CS5562: Trustworthy Machine Learning

Federated Learning → Design and Privacy Analysis

Reza Shokri[a]

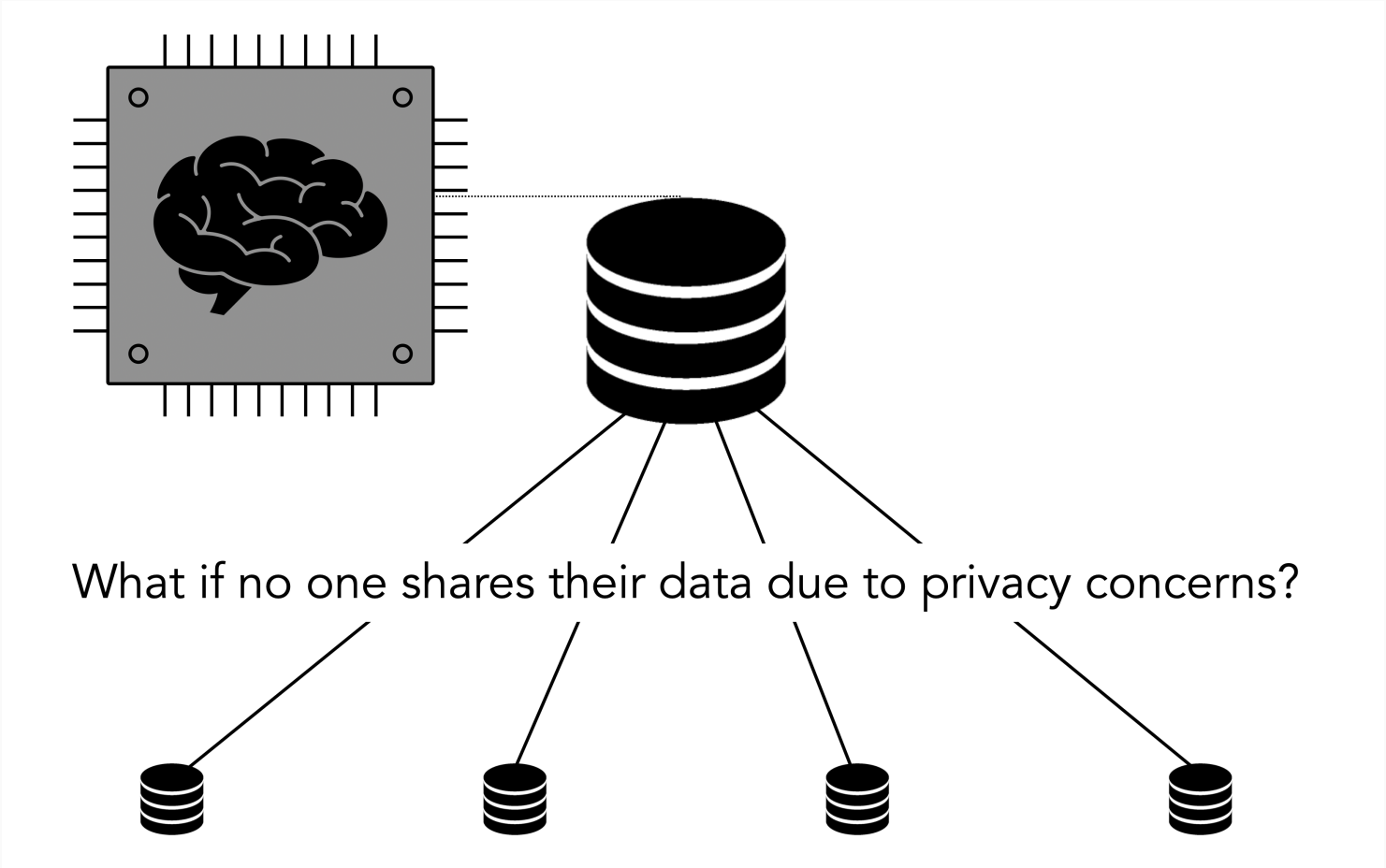
Aug 2023

[a]Acknowledgment. The wonderful teaching assistants: Hongyan Chang, Martin Strobel, Jiashu Tao, Yao Tong, Jiayuan Ye

