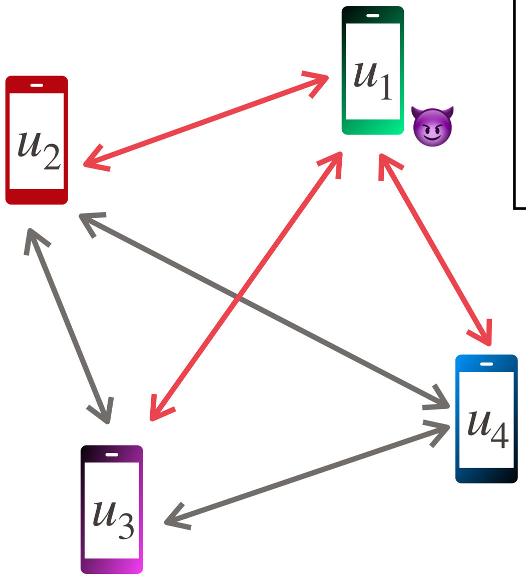


Adversarial Knowledge:

$$\boxed{\begin{array}{c}\Theta_2^{t+1/2}\\ \Theta_3^{t+1/2}\\ \Theta_4^{t+1/2}\end{array}}$$

- Every new neighbor grants the adversary with a **new and different view** of the state of the underlying system.

Adversarial “System-Knowledge” :

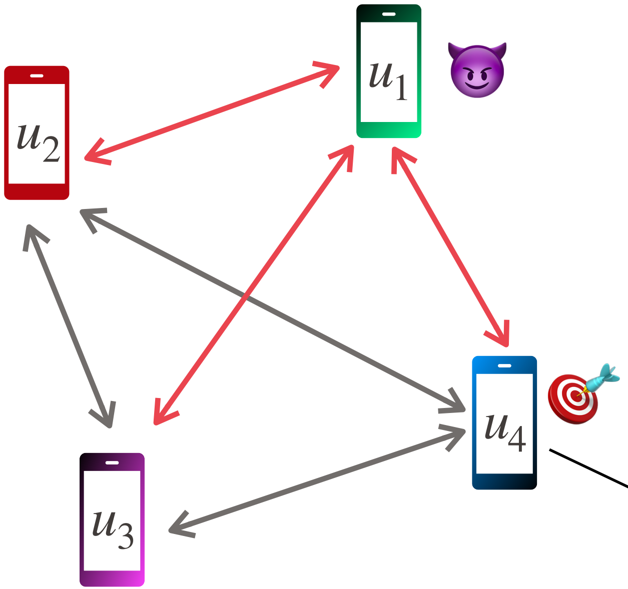


Adversarial Knowledge:

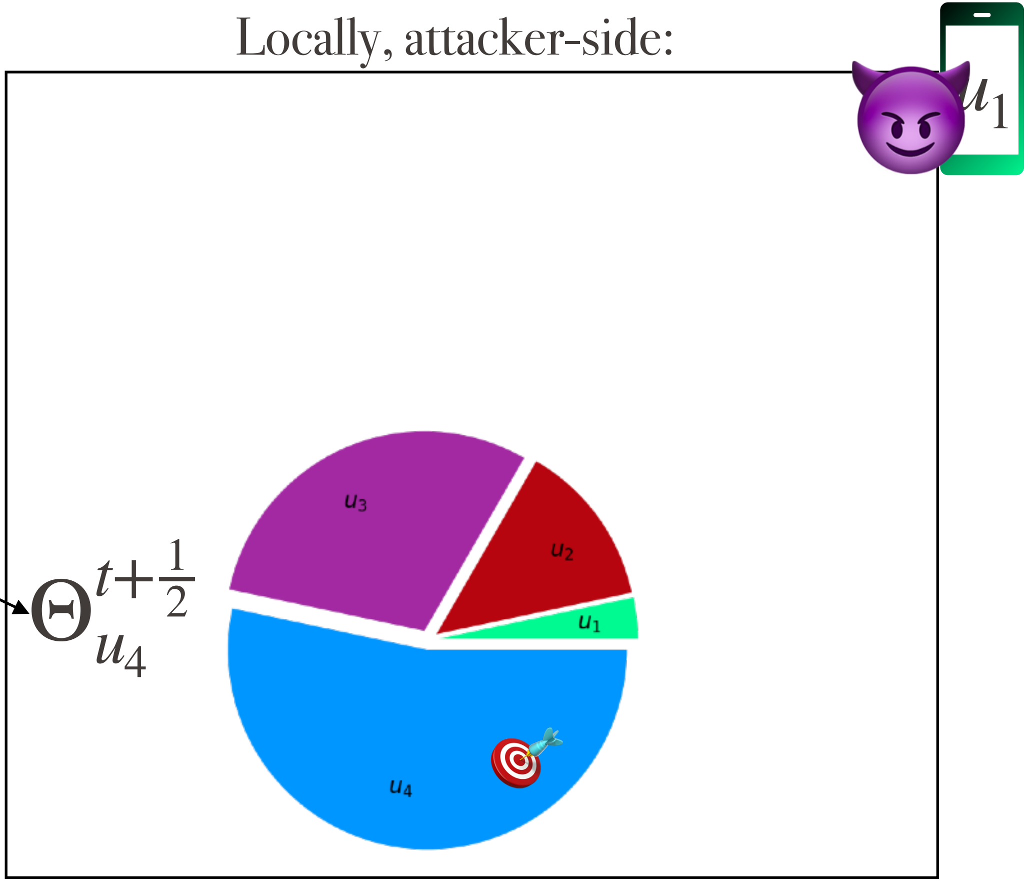
$\Theta_2^{t+1/2}$
$\Theta_3^{t+1/2}$
$\Theta_4^{t+1/2}$

- Every new neighbor grants the adversary with a **new and different view** of the state of the underlying system.
- **This provides novel and unexpected capabilities to adversaries.** Mainly:
  - Disentangle users’ contributions and artificially reduce generalization in the system.

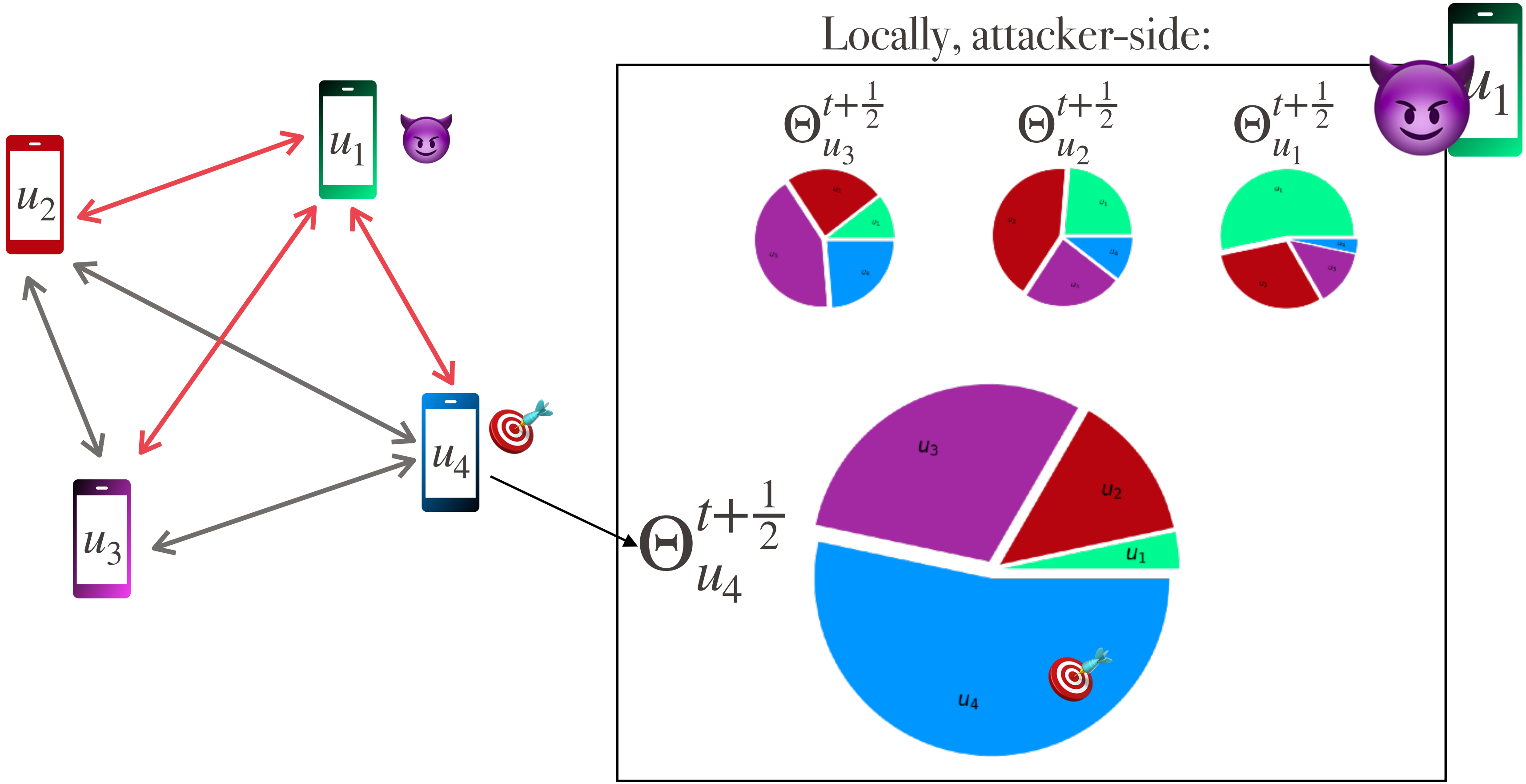
The Functional Marginalization attack:



Locally, attacker-side:



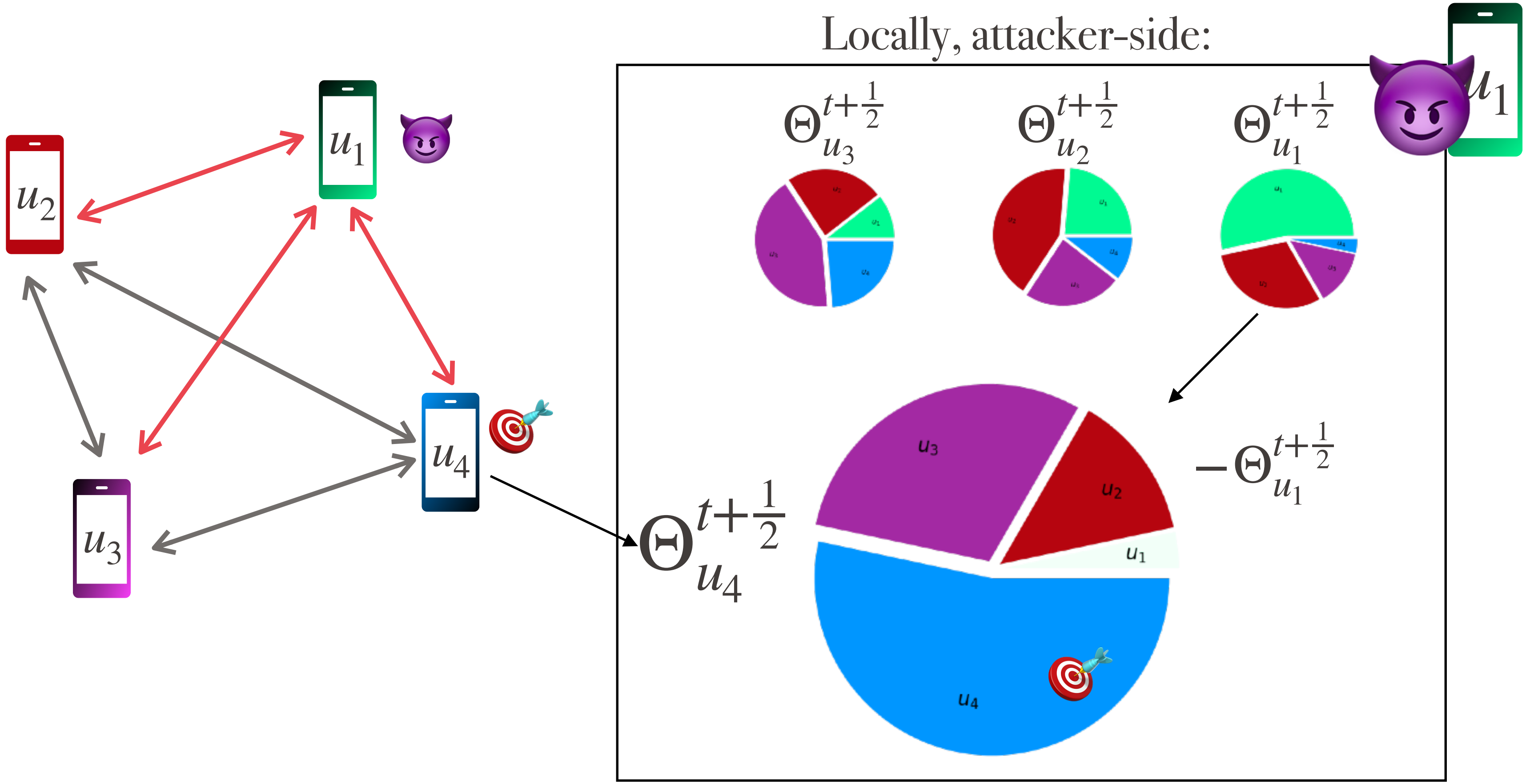
The Functional Marginalization attack:



Locally, attacker-side:

u_1
u_2
u_3
u_4
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_1}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_3
u_2
u_1
u_4

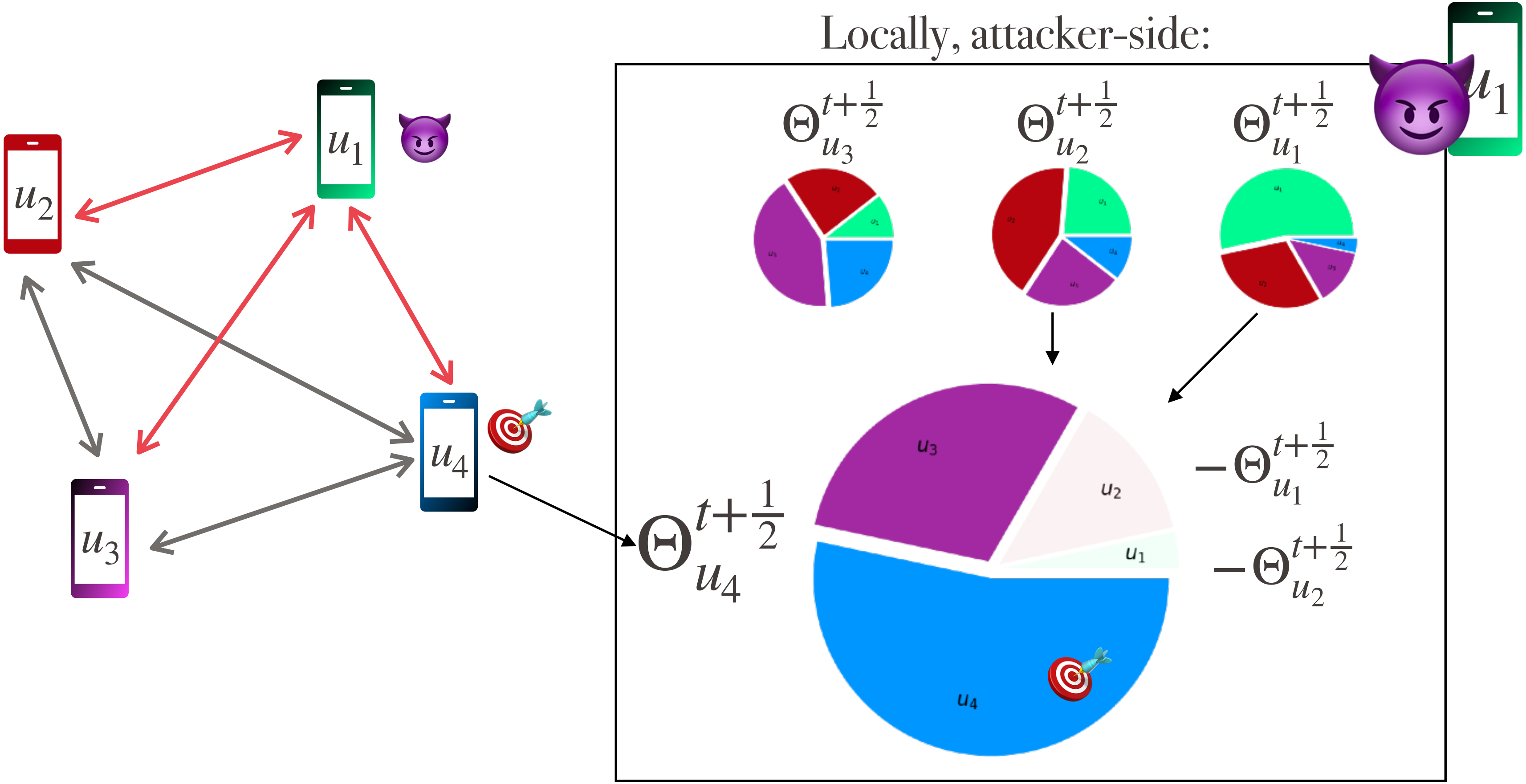
The Functional Marginalization attack:



Locally, attacker-side:

u_1
u_2
u_3
u_4
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_1}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
$-\Theta_{u_1}^{t+\frac{1}{2}}$
u_3
u_2
u_1
u_4

The Functional Marginalization attack:

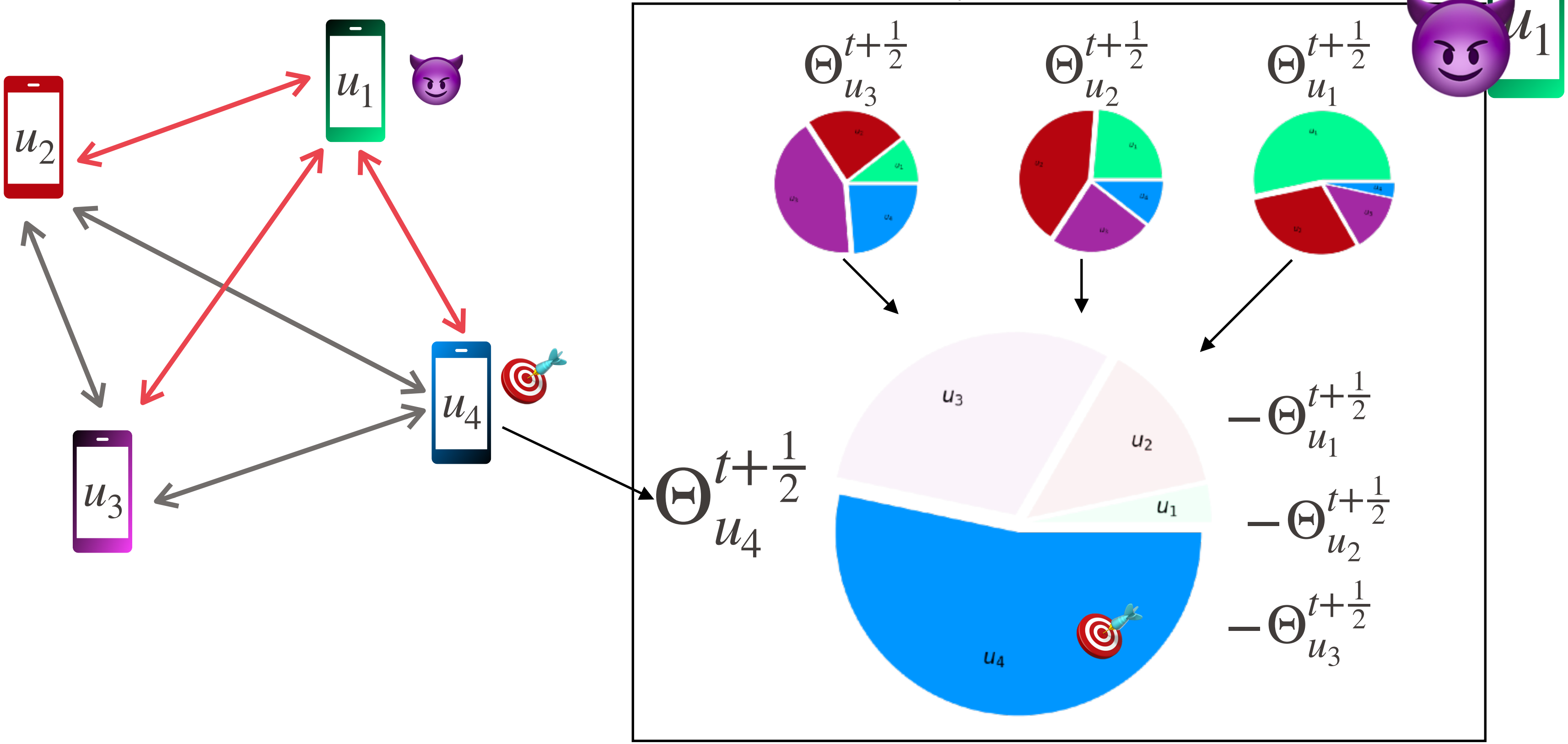


Locally, attacker-side:

u_1
u_2
u_3
u_4
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_1}^{t+\frac{1}{2}}$
u_1
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_3
u_2
u_1
u_4
$-\Theta_{u_1}^{t+\frac{1}{2}}$
$-\Theta_{u_2}^{t+\frac{1}{2}}$

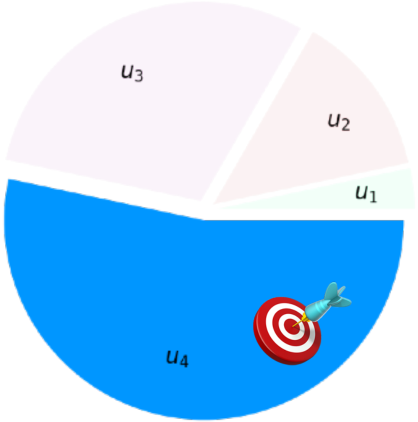
The Functional Marginalization attack:

Locally, attacker-side:

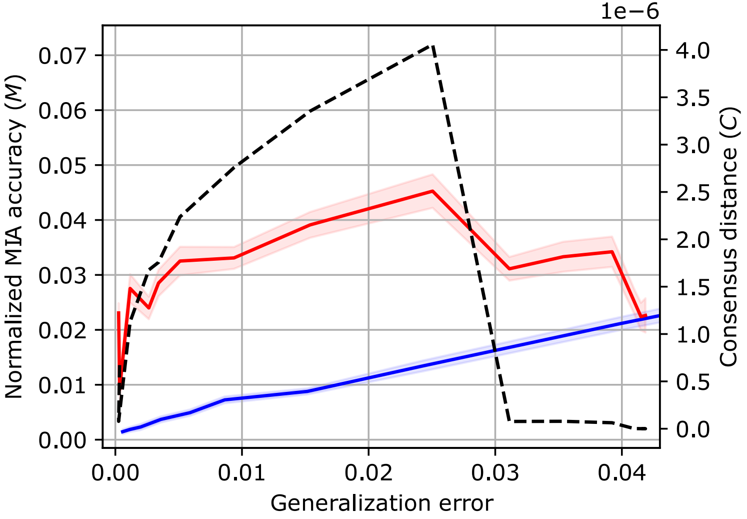

u_1
u_2
u_3
u_4
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_1}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
$-\Theta_{u_1}^{t+\frac{1}{2}}$
$-\Theta_{u_2}^{t+\frac{1}{2}}$
$-\Theta_{u_3}^{t+\frac{1}{2}}$
u_3
u_2
u_1
u_4

MIAs on Functionally Marginalized models :

Functionally Marginalized model:



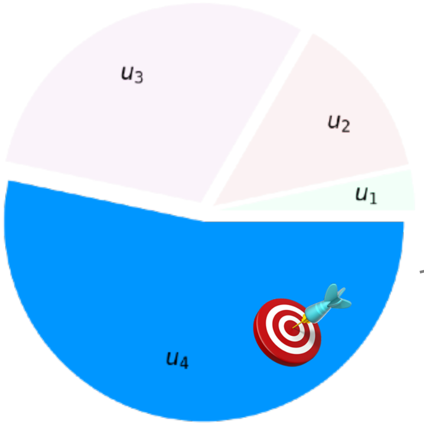
$$\dot{\Theta}^{t}_{u_4} = 4 \cdot (\Theta^{t+\frac{1}{2}}_{u_4} - \frac{\Theta^{t+\frac{1}{2}}_{u_1} + \Theta^{t+\frac{1}{2}}_{u_2} + \Theta^{t+\frac{1}{2}}_{u_3}}{4})$$