Contents

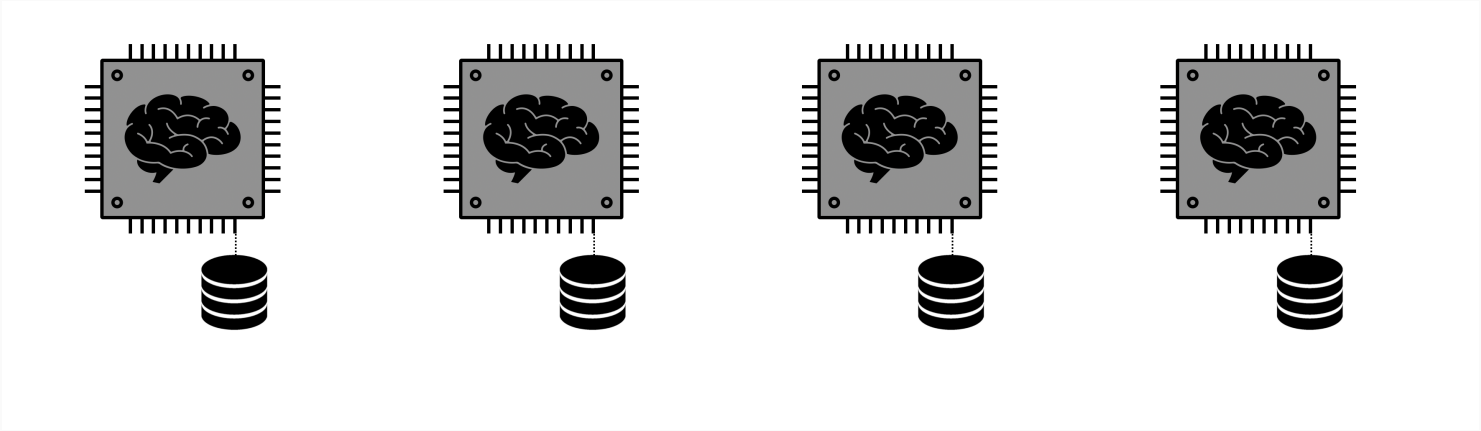
Learning in a Multi-Party Setting

Centralized Learning

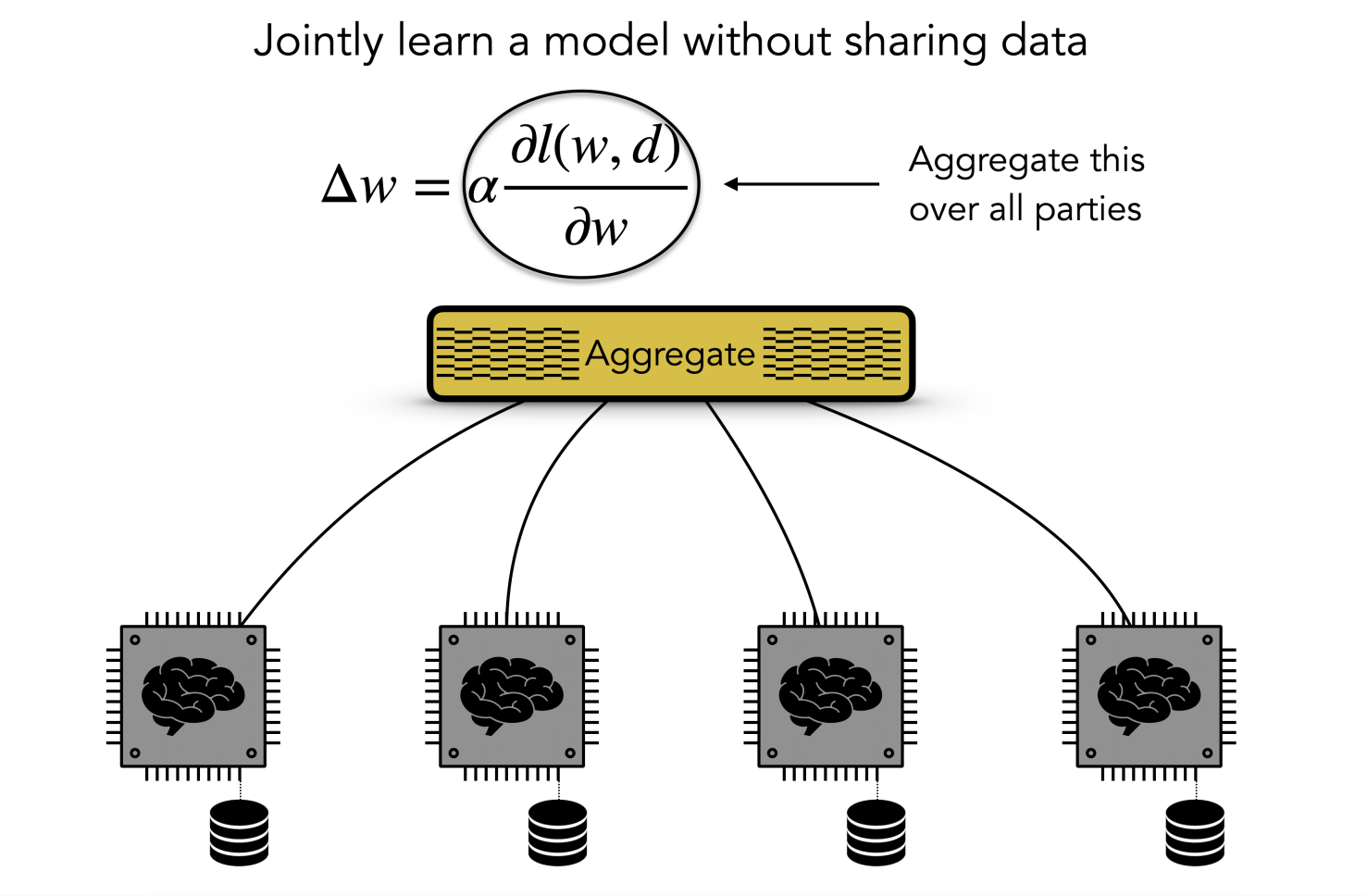


Standalone Learning

Parties do not share their data. Models, however, overfit to local data

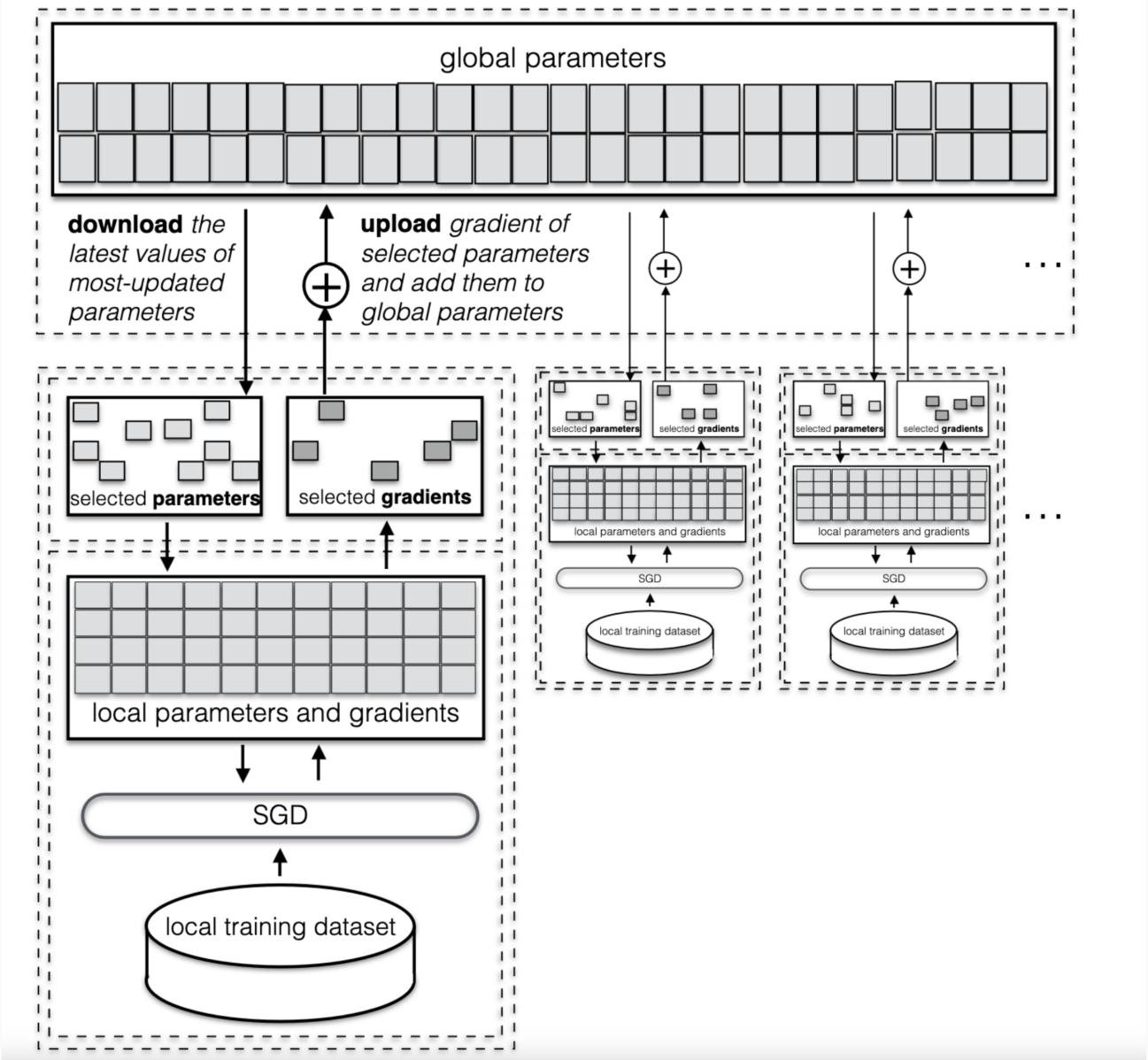


De-centralized Learning



Source: [Shokri and Shmatikov, 2015]

De-centralized Learning



De-centralized Learning

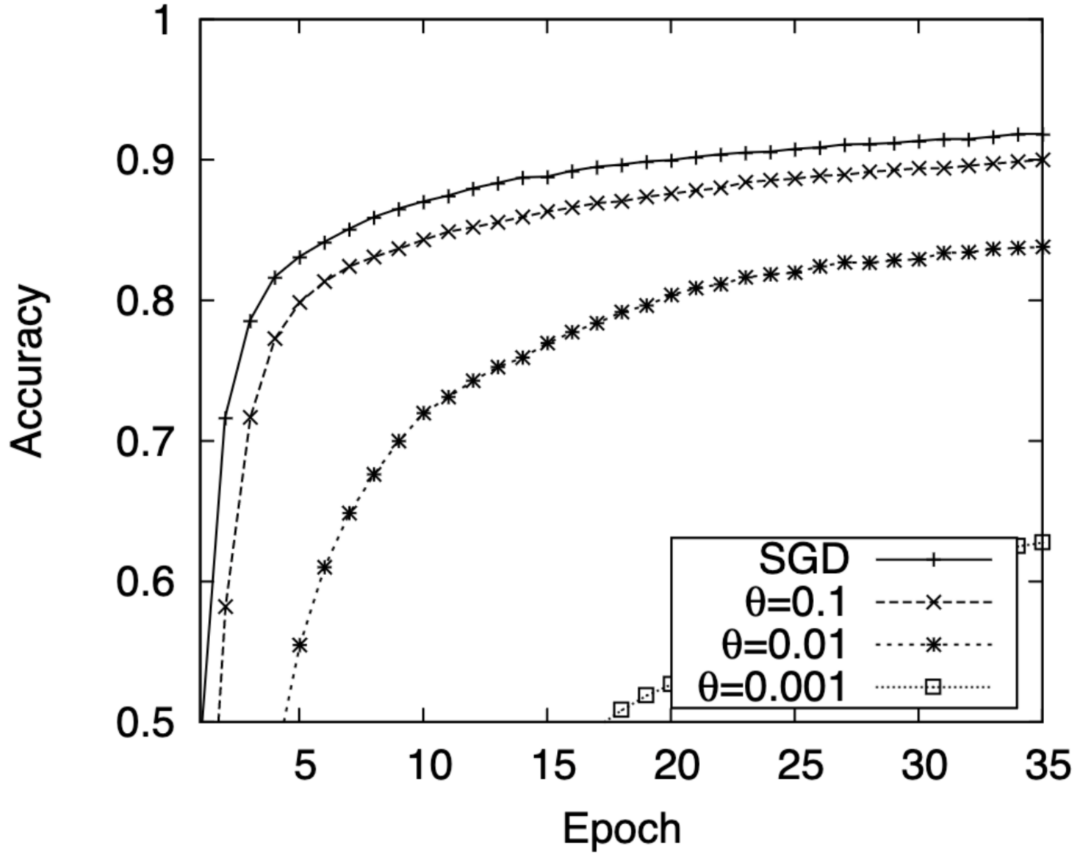
Choose initial parameters $\mathbf{w}^{(i)}$ and learning rate α.

Repeat until an approximate minimum is obtained:

1. Download $\theta_d \times |\mathbf{w}^{(i)}|$ parameters from server and replace the corresponding local parameters.
2. Run SGD on the local dataset and update the local parameters $\mathbf{w}^{(i)}$ according to $w_j := w_j - \alpha \frac{\partial E_i}{\partial w_j}$
3. Compute gradient vector $\Delta\mathbf{w}^{(i)}$ which is the vector of changes in all local parameters due to SGD.
4. Upload $\Delta\mathbf{w}_S^{(i)}$ to the parameter server, where S is the set of indices of at most $\theta_u \times |\mathbf{w}^{(i)}|$ gradients that are selected according to one of the following criteria:
   - *largest values*: Sort gradients in $\Delta\mathbf{w}^{(i)}$ and upload θ_u fraction of them, starting from the biggest.
   - *random with threshold*: Randomly subsample the gradients whose value is above threshold τ.

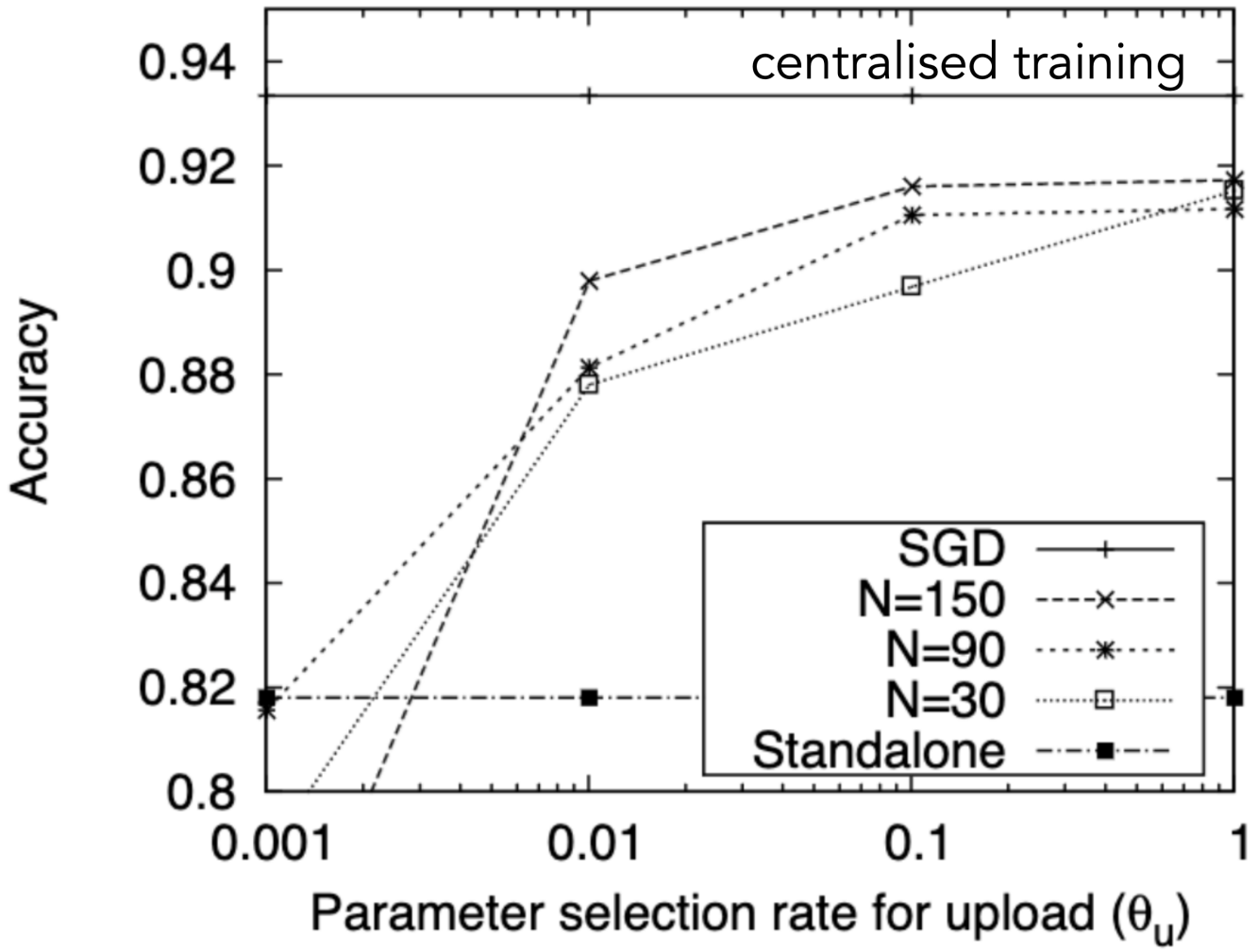
   The selection criterion is fixed for the entire training.

De-centralized Learning



A small fraction of the gradient vector is enough for training

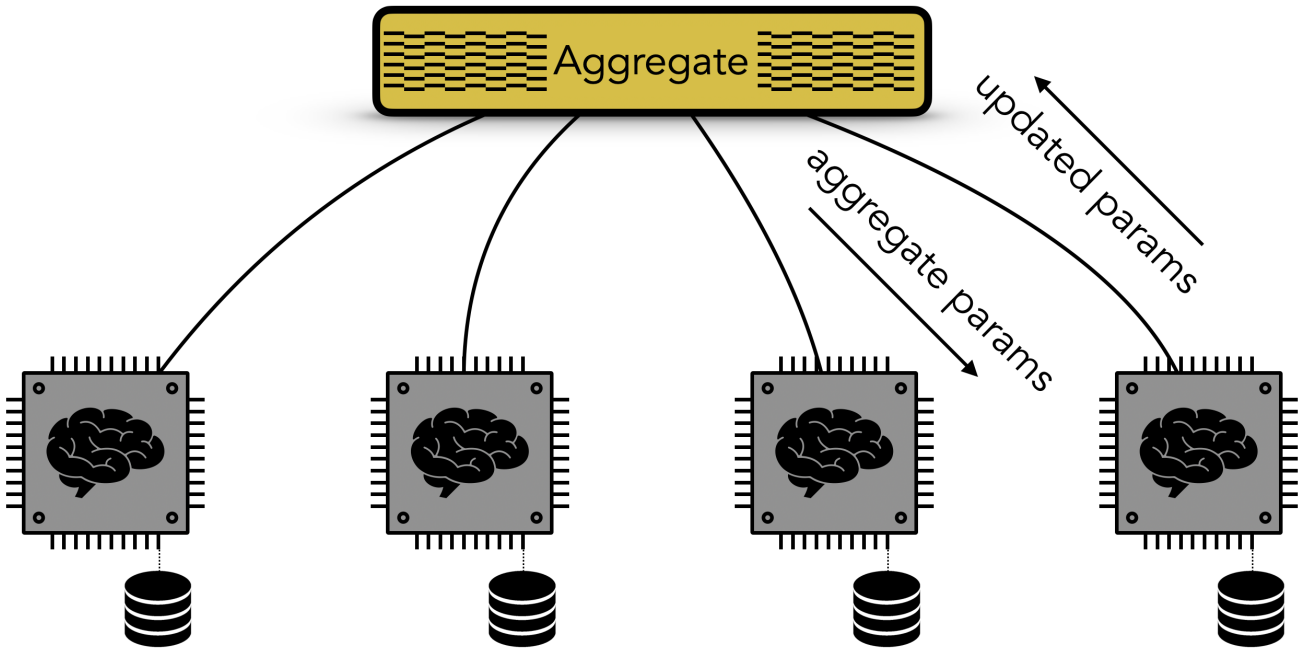
De-centralized Learning



Asynchronous decentralised SGD (FedSGD)

Federated Learning via Averaging Parameters

Aggregate **Parameters** instead of aggregating gradients



FedAvg: Algorithm

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

initialize w_0

for each round $t = 1, 2, \dots$ **do**

$m \leftarrow \max(C \cdot K, 1)$

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

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

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

$m_t \leftarrow \sum_{k \in S_t} n_k$

$w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*

$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)

for each local epoch i from 1 to E **do**

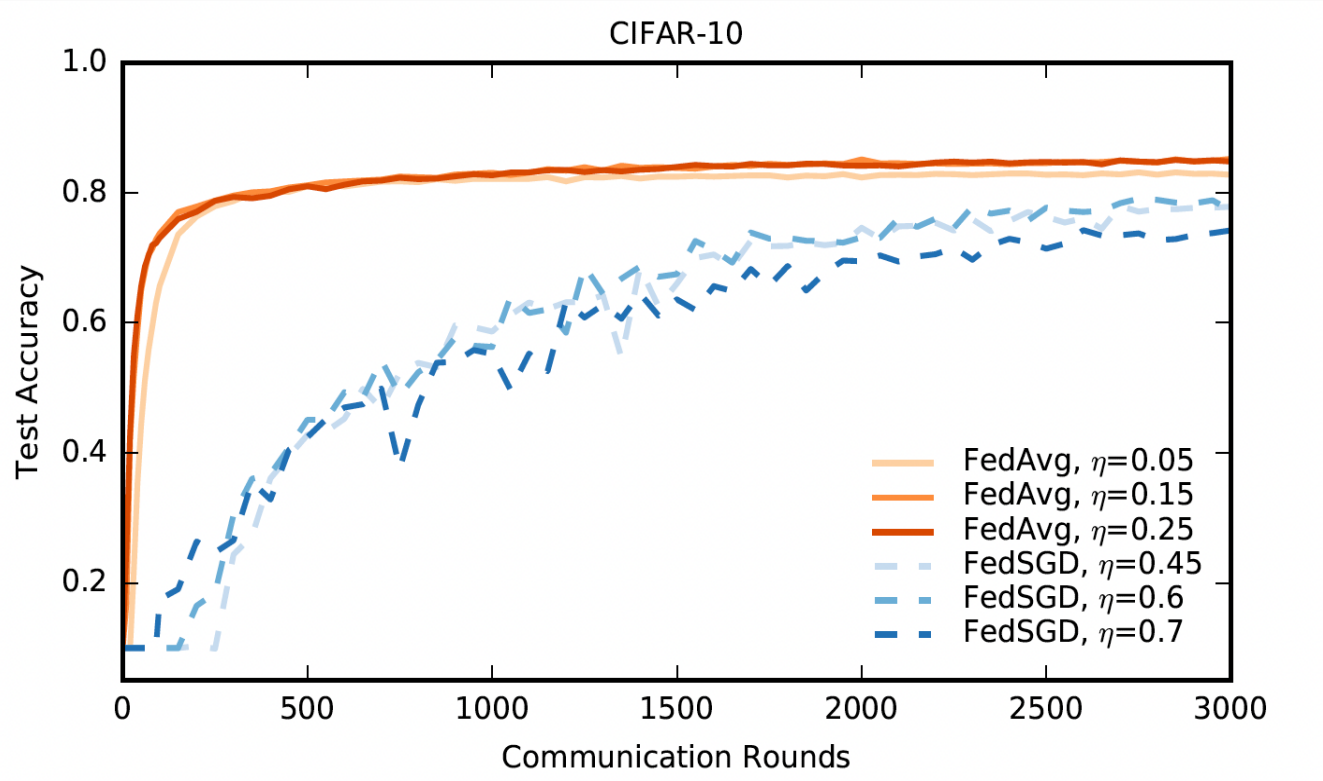
for batch $b \in \mathcal{B}$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

return w to server

Source: [McMahan et al., 2017a]

Performance of FedSGD and FedAvg



Performance on CIFAR-10. Clients train models 5 epochs locally in FedAvg in each round.

Source: [McMahan et al., 2017a]

Applications of Federated Learning at Apple

ARTIFICIAL INTELLIGENCE

How Apple personalizes Siri without hoovering up your data

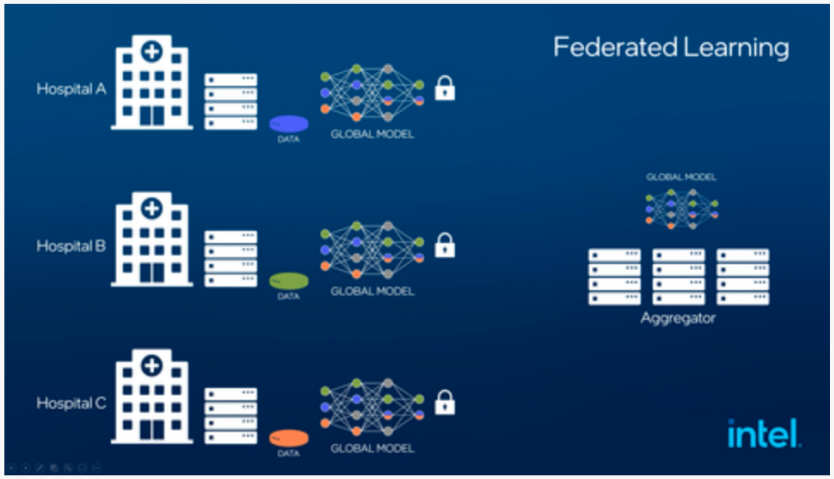
The tech giant is using privacy-preserving machine learning to improve its voice assistant while keeping your data on your phone.