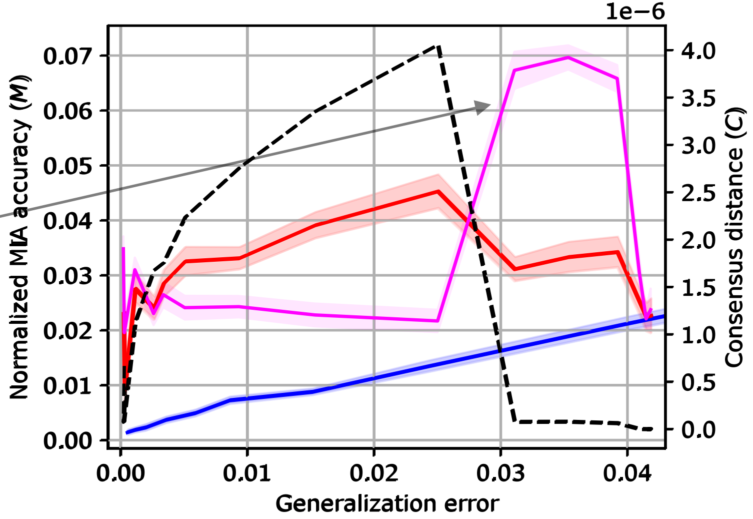


Torus-36, CIFAR-100, ResNet-20

MIAs on Functionally Marginalized models :

Functionally Marginalized model:





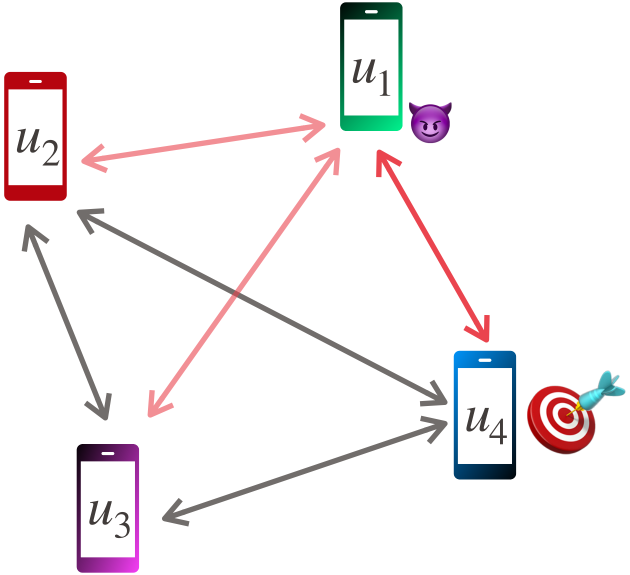
$$\boxed{\dot{\Theta}^{t}_{u_4} = 4 \cdot (\Theta^{t+\frac{1}{2}}_{u_4} - \frac{\Theta^{t+\frac{1}{2}}_{u_1} + \Theta^{t+\frac{1}{2}}_{u_2} + \Theta^{t+\frac{1}{2}}_{u_3}}{4})}$$

Torus-36, CIFAR-100, ResNet-20

The real implications of "System Knowledge":

- When the attacker is **connected to all the target's neighbors**, i.e.,:

$$\boxed{N(u_4) \subseteq N(u_1)}$$



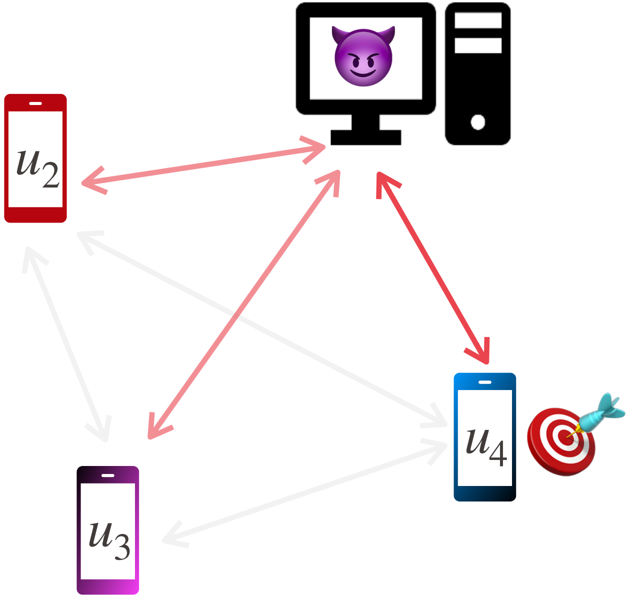
$$\boxed{N(u) = u' \text{s neighbors}}$$

The real implications of “System Knowledge”:

- When the attacker is **connected to all the target’s neighbors,** i.e.,:

$$\boxed{N(u_4) \subseteq N(u_1)}$$

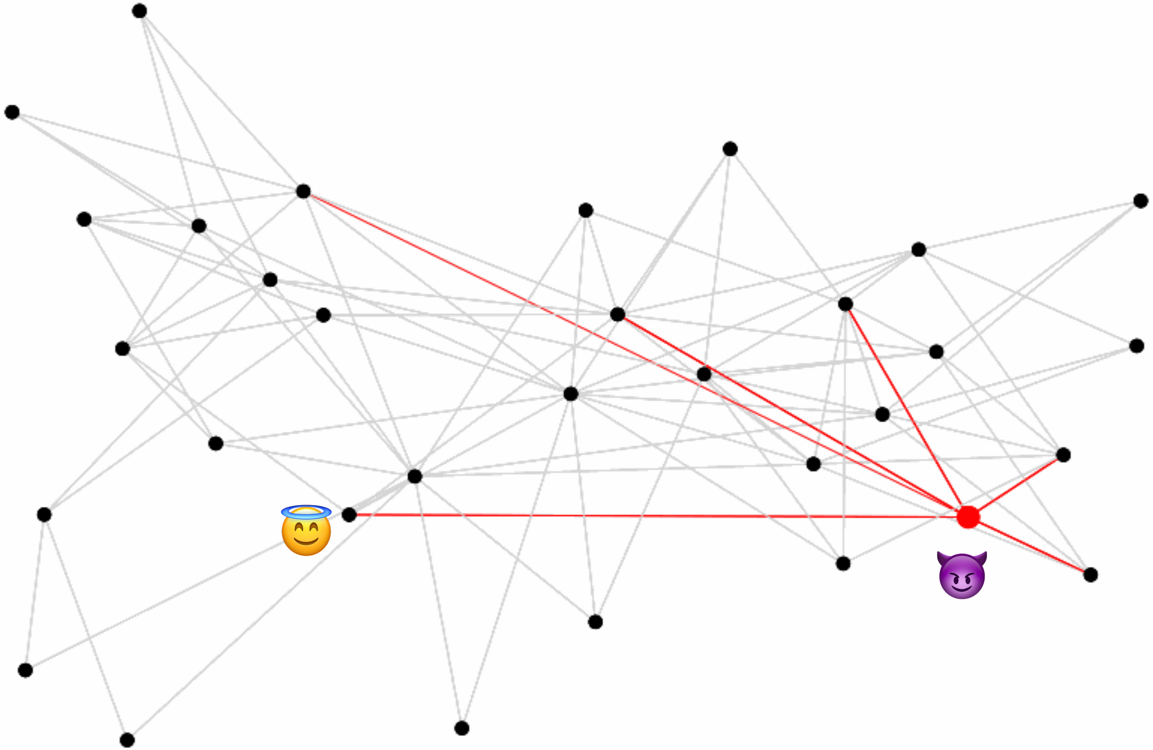
- **It achieves the same adversarial capabilities of a parameter server in FL:**
  - 👀 Access individual gradients produced by the targets(s) **[semi-honest].**
  - ✍️ Decide the local parameters of the targets(s) **[malicious].**



$$\boxed{N(u) = u'\text{s neighbors}}$$

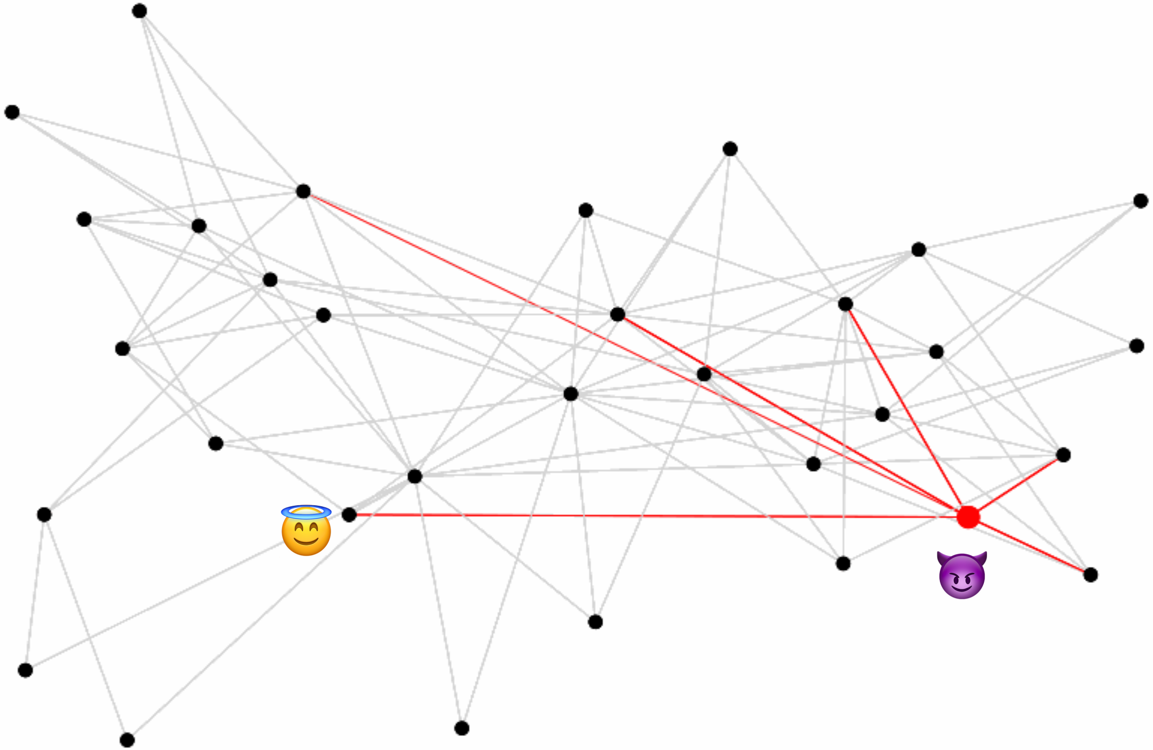
A decentralized user A becomes equivalent to a parameter server in FL for every node V s.t.:

$$N(V) \subseteq N(A)$$



($N(\cdot)$: set of neighbors)

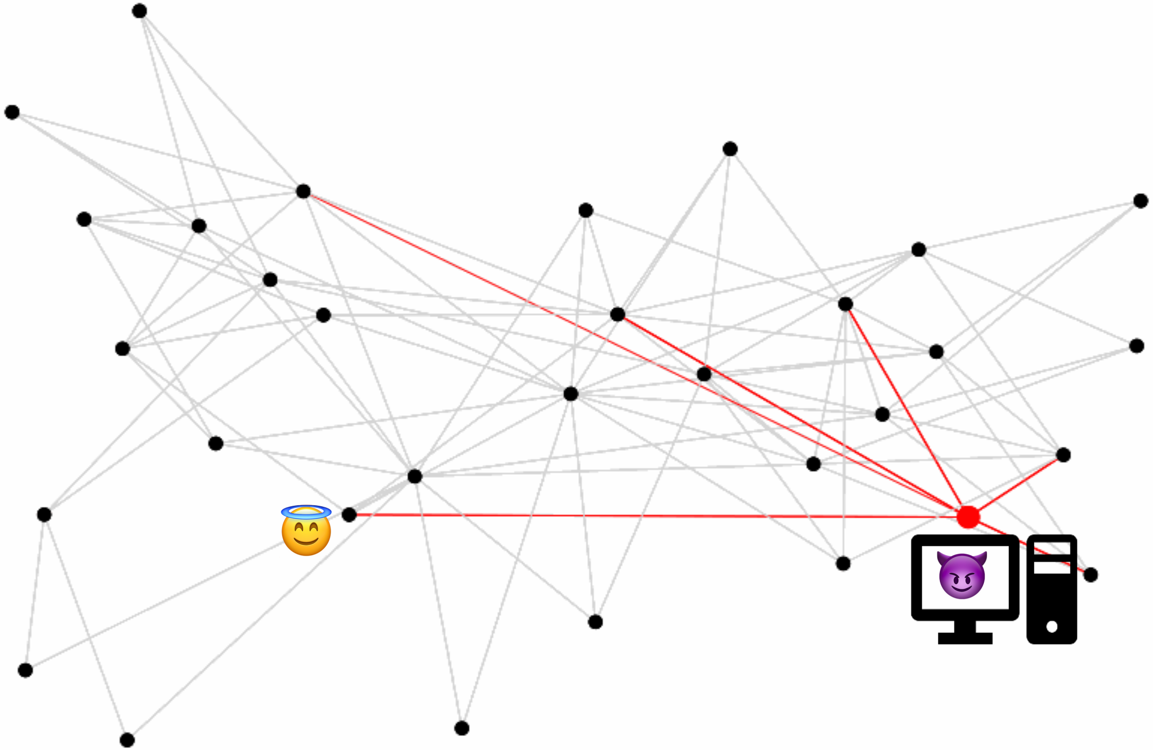
A decentralized user A becomes equivalent to a parameter server in FL for every node V s.t.:
$$N(V) \subseteq N(A)$$