By Karen Hao December 11, 2019

Source: https://www.technologyreview.com/2019/12/11/131629/apple-ai-personalizes-siri-federated-learning/

Applications of Federated Learning at Intel

Intel and Penn Medicine Announce Results of Largest Medical Federated Learning Study



Using Intel federated learning technology paired with Intel Software Guard Extensions (SGX), researchers were able to address numerous data privacy concerns by keeping raw data inside the data holders' compute infrastructure and only allowing model updates computed from that data to be sent to a central server or aggregator, not the data itself. (Credit: Intel Corporation)

Source: https://www.intc.com/news-events/press-releases/detail/1593/intel-and-penn-medicine-announce-results-of-largest-medical

Applications of Federated Learning at NVIDIA

NVIDIA releases AI-based healthcare tools for hospitals, research organizations

The platform is aiding researchers at Massachusetts General Hospital develop an AI model to more accurately diagnose brain aneurysms.

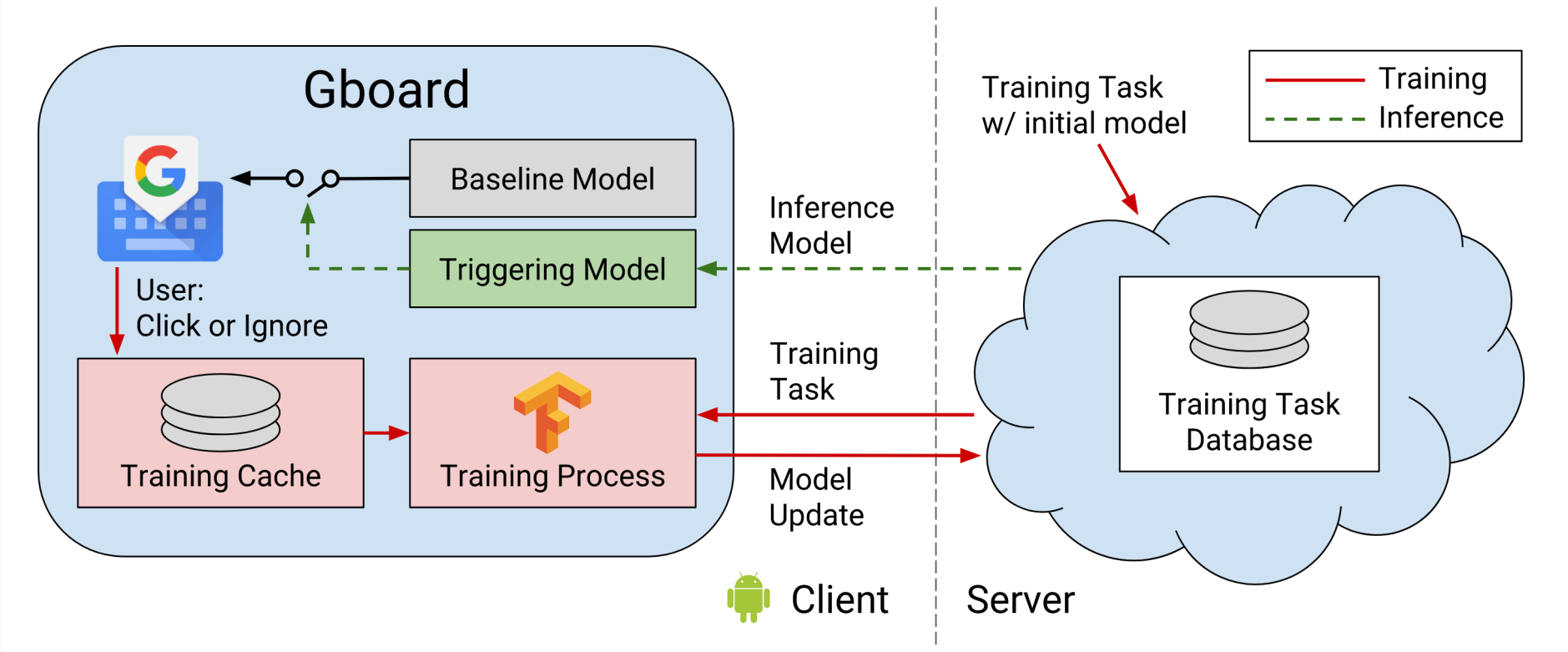
Nathan Eddy

Photo: Westend61/Getty Images

Chipmaker NVIDIA announced the launch of FLARE (Federated Learning Application Runtime Environment), an open-source software platform offering a common computing foundation designed to improve collaboration on AI model development in healthcare.

Source: https://www.healthcarefinancenews.com/news/nvidia-releases-ai-based-healthcare-tools-hospitals-research-organizations

Applications of Federated Learning at Google



Source: [Yang et al., 2018]

Cross-Device Federated Learning

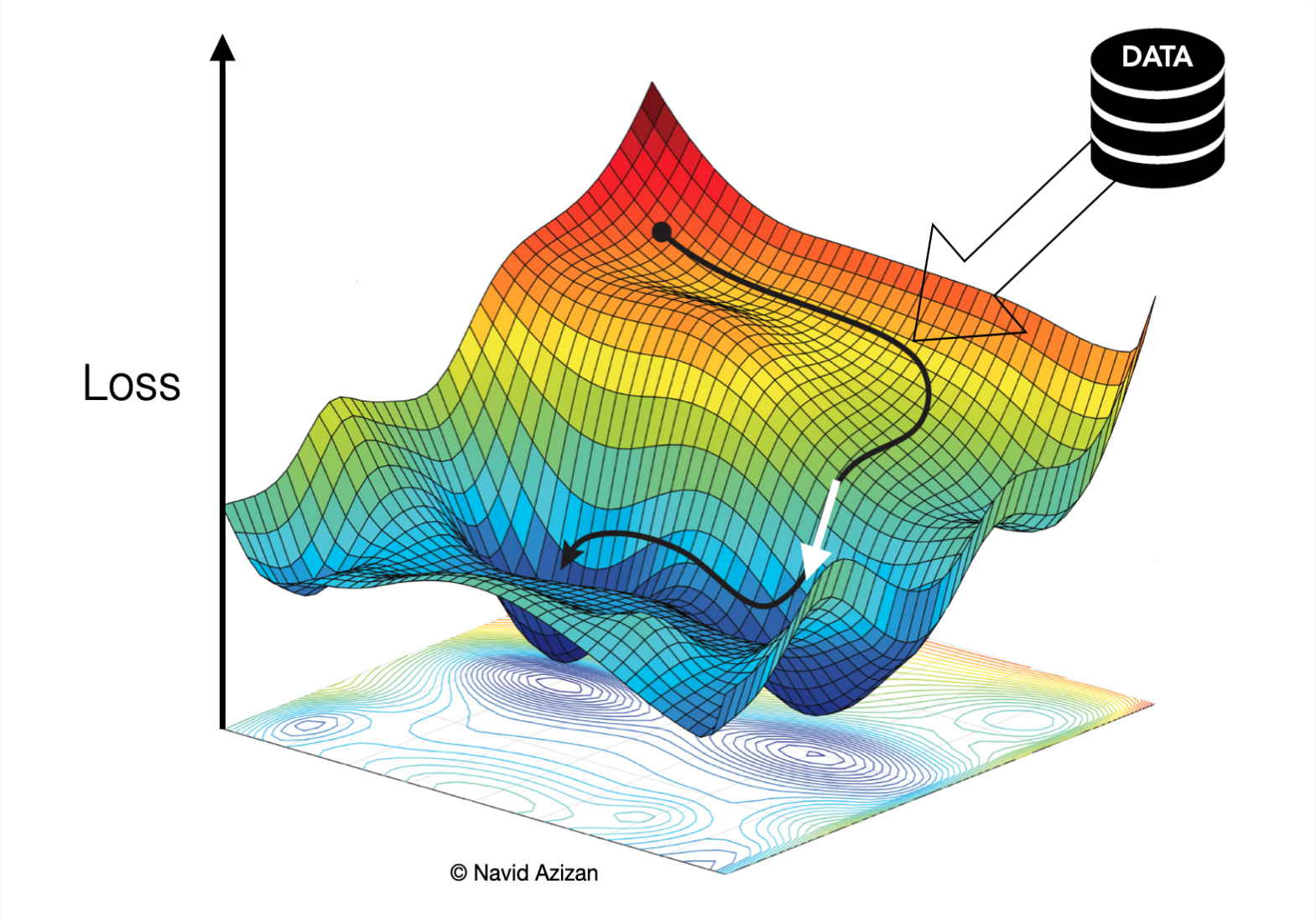
- Involves millions of intermittently available clients, e.g., Gboard.
- Server can only access a small fraction of clients in each round.
- Most clients may only participate once.
- Communication is usually the bottleneck.
- Data is horizontally partitioned.

Cross-Silo Federated Learning

- Involves a smaller number of clients with high availability, e.g., institutions.
- Most clients participate in every round.
- Both communication and computation can be the bottleneck.
- Data can be horizontally or vertically partitioned.
- In the context of vertical partitioning, consider a scenario where a health insurance company wishes to collaborate with hospitals to evaluate the health risk of an individual. The insurance company possesses the individual's insurance records, whereas the hospitals maintain the medical records. Each institution has distinct features about the individual.

Is FL Privacy Preserving?

Information Leakage via the Gradient



Model updates can reveal information about the training data.

Information Leakage via the Gradient

- Deep learning models usually utilize an embedding layer to convert such discrete inputs into continuous, lower-dimensional vector representations.
- We use *words* to denote discrete tokens.
- Each word from the training data gets its vector representation from the embedding matrix $W_{emb} \in \mathbb{R}^{|V| \times d}$, where $|V|$ is the vocabulary size.
- For a word v in the vocabulary, its representation is computed as follows:

$$w_{vector} = W_{emb}(v) \tag{1}$$

Source: [Melis et al., 2019]

Information Leakage via the Gradient - Continued

- When clients update the model, the embedding matrix for a word v is updated as follows:

$$W_{emb}(v) - \eta \nabla_{W_{emb}(v)} \text{Loss} \tag{2}$$

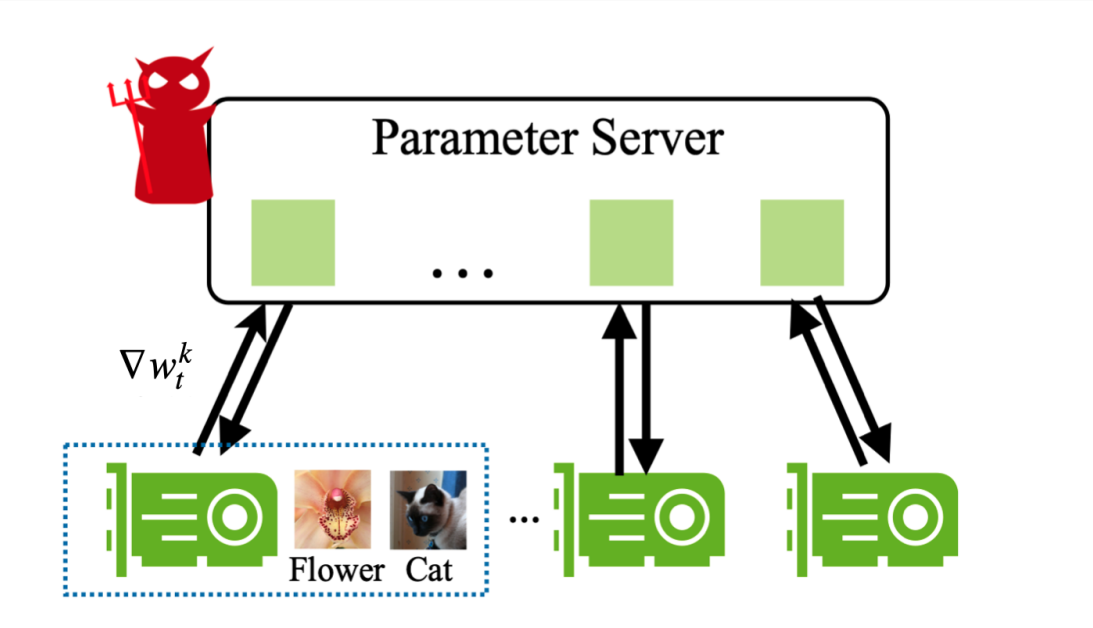
- Only the vectors of words present in the training dataset get updated.
- If a word isn't in the training dataset, its vector won't affect the model's predictions. Hence, its gradient remains 0.
- When the adversary knows the mapping between the word and the embedding vector, they can infer the membership of the word in the training dataset via gradients.

Source: [Melis et al., 2019]

Information Leakage via the Gradient

- The previous example only demonstrates the leakage of words, but it is hard to infer the original sentence due to ambiguity.
- *Is it possible to reconstruct the training data?*
- Let us see an example when the client updates the model on a single data point.

Deep Leakage via Gradient



- Given the gradient ∇w_t^k, can the adversary generate (x, y) matches the record in training dataset D_k from client k?

Source: [Zhu et al., 2019]

Deep Leakage Through Gradients

- Identify an example for which the gradient when computed on w, closely matches ∇w.

Source: [Zhu et al., 2019]

Deep Leakage Through Gradients

- Randomly initialize a dummy input x' and label input y'.
- Feed these "dummy data" into models and get "dummy gradients".

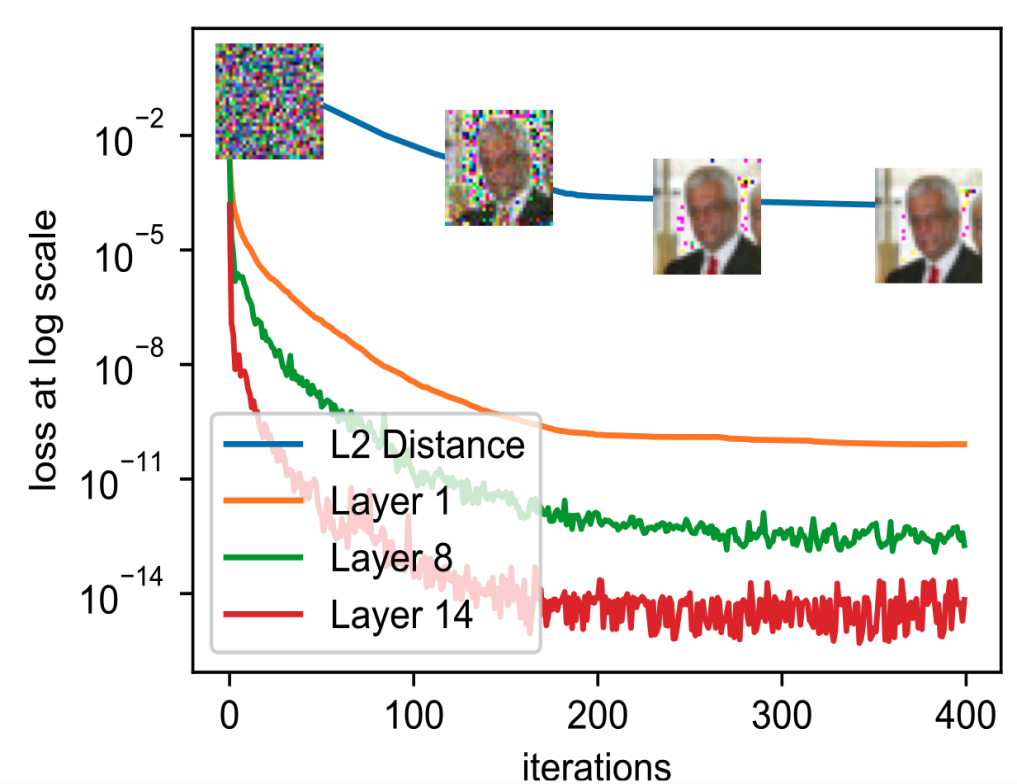
$$\nabla w' = \frac{\nabla \ell(w_{t-1}; (x', y'))}{w} \tag{3}$$

- Optimized the following objective using standard gradient-based methods

$$\nabla x'^{*}, y'^{*} = \arg\min_{x', y'} \| \nabla w' - \underbrace{\nabla w_t^k}_{\text{Local updates from client } k} \| \tag{4}$$

- We need to make a mild assumption that model prediction is twice differentiable, which holds for the majority of modern machine learning models (e.g., most neural networks) and tasks.

Deep Leakage via Gradient



Results on LFW dataset with ResNet56. Layer-i means MSE between real and dummy gradients of i-th layer. When the gradients' distance gets smaller, the MSE between the leaked and original images also gets smaller.

Source: [Zhu et al., 2019]

Membership Information Leakage

- Since the model sharing is not privacy-preserving, what are the privacy risks for clients during FL?
- Remember that we audit the privacy risks in a centralized setting using membership inference attacks.
- Let us analyze the privacy risks in FL in the same manner.

Threat Model in FL

Position of the Attacker

- Centralized parameter server
- One of the participants

Both types of attackers can observe *multiple* snapshots of the *model parameters*

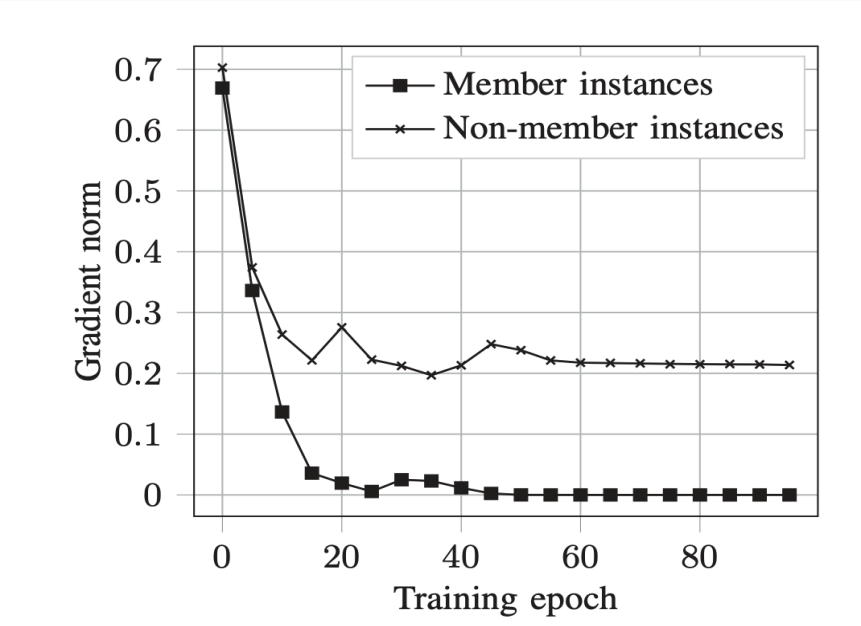
Attack Model

- Passive attack (make observations without modifying the learning process)
- Active attack (actively influence the target model to extract more information about its training set)

Source: [Nasr et al., 2019]

Additional Information Leaked in FL - White-box Access

White-box access: The adversary can access the model parameters, which may encode more information.



Gradient norms of the last layer during learning epochs for member and non-member instances (for Purchase100).

Source: [Nasr et al., 2019]

Additional Information Leaked in FL - White-box Access

White-box access: The adversary can access the model parameters, which may encode more information.