($N(\cdot)$: set of neighbors)

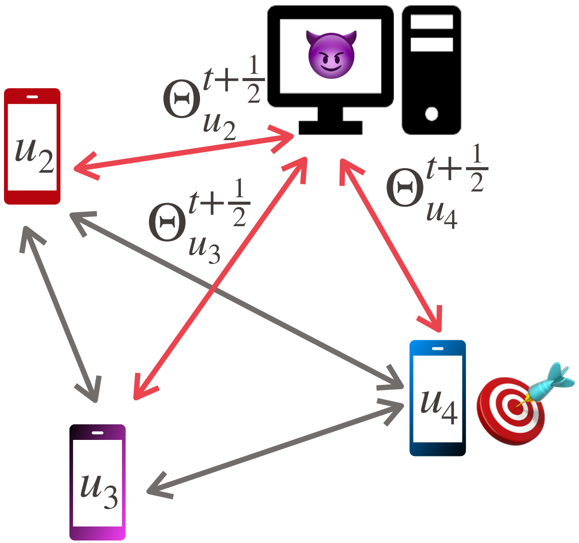
A decentralized user A becomes equivalent to a parameter server in FL for every node V s.t.:

$$N(V) \subseteq N(A)$$



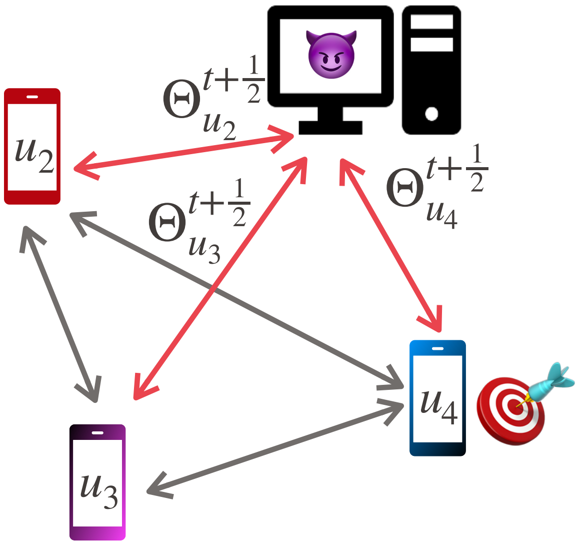
$(N(\cdot)$: set of neighbors$)$

Recover target(s)' gradient:


$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_2
u_3
u_4

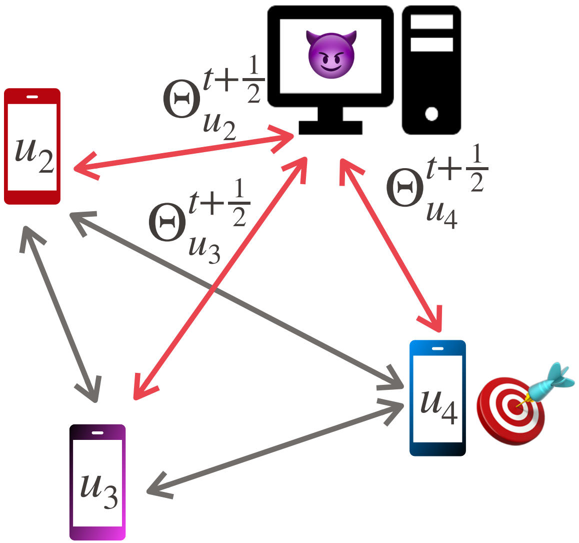
- **Gradient recovery** 👀:
  - The attacker can retrieve the gradient of the target(s) by observing two consecutive rounds:

Recover target(s)' gradient:


$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_2
u_3
u_4

- **Gradient recovery** 👀:
  - The attacker can retrieve the gradient of the target(s) by observing two consecutive rounds:

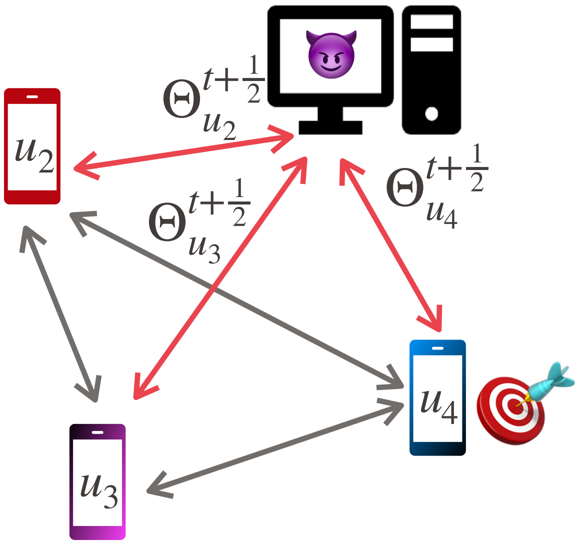
Recover target(s)' gradient:


$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_2
u_3
u_4

- **Gradient recovery 👀:**
  - The attacker can retrieve the gradient of the target(s) by observing two consecutive rounds:

$$\boxed{\frac{1}{\eta}(\Theta_4^{t+\frac{1}{2}} - \frac{\Theta_1^{t-\frac{1}{2}} + \Theta_2^{t-\frac{1}{2}} + \Theta_3^{t-\frac{1}{2}} + \Theta_4^{t-\frac{1}{2}}}{4}) = \nabla_{\Theta_i^t}(x^t)}$$

Recover target(s)' gradient:

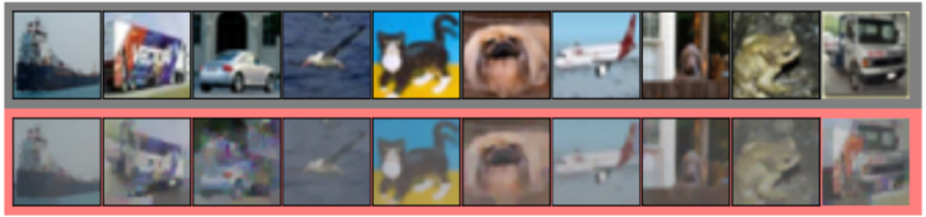


$\Theta_{u_2}^{t+\frac{1}{2}}$
$\Theta_{u_3}^{t+\frac{1}{2}}$
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_2
u_3
u_4

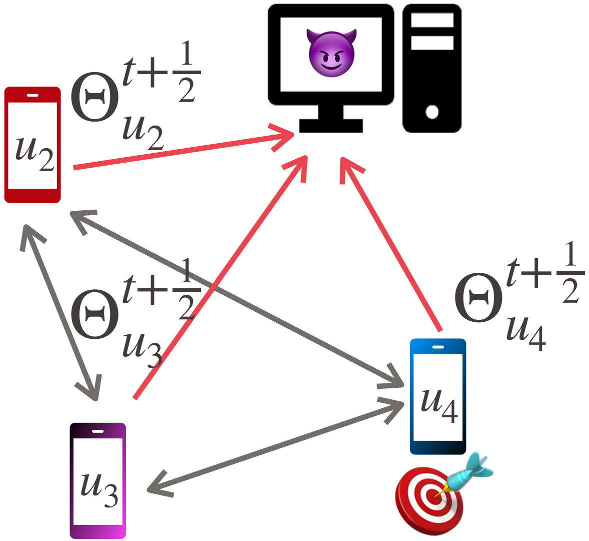
- **Gradient recovery** 👀:
  - The attacker can retrieve the gradient of the target(s) by observing two consecutive rounds:

$$\boxed{\frac{1}{\eta}\left(\Theta_4^{t+\frac{1}{2}} - \frac{\Theta_1^{t-\frac{1}{2}} + \Theta_2^{t-\frac{1}{2}} + \Theta_3^{t-\frac{1}{2}} + \Theta_4^{t-\frac{1}{2}}}{4}\right) = \nabla_{\Theta_i^t}(x^t)}$$

Opt. based gradient inversion (passive) [CIFAR10]:

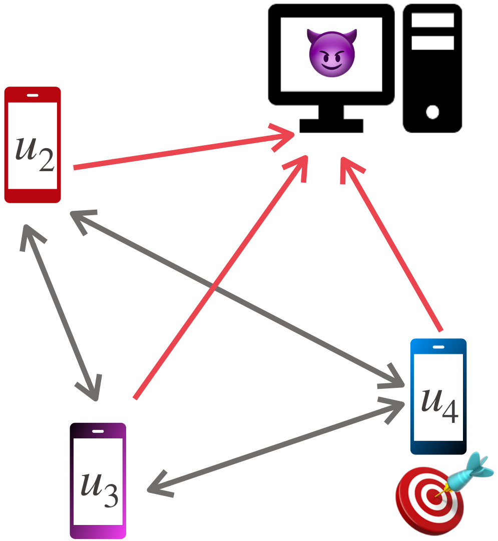


Deciding target(s)' local parameters:


$\Theta_{u_2}^{t+\frac{1}{2}}$
u_2
$\Theta_{u_3}^{t+\frac{1}{2}}$
u_3
$\Theta_{u_4}^{t+\frac{1}{2}}$
u_4

- **State override attack** ✍️:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

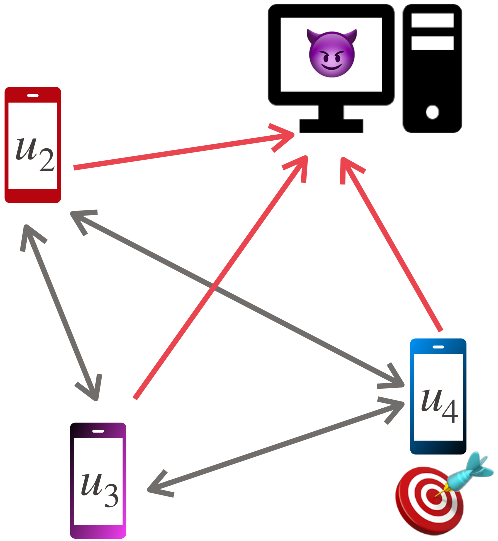
Deciding target(s)' local parameters:



- **State override attack** ✍:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

$$\boxed{\tilde{\Theta} = -\left(\Theta_{u_2}^{t+\frac{1}{2}} + \Theta_{u_3}^{t+\frac{1}{2}} + \Theta_{u_4}^{t+\frac{1}{2}}\right) + \text{🧨}}$$

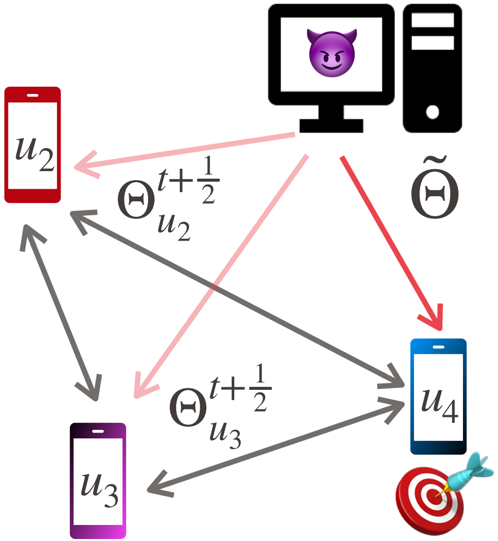
Deciding target(s)' local parameters:



- **State override attack** ✍️:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

$$\boxed{\tilde{\Theta} = -(\Theta_{u_2}^{t+\frac{1}{2}} + \Theta_{u_3}^{t+\frac{1}{2}} + \Theta_{u_4}^{t+\frac{1}{2}}) + \text{🧨}}$$

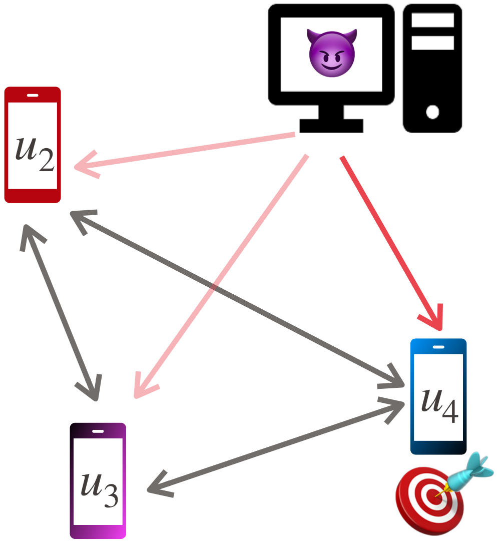
Deciding target(s)' local parameters:


u_2
$\Theta_{u_2}^{t+\frac{1}{2}}$
$\tilde{\Theta}$
$\Theta_{u_3}^{t+\frac{1}{2}}$
u_3
u_4

- **State override attack** ✍:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

$$\tilde{\Theta} = -\left(\Theta_{u_2}^{t+\frac{1}{2}} + \Theta_{u_3}^{t+\frac{1}{2}} + \Theta_{u_4}^{t+\frac{1}{2}}\right) + \text{🧨}$$

Deciding target(s)' local parameters:

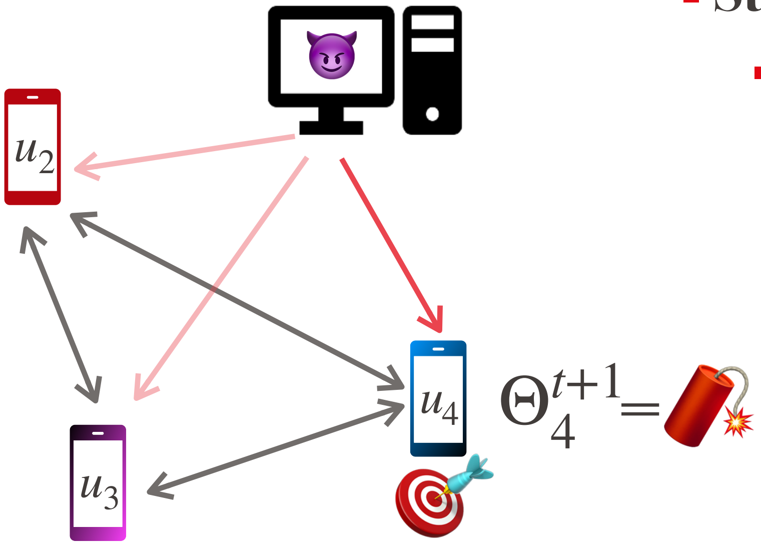


$\Theta_4^{t+1} =$ 🧨

- **State override attack** ✍️:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

$$\boxed{\tilde{\Theta} = -\left(\Theta_{u_2}^{t+\frac{1}{2}} + \Theta_{u_3}^{t+\frac{1}{2}} + \Theta_{u_4}^{t+\frac{1}{2}}\right) + \text{🧨}}$$

Deciding target(s)' local parameters:



- **State override attack** ✍️:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

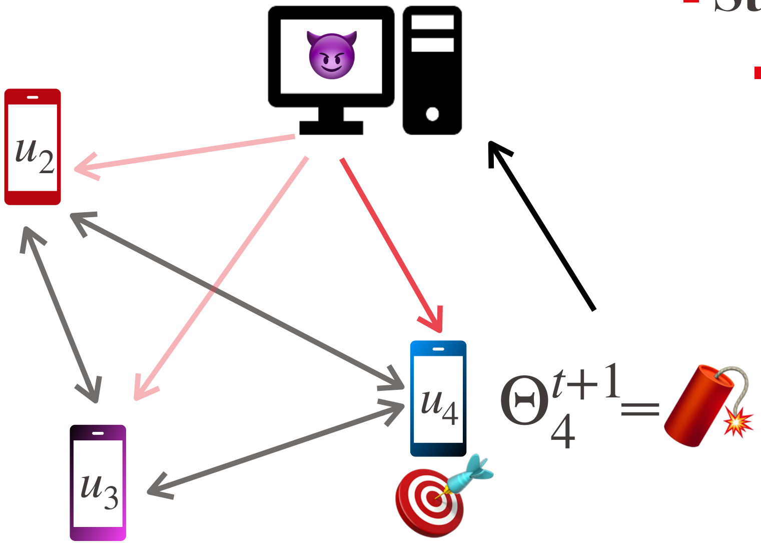
$$\boxed{\tilde{\Theta} = -\left(\Theta_{u_2}^{t+\frac{1}{2}} + \Theta_{u_3}^{t+\frac{1}{2}} + \Theta_{u_4}^{t+\frac{1}{2}}\right) + \text{🧨}}$$

  - Payload (🧨): trap weights [1, 2]

[1] Wen et al. *"Fishing for user data in large-batch federated learning via gradient magnification"* PMLR'22

[2] Boenisch et al *"When the curious abandon honesty: Federated learning is not private"* *EuroS&P'23*

Deciding target(s)' local parameters:

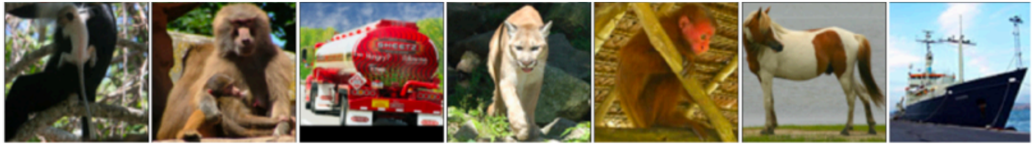


- **State override attack** ✍️:
  - The attacker sets the target(s)' parameters to a chosen **payload** 🧨 by.
    1. Waiting for neighbors model updates.
    2. Creating an adversarial model update:

$$\boxed{\tilde{\Theta} = -\left(\Theta_{u_2}^{t+\frac{1}{2}} + \Theta_{u_3}^{t+\frac{1}{2}} + \Theta_{u_4}^{t+\frac{1}{2}}\right) + \text{🧨}}$$

  - Payload (🧨): trap weights [1, 2]

Verbatim copies of some of the batch data [STL10]:



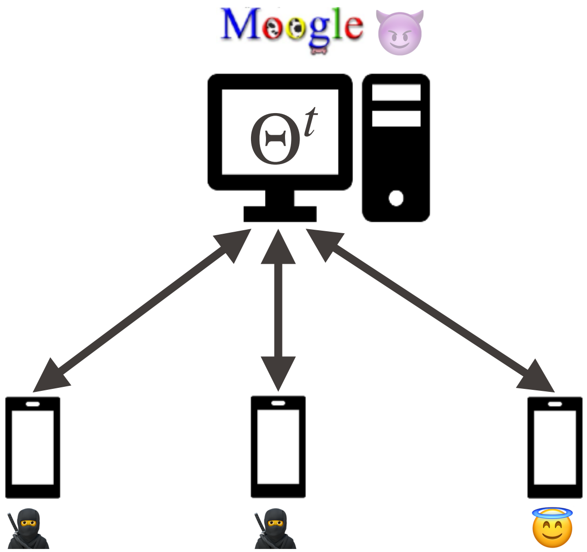
[1] Wen et al. "*Fishing for user data in large-batch federated learning via gradient magnification*" PMLR'22

[2] Boenisch et al "*When the curious abandon honesty: Federated learning is not private*" *EuroS&P'23*

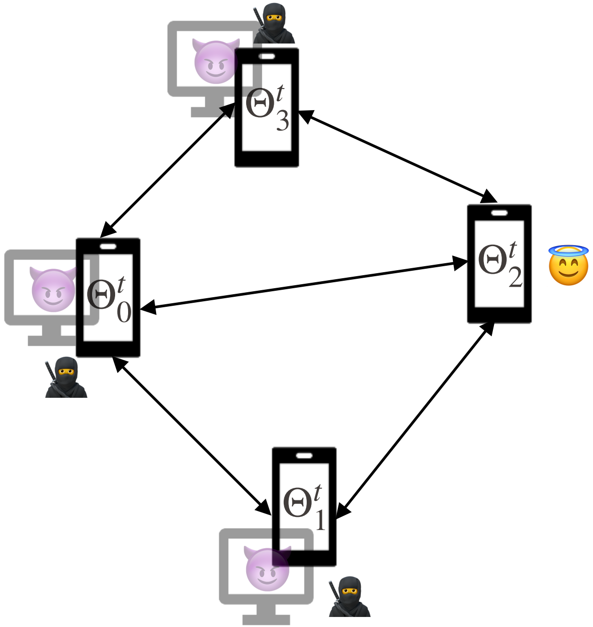
Decentralization; the aftermath:

- A **decentralized user** can be as powerful as a **federated server**, but:

(a) In FL, there is a single and well-defined server:



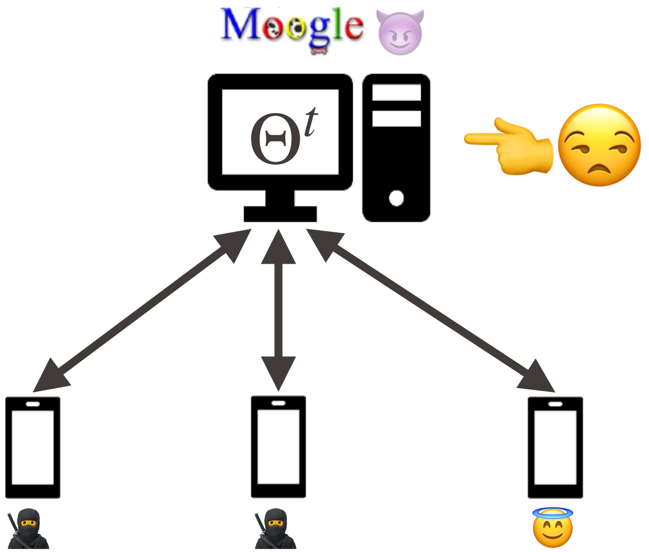
(b) In DL, we have multiple, equivalently powerful, (anonymous) users:



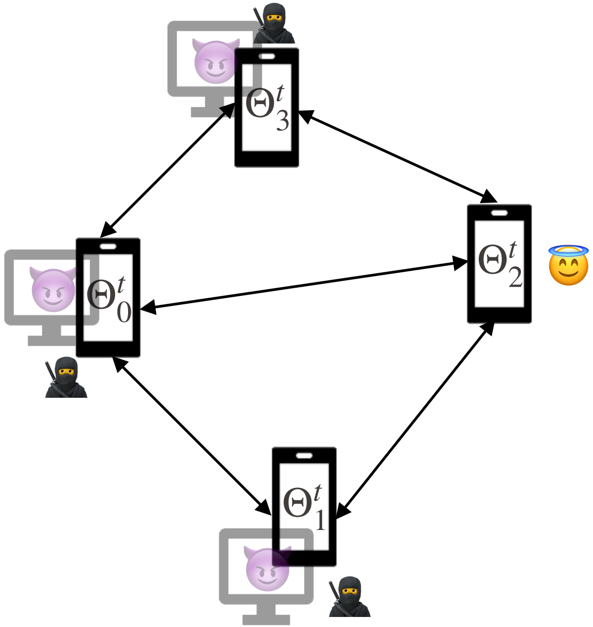
Decentralization; the aftermath:

- A **decentralized user** can be as powerful as a **federated server**, but:

(a) In FL, there is a single and well-defined server:



(b) In DL, we have multiple, equivalently powerful, (anonymous) users:



Summing Up:

- Privacy offered by DL is a function of the underlying topology. However:

Summing Up:

- Privacy offered by DL is a function of the underlying topology. However:
- Any configuration seems to provide only less privacy than FL.

Summing Up:

- Privacy offered by DL is a function of the underlying topology. However:
- Any configuration seems to provide only less privacy than FL.
  - Every **sparse topology** induces local generalization.

Summing Up:

- Privacy offered by DL is a function of the underlying topology. However:
- Any configuration seems to provide only less privacy than FL.
  - Every **sparse topology** induces local generalization.
  - **Dense topologies** allow the adversary to collect system knowledge and become as powerful as a parameter server in FL (what DL aimed to prevent).

Summing Up:

- Privacy offered by DL is a function of the underlying topology. However:
- Any configuration seems to provide only less privacy than FL.
  - Every **sparse topology** induces local generalization.
  - **Dense topologies** allow the adversary to collect system knowledge and become as powerful as a parameter server in FL (what DL aimed to prevent).
    - Multiple **super-nodes** can now exist simultaneously.

Open problems:

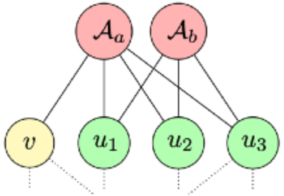
Open problems:

- **Main problem with DL is that attackers can choose their neighbors.** Could we enforce "secure topologies" without a super-node?

Open problems:

- **Main problem with DL is that attackers can choose their neighbors.** Could we enforce "secure topologies" without a super-node?

- **DL-aware Secure Aggregation protocols are needed** (in the paper, we show that standard ones can be evaded).



Is there still time?

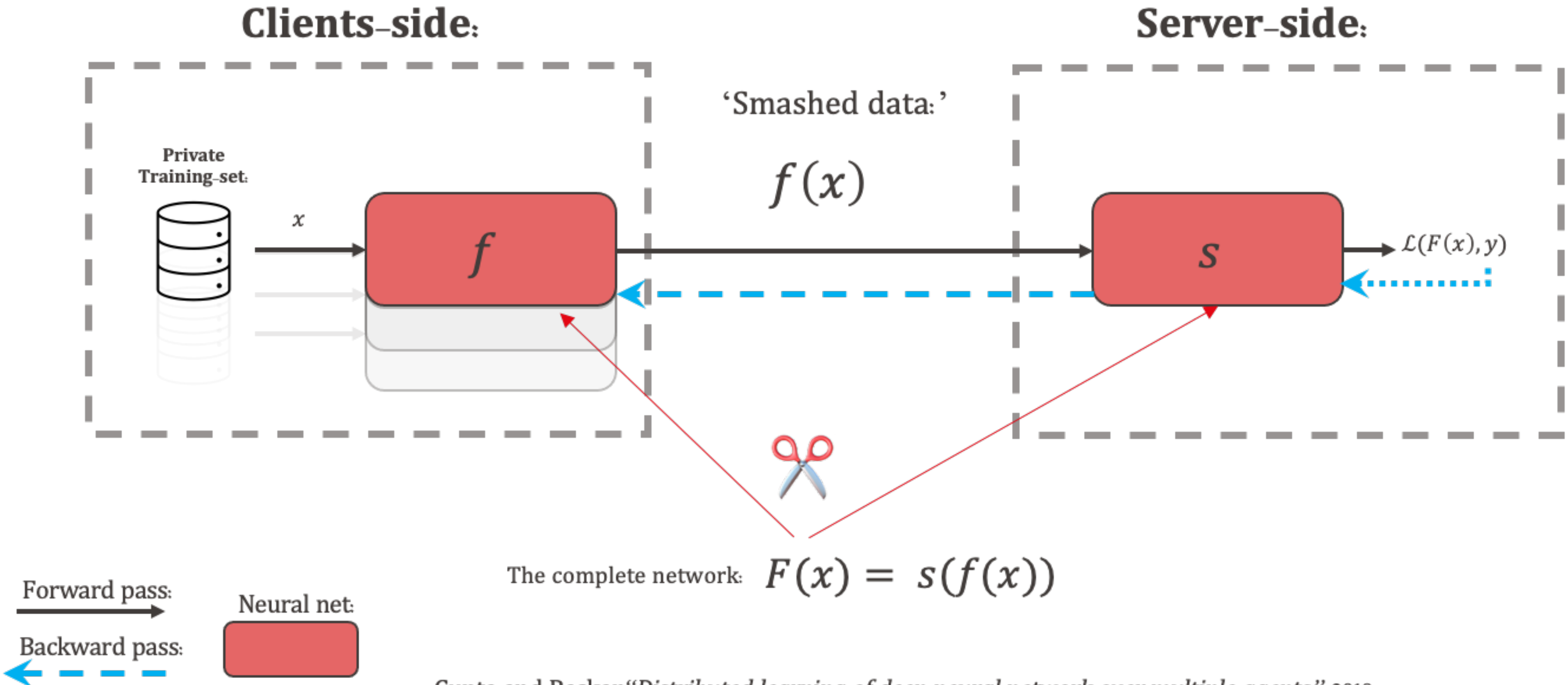
- [Yes] talk about Split Learning;
- [No] go to conclusions;

Different ingredients, same result (a bit worse).

Is **Split Learning** Private? No!

Pasquini, Ateniese, and Bernaschi "*Unleashing the Tiger: Inference Attacks on Split Learning*" ACM CCS'21

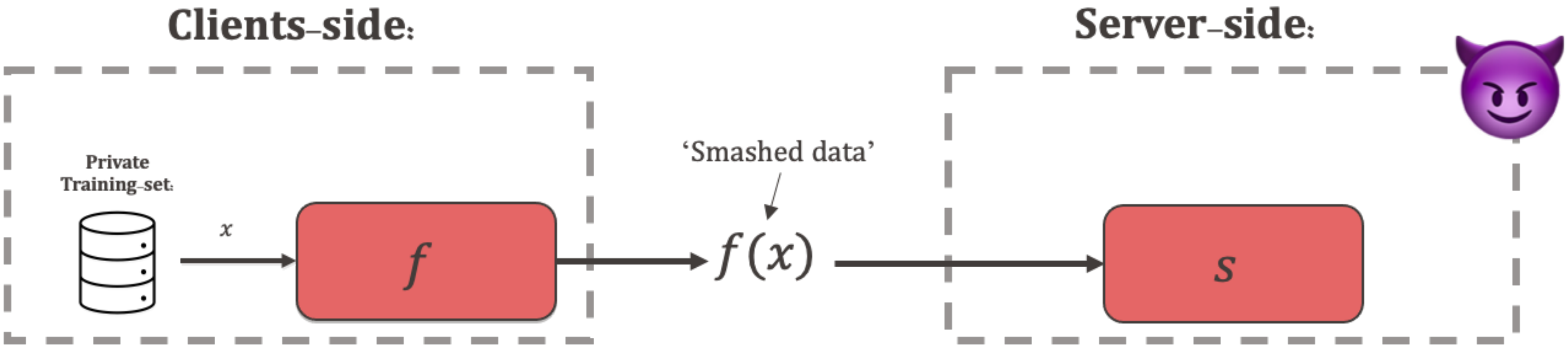
Split Learning



The complete network: $F(x) = s(f(x))$

Gupta and Raskar *"Distributed learning of deep neural network over multiple agents"* 2018

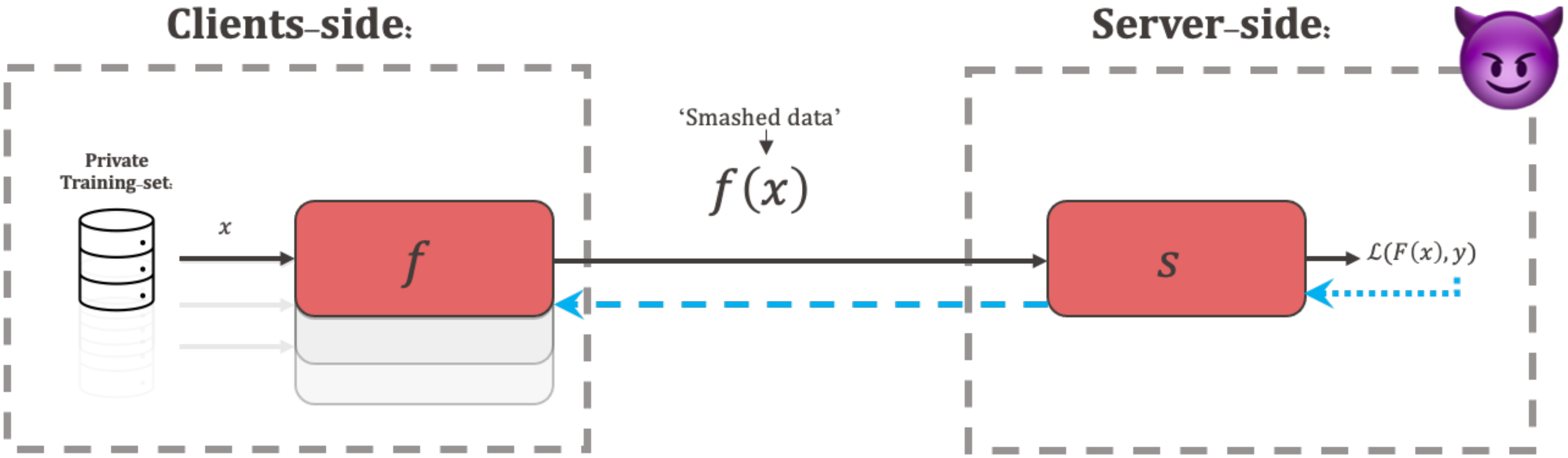
Split Learning is private 'cause:

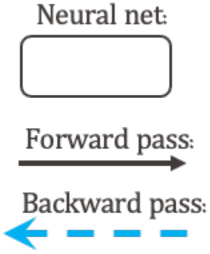


The privacy-preserving property of Split Learning hinges on:

1. The server does not observe the raw data; only the smashed data $f(x)$.
2. Smashed data, per se, do not leak information about the raw data.
3. The server cannot invert the smashed data because it does not know the function f.

The problem with Split Learning:

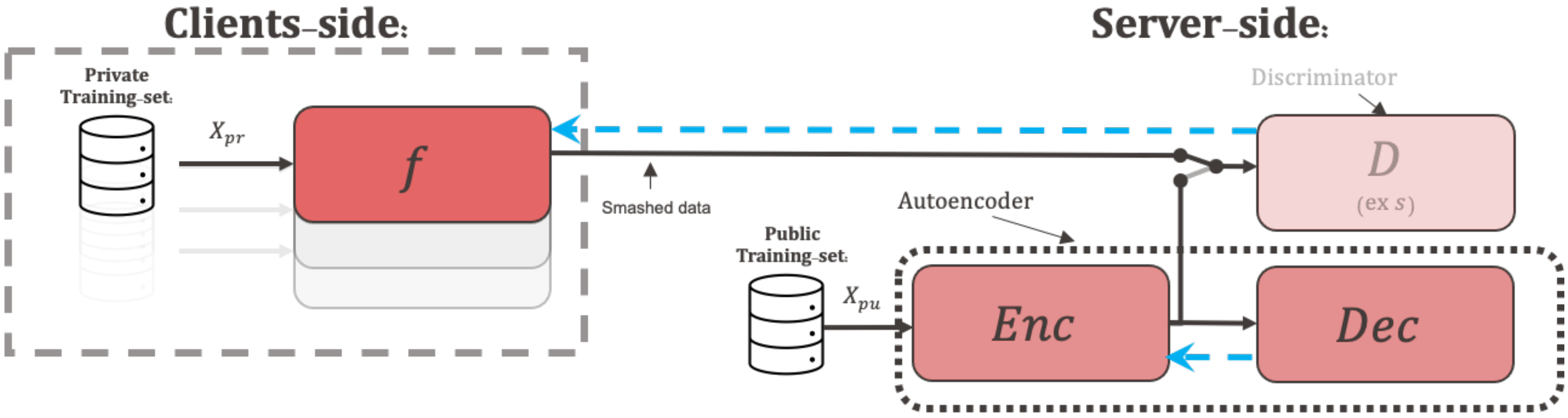


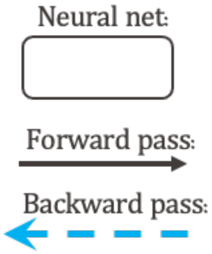


❑ **The server controls client's learning process.**

- The server can just "*train*" f to leak information about x

The feature-space hijacking attack:



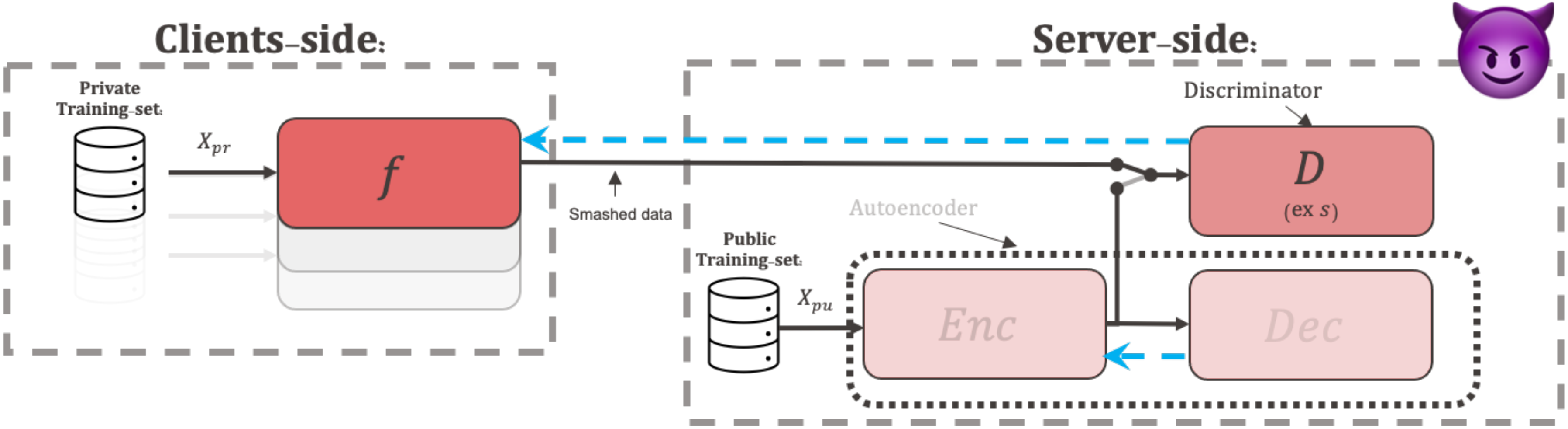


Autoencoding loss: $L_{Enc,Dec} = MSE(X_{pu}, Dec(Enc(X_{pu})))$

Adversarial losses: $L_f = \log(1 - D(f(X_{pr})))$ $L_D = \log\left(1 - D\left(Enc(X_{pr})\right)\right) + \log(D(f(X_{pr})))$

Let's force ***f*** to map its input in the feature-space defined by ***Enc***

The feature-space hijacking attack:



Neural net: □

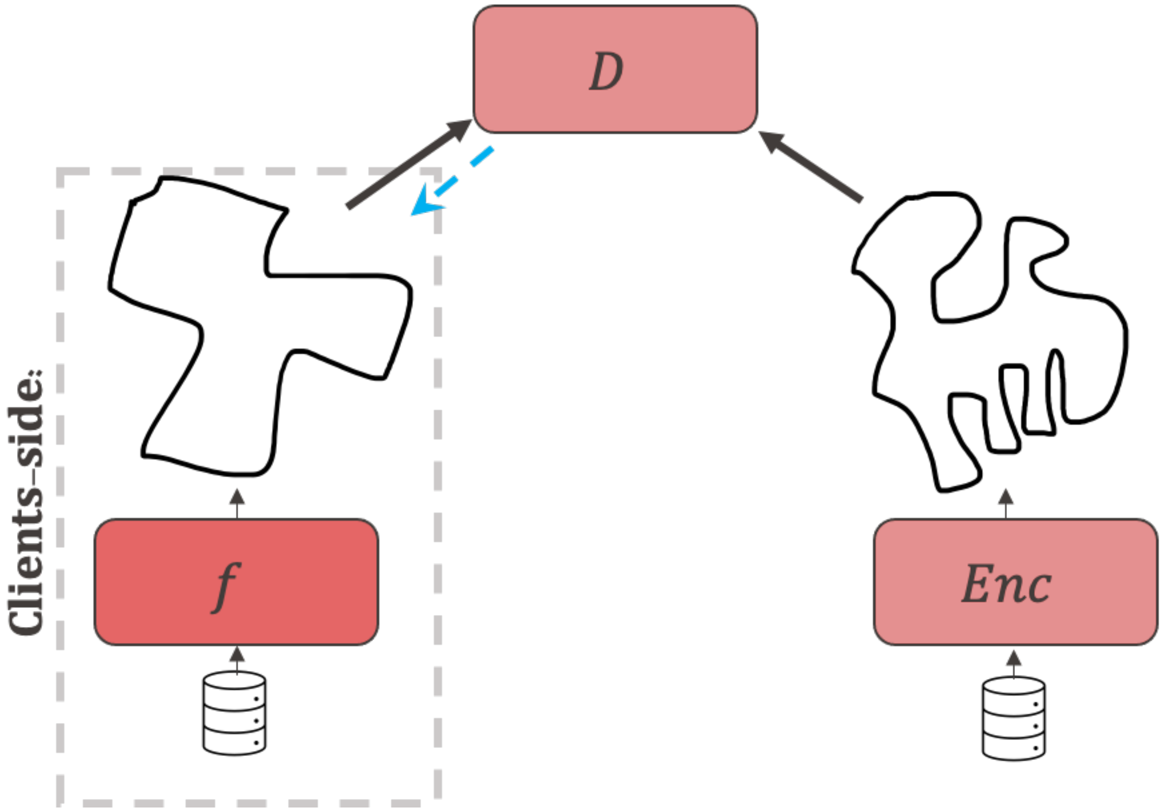
Forward pass: →

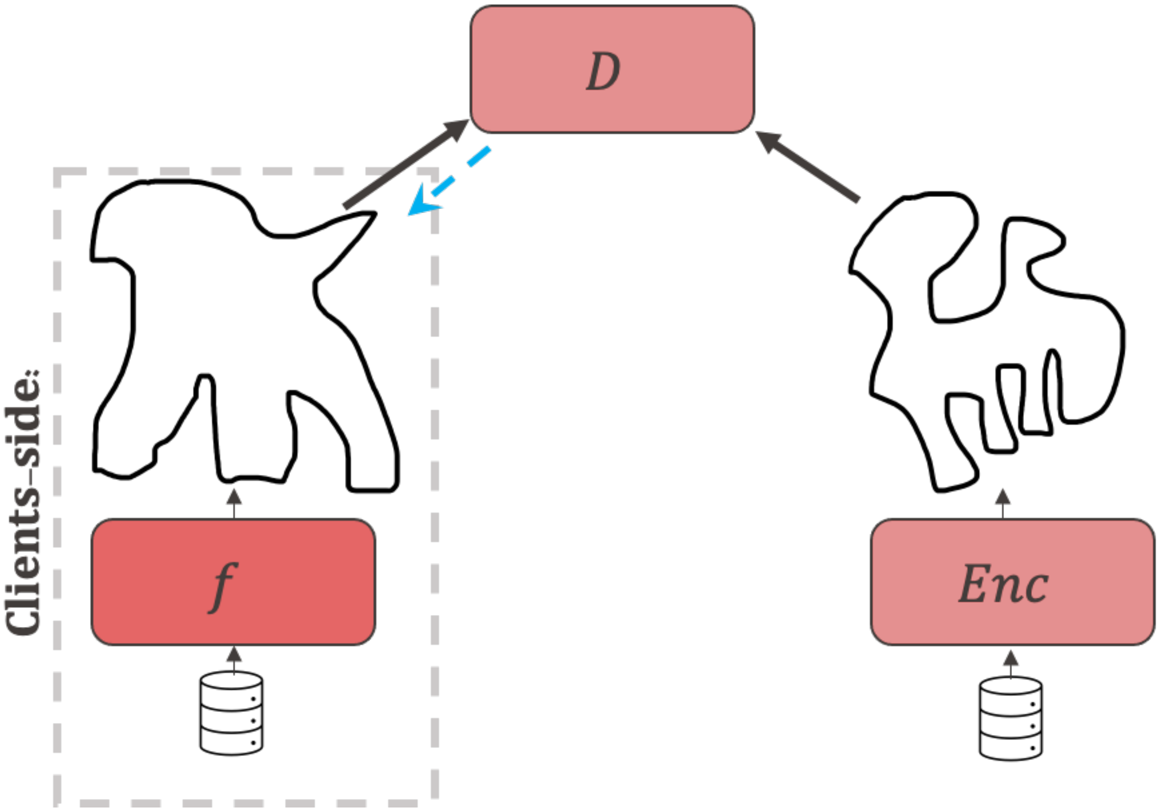
Backward pass: ← - - -

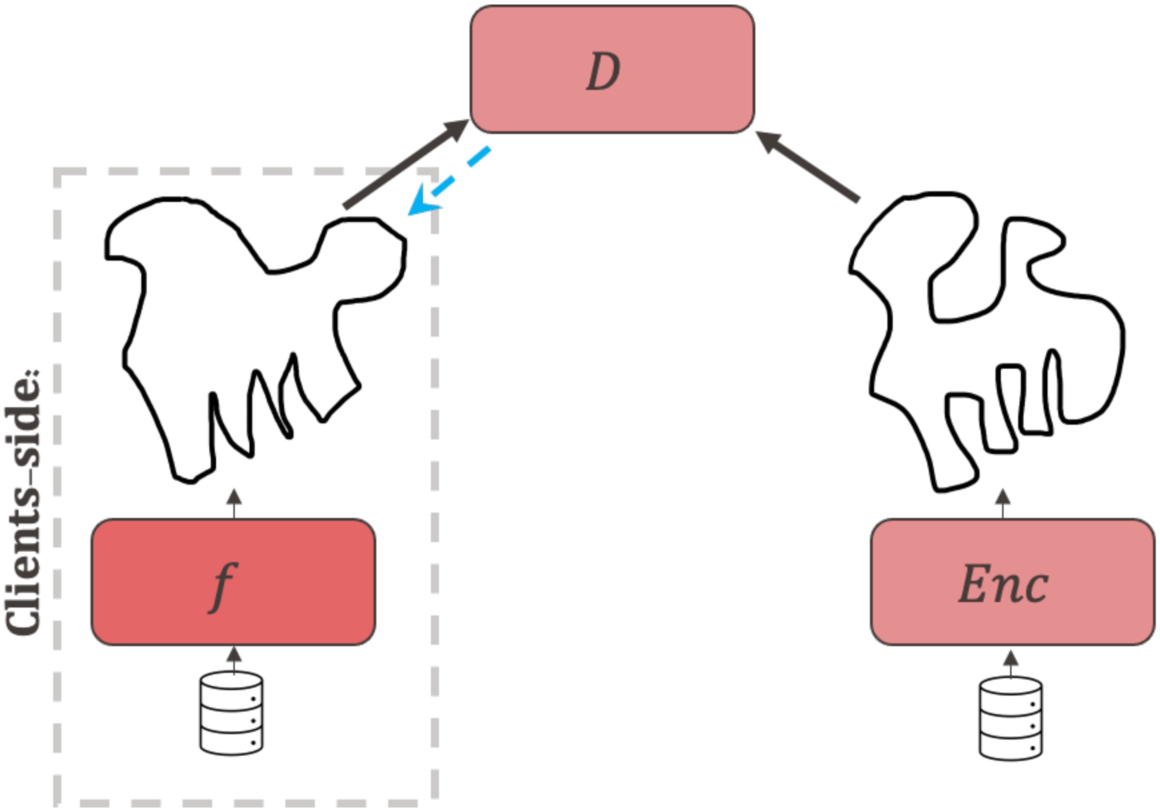
Autoencoding loss: $L_{Enc,Dec} = MSE(X_{pu}, Dec(Enc(X_{pu})))$