Pre-trained Target Model				Attack Accuracy		
Dataset	**Architecture**	**Train Accuracy**	**Test Accuracy**	**Black-box**	**White-box (Outputs)**	**White-box (Gradients)**
CIFAR100	Alexnet	99%	44%	74.2%	74.6%	75.1%
CIFAR100	ResNet	89%	73%	62.2%	62.2%	64.3%
CIFAR100	DenseNet	100%	82%	67.7%	67.7%	74.3%
Texas100	Fully Connected	81.6%	52%	63.0%	63.3%	68.3%
Purchase100	Fully Connected	100%	80%	67.6%	67.6%	73.4%

Large capacity

High generalizability (Best available model)

Low privacy (Significant leakage through parameters)

Source: [Nasr et al., 2019]

Additional Information Leaked in FL - Multiple Observations

Multiple Observations: The adversary can observe the whole training process.

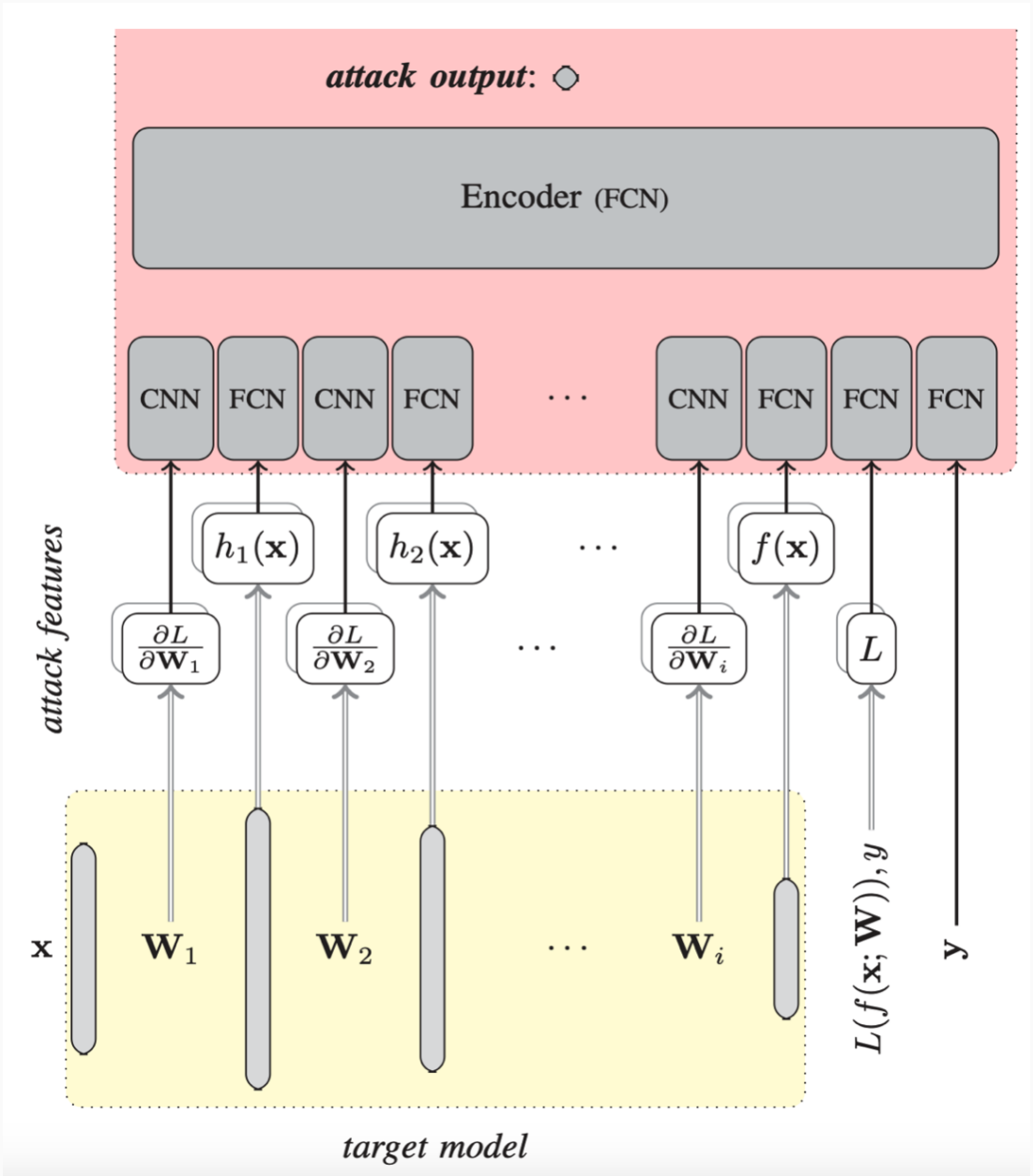
- By simply averaging the loss over the last 10 rounds as the membership signal, the adversary can achieve an AUC of 0.712 compared to 0.687 when using the last round only.

Training an attack model for extracting the membership signal

How can we extract the membership information from white-box access from multiple snapshots of the model?

- We assume that the adversary has access to some portion of the training dataset and test dataset of a client k.
- The adversary can train a powerful attack model to distill all the membership information encoded in different signals (e.g., activations, gradients) into a single membership signal.
- To make use of multiple observations, the adversary concatenates each signal history, using it as an input feature for the attack model.

Attack Model



MIA Performance in FL

Observed Epochs	Attack Accuracy
5, 10, 15, 20, 25	57.4%
10, 20, 30, 40, 50	76.5%
50, 100, 150, 200, 250	79.5%
100, 150, 200, 250, 300	85.1%

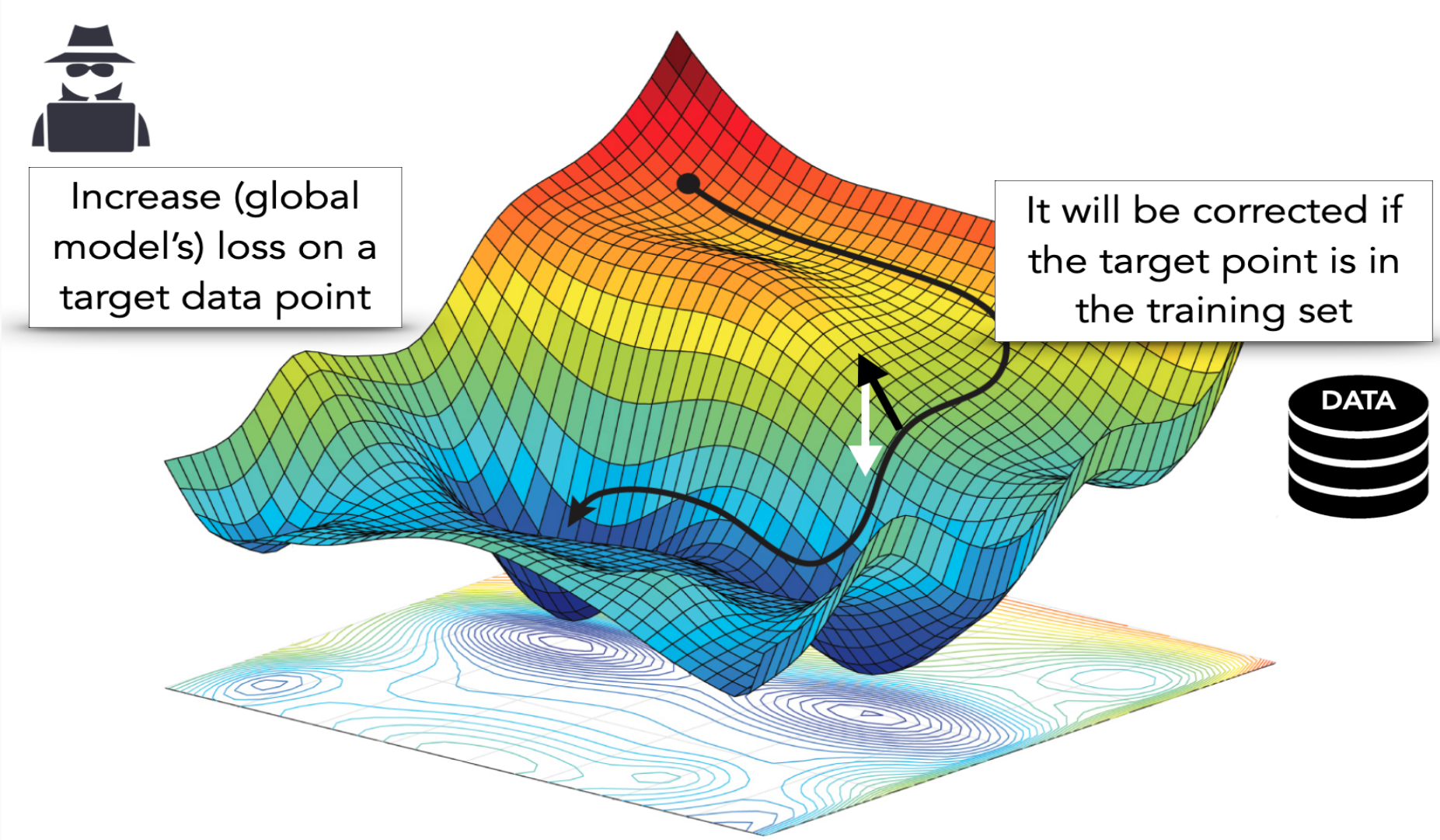
Accuracy of the passive global attacker in the FL when the attacker uses various training epochs. (CIFAR100-Alexnet).

Additional Information Leaked in FL - Active Attack

Active Attack: An adversarial server/participant can influence others' models

- Can active attacks (model poisoning) be used to infer more information on whether a target data is part of the training set?

Gradient Ascent Attack



Gradient Ascent Attack

Model	Global Attacker (the parameter aggregator)				Local Attacker (a participant)	
CIFAR100	Passive	Active			Passive	Active
Architecture		Gradient Ascent	Isolating	Isolating Gradient Ascent		Gradient Ascent
Alexnet	85.1%	88.2%	89.0%	92.1%	73.1%	76.3%
DenseNet	79.2%	82.1%	84.3%	87.3%	72.2%	76.7%

Privacy-Preserving FL

Defense against Privacy Attacks in FL

How can a client protect its privacy in FL?

Differential Privacy in FL

Applying differential privacy is an answer.

- **Client-level differential privacy**: The server is *trusted* and adds noise to ascertain whether a client's gradient is part of the aggregation. This has a limited impact on the aggregated model.
- **Record-level differential privacy**: The server *is not trusted*. Thus, clients add noise to their local updates to ensure that the inclusion or exclusion of a specific data point in the training dataset has a limited impact on the local updates.

In either case, applying DP in FL has a significant utility cost.

We will only cover client-level DP for simplicity of presentation.

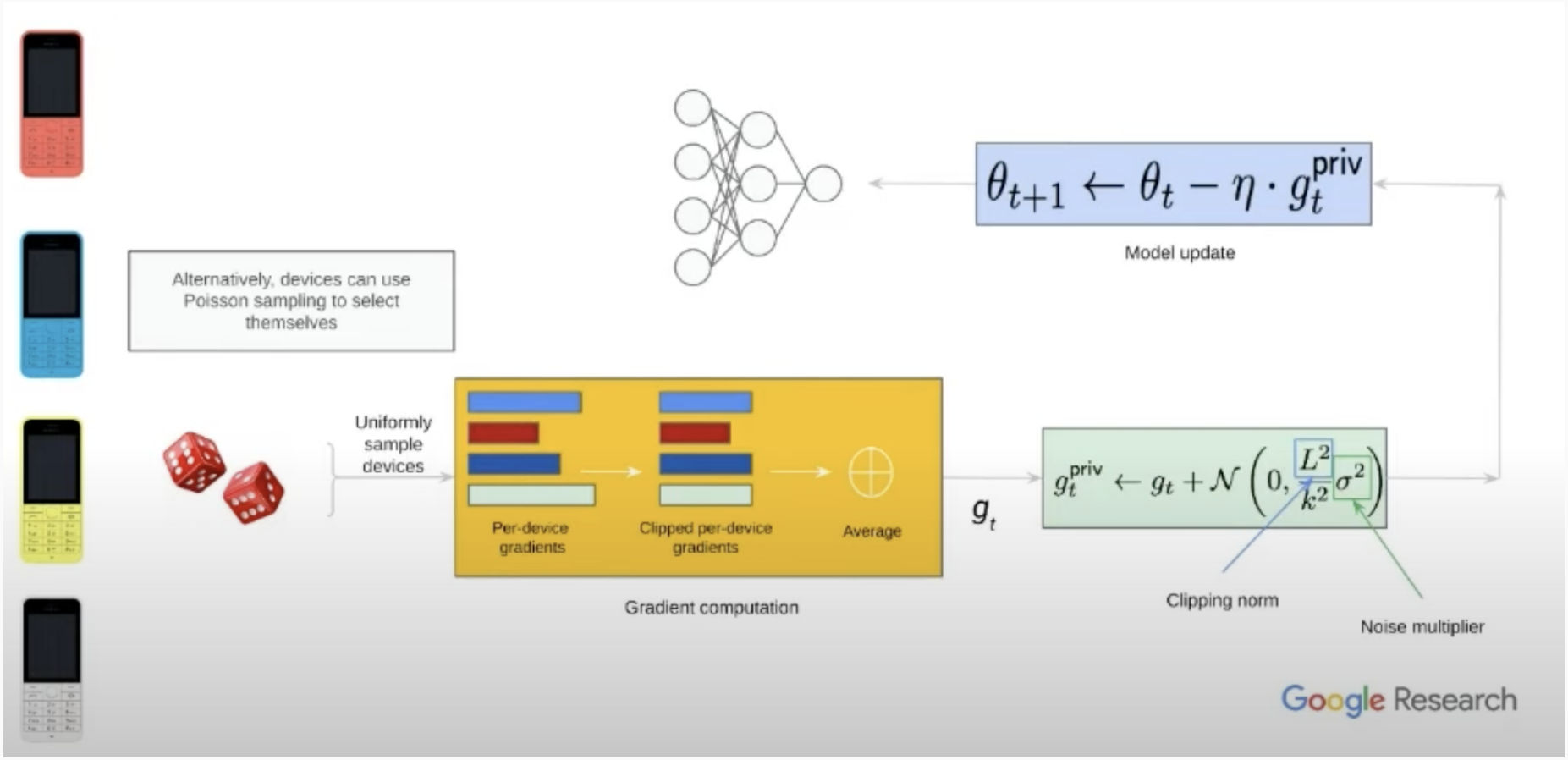
Ensuring Client-level Differential Privacy

A randomized algorithm $\mathcal{A}$ satisfies (ε, δ)-client-level DP, if for any two datasets D and D' that differ in **one user's** data records, and all sets S

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in S] + \delta$$

- Server is *trusted*
- Server *adds noise* to ascertain whether a client is part of aggregation

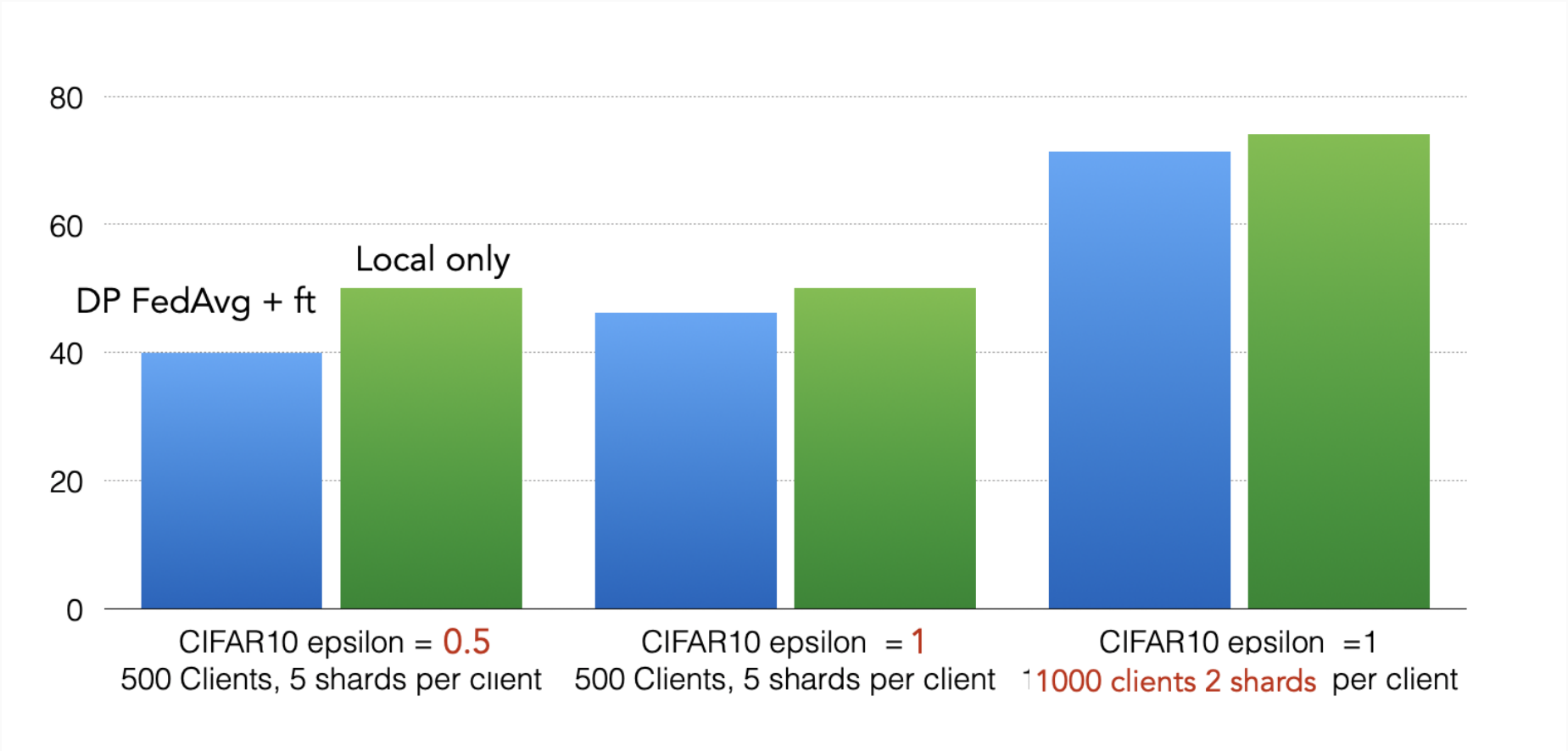
Client-level Differentially Private Federated Averaging



[McMahan et al., 2017b]

DP-FedAvg + Fine-tuning

- For better performance, each client can further **fine-tune** the global model (trained via DP-FedAvg) **on individual data** (e.g. fine-tuning classification head but freeze other layers)
- This further fine-tuning step is **free of privacy cost** because the local fine-tuned model is *not* released outside each local client.