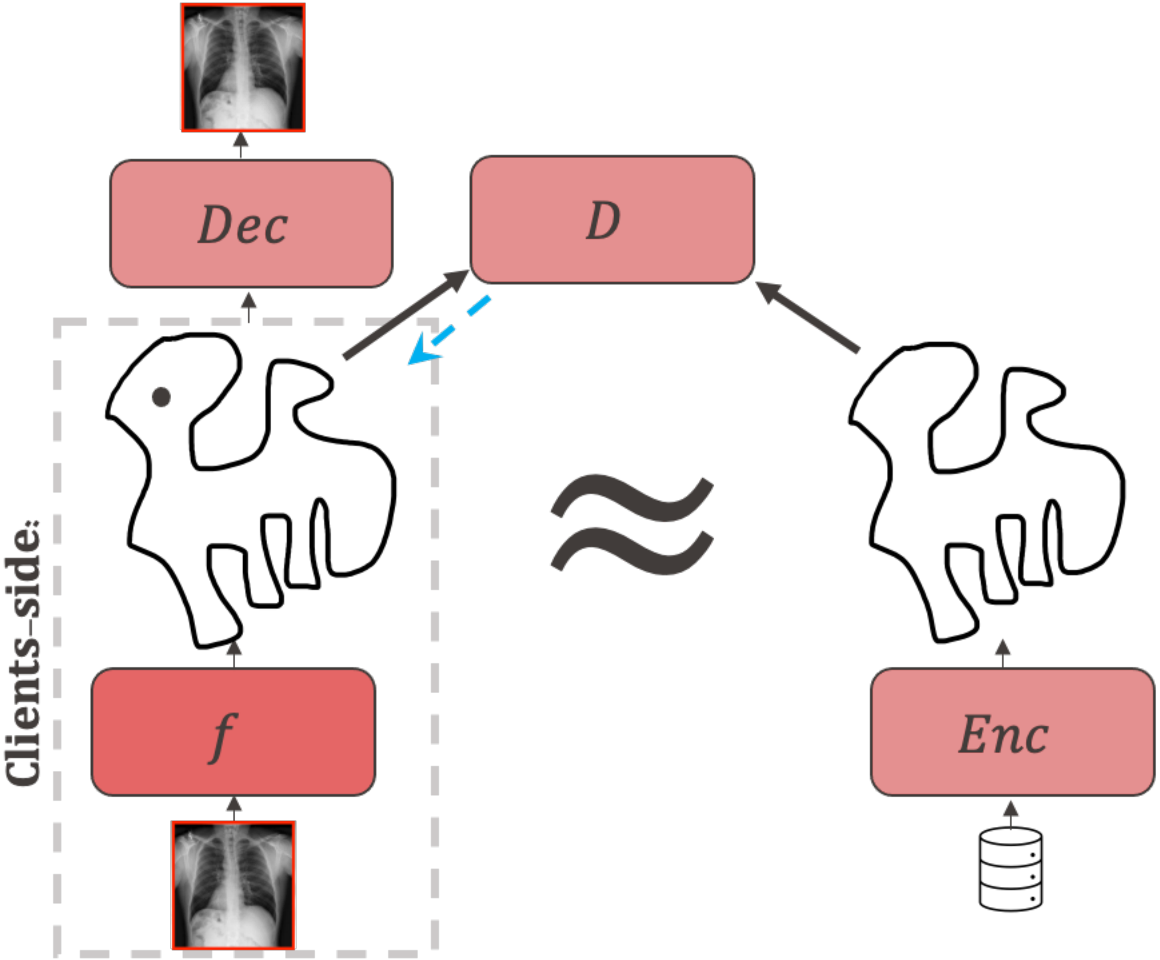
Adversarial losses: $\mathbf{L_f = \log(1 - D(f(X_{pr})))}$ $L_D = \log\left(1 - D\left(Enc(X_{pr})\right)\right) + \log(D(f(X_{pr})))$

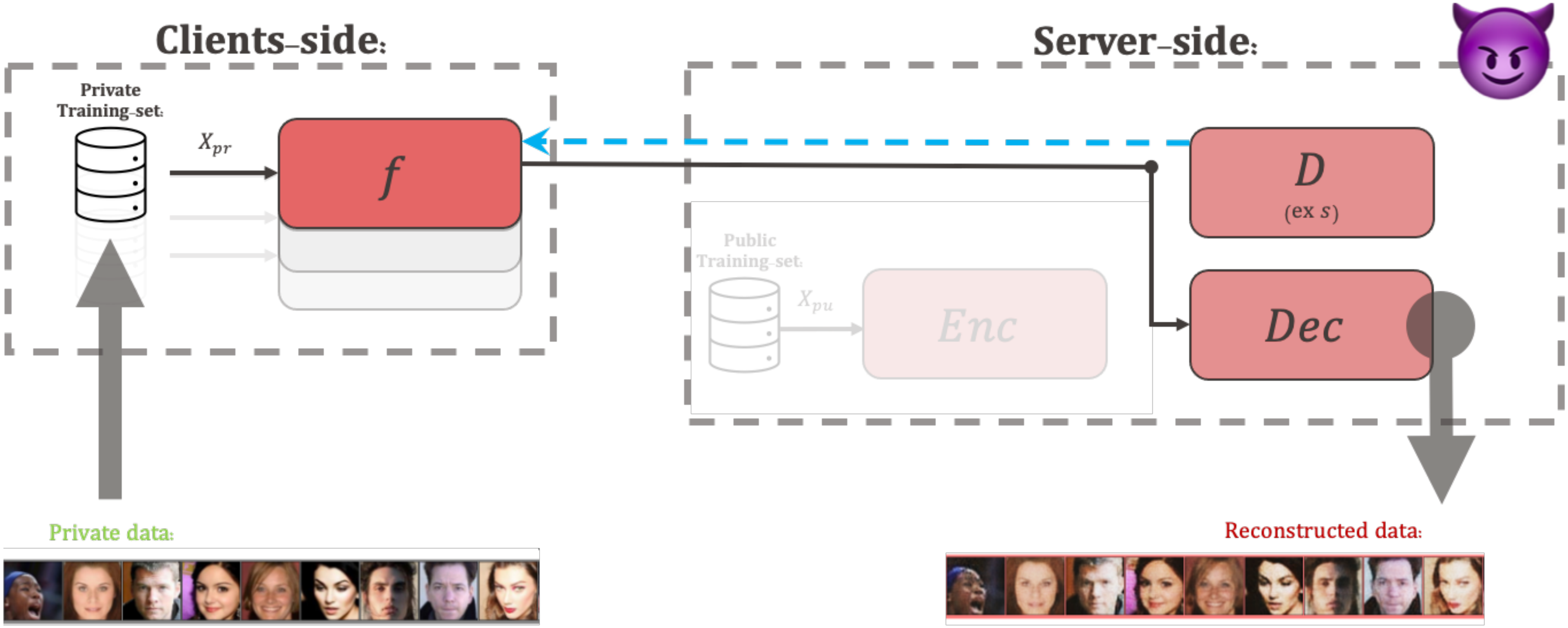
Let's force ***f*** to map its input in the feature-space defined by ***Enc***


D
Clients-side:
f
Enc


D
Clients–side:
f
Enc


D
Clients-side:
f
Enc


Dec
D
Clients–side:
f
≈
Enc



Clients-side:

Private Training-set:

X_{pr}

f

Server-side:

D

(ex s)

Public Training-set:

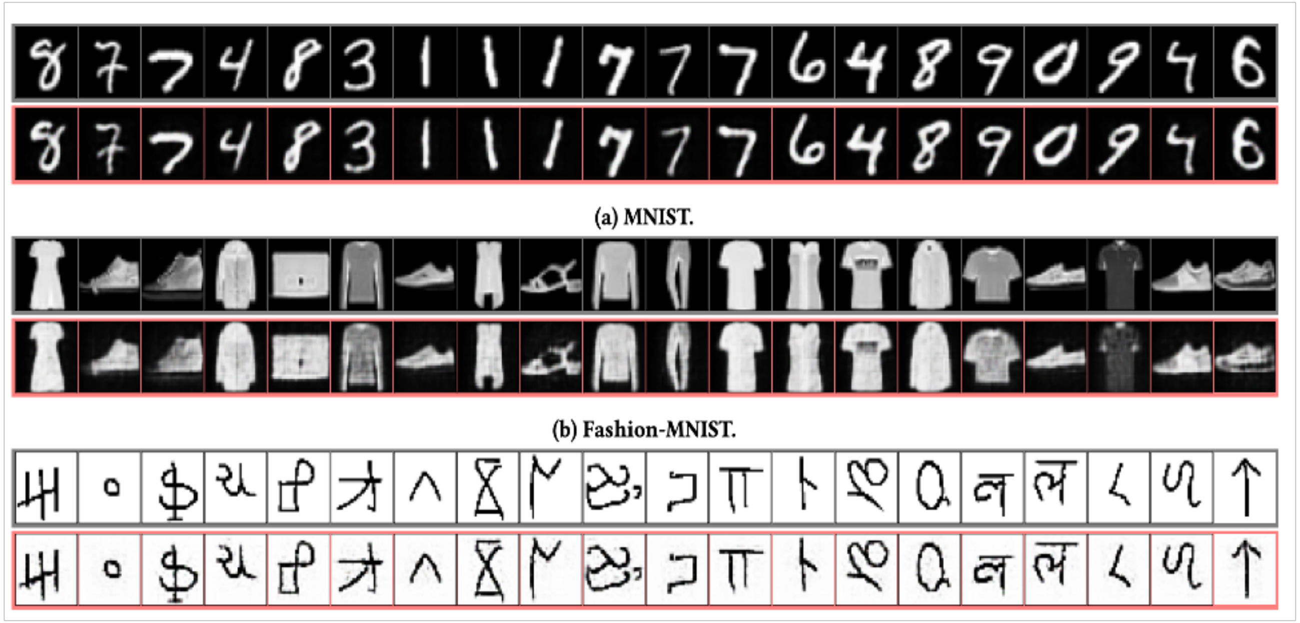
X_{pu}

Enc

Dec

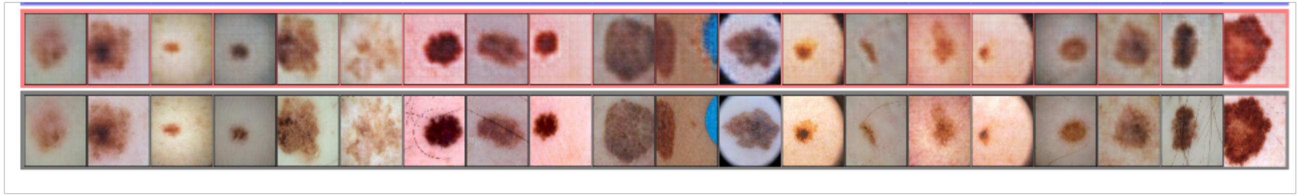
Private data:

Reconstructed data:



(a) MNIST.

(b) Fashion-MNIST.



Conclusions

What we should have learned:

What we should have learned:

- “**CML is privacy preserving**” has been **mistakenly normalized** by the scientific community:
    - Despite the huge interest and research throughput:
        - Current protocols are not a solution for your privacy issues.

What we should have learned:

- **“CML is privacy preserving”** has been **mistakenly normalized** by the scientific community:
  - Despite the huge interest and research throughput:
    - Current protocols are not a solution for your privacy issues.

- Usually, trying to patch something inherently insecure does not bring anywhere.
  - Many **existing** techniques to improve CML’s privacy don′t actually help.

What we should have learned:

- **“CML is privacy preserving”** has been **mistakenly normalized** by the scientific community:
  - Despite the huge interest and research throughput:
    - Current protocols are not a solution for your privacy issues.

- Usually, trying to patch something inherently insecure does not bring anywhere.
  - Many **existing** techniques to improve CML’s privacy don′t actually help.

- The only suitable direction to solve CML is to embrace formal security definitions:
  - End-to-end cryptography (with sound threat models).
  - At worst, weaker forms of privacy such as Differential Privacy (with sound threat models and met assumptions).
- **and accept that this comes with massive efficiency & utility costs**.

What we should have learned:

- **“CML is privacy preserving”** has been **mistakenly normalized** by the scientific community:
  - Despite the huge interest and research throughput:
    - Current protocols are not a solution for your privacy issues.

- Usually, trying to patch something inherently insecure does not bring anywhere.
  - Many **existing** techniques to improve CML’s privacy don′t actually help.

- The only suitable direction to solve CML is to embrace formal security definitions:
  - End-to-end cryptography (with sound threat models).
  - At worst, weaker forms of privacy such as Differential Privacy (with sound threat models and met assumptions).
- **and accept that this comes with massive efficiency & utility costs**.

What we should have learned:

- **"CML is privacy preserving"** has been **mistakenly normalized** by the scientific community:
    - Despite the huge interest and research throughput:
        - Current protocols are not a solution for your privacy issues.

- Usually, trying to patch something inherently insecure does not bring anywhere.
    - Many **existing** techniques to improve CML's privacy don′t actually help.

- The only suitable direction to solve CML is to embrace formal security definitions:
    - End-to-end cryptography (with sound threat models).
    - At worst, weaker forms of privacy such as Differential Privacy (with sound threat models and met assumptions).
- **and accept that this comes with massive efficiency & utility costs**.

- **Everything outside this spectrum, unfortunately, offers only a "false sense of security"**.

Time for questions.

All images in the slides have been generated by DALL·E

What's next?

Differential privacy.

Or better, incorrect applications of DP:

- what happens when assumptions are not met.