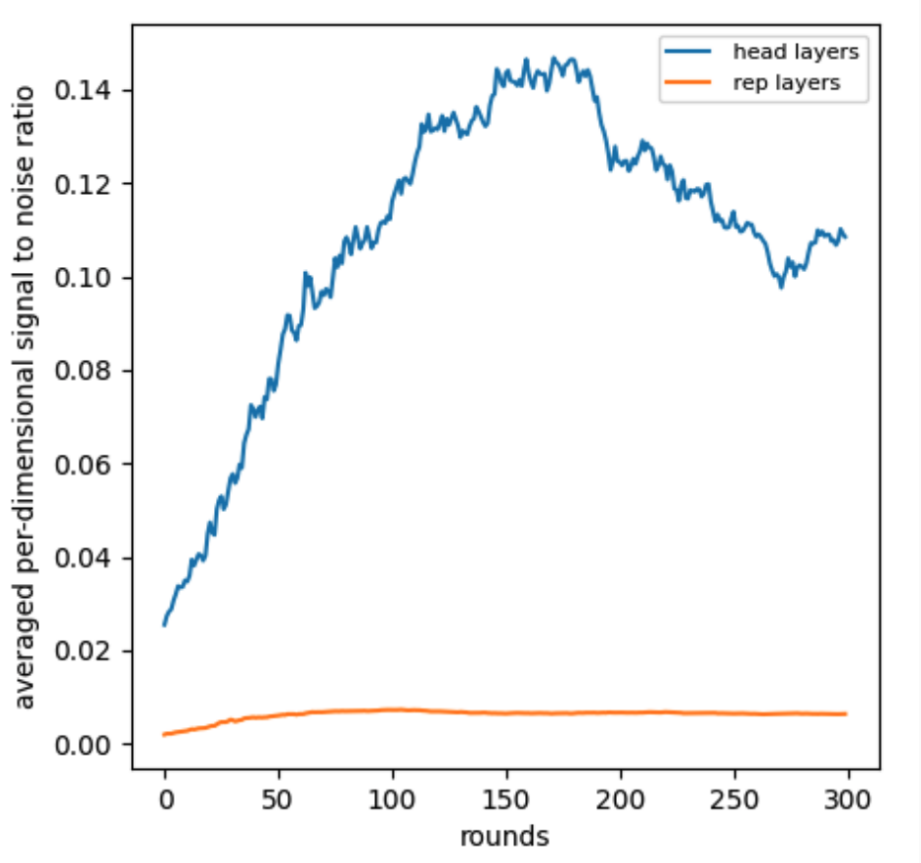
In many cases under **data heterogeneity**, DP-FedAvg + finetuning could *not* outperform **local-only** training (especially for small privacy budget and number of clients)

What restricts DP-FedAvg + Fine-tuning from accurately learning representations under differential privacy?

- Standard non-dp algorithm FedAvg + ft is known to have remarkable ability to learn accurate representations, even under data heterogeneity [Collins et al., 2022, Yu et al., 2020]
- DP FedAvg introduces per-example gradient **clipping** and additive **noise**, which then **degrades** the quality of learned representation.

Effect of Clipping and Additive Noise

The averaged per-dimensional **signal to noise ratio** (or gradient value) on classification **heads** is significantly larger than that on the **representation** layers



Per-dimensional signal-to-noise ratio is the absolute value of gradient after clipping divided over the noise standard deviation

Could we protect privacy via client-level DP, while still benefiting from collaborating?

How to learn accurate representation with differential privacy?

Standard FL Objective (Global Concensus):

$$\arg\min_{\theta} \mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; D_i) \text{ where } \ell(\theta; D_i) = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(f_\theta(x), y)$$

Issues: DP sharing of heterogeneous classification heads reduces signal-to-noise ratio for other layers, thus hindering accurate representation learning

How to learn accurate representation with differential privacy?

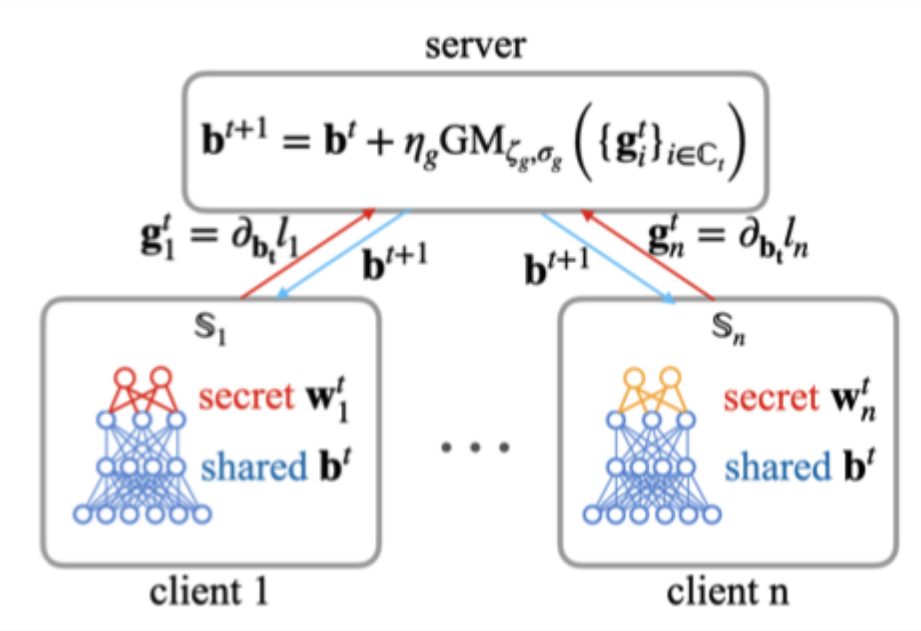
New Federated Representation Learning Objective:

$$\arg\min_{b} \frac{1}{n} \sum_{i=1}^{n} \min_{w_i} \left\{ \ell([w_i, b]; D_i) = \frac{1}{|D_i|} \sum_{(x,y) \in D_i} \ell(f_{w_i} \circ f_b(x), y) \right\}$$

Share the (Training of) Representation Only: clients jointly learn the representation layer parameters b, while each client i separately optimize the classification head w_i on its own data D_i

The Centaur Algorithm [Shen et al., 2023]

In each global round t, clients keep their classification head w_i^t secret while updating the shared representation $b^t \rightarrow b^{t+1}$ based on perturbed gradients g_i^t from sampled clients $i \in \mathbb{C}_t$



- Built on the non-DP Fed-Rep algorithm [Collins et al., 2021], and is similar to many other DP model personalization methods [Jain et al., 2021, Bietti et al., 2022]

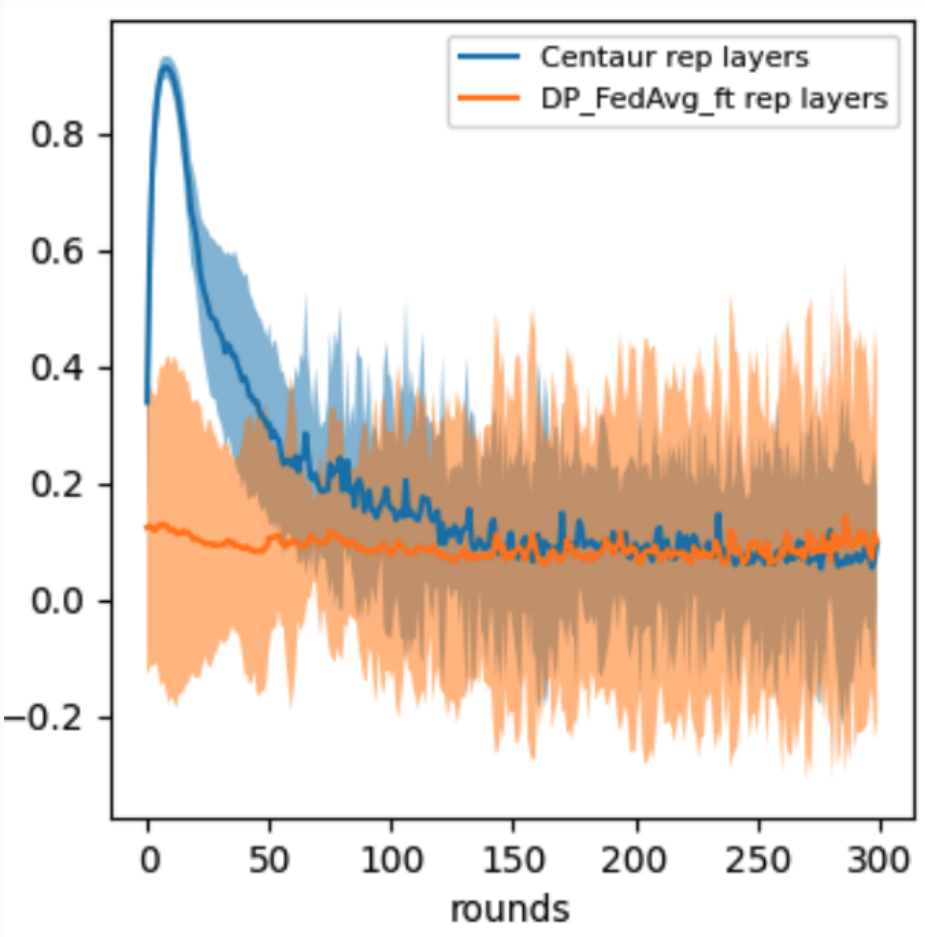
The Centaur Algorithm [Shen et al., 2023]

Algorithm 2 CLIENT procedure of CENTAUR in the general case (for client i)

procedure CLIENT($\boldsymbol{b}^t$, $\bar{m}$, T_l, η_l)
  [Phase 1: Local classifier update.] $\boldsymbol{w}_i^{t+1} = \arg\min_{\boldsymbol{w}} l([\boldsymbol{b}^t, \boldsymbol{w}]; \mathbb{S}_i)$.
  [Phase 2: Local representation function update.] Set $\boldsymbol{b}_i^{t,0} = \boldsymbol{b}^t$;
  for $s \leftarrow 0$ to $T_l - 1$ **do**
    Sample a subset $\mathbb{S}_i^s$ of size $\bar{m}$ from the local dataset $\mathbb{S}_i$ without replacement
    Update the local representation function $\boldsymbol{b}_i^{t,s+1} := \boldsymbol{b}_i^{t,s} - \eta_l \cdot \partial_{\boldsymbol{b}} l([\boldsymbol{b}_i^{t,s}, \boldsymbol{w}_i^{t+1}]; \mathbb{S}_i^s)$.
  [Phase 3: Summarize the local update direction.] **return** $\boldsymbol{g}_i^t := \boldsymbol{b}_i^{t,T_l} - \boldsymbol{b}^t$.

Only Sharing Representation Enables Higher Agreement among Clients

Clients **agree more** (shown by higher mean and lower std) on the direction to update the representation layers **in Centaur** than **in DP FedAvg + fb**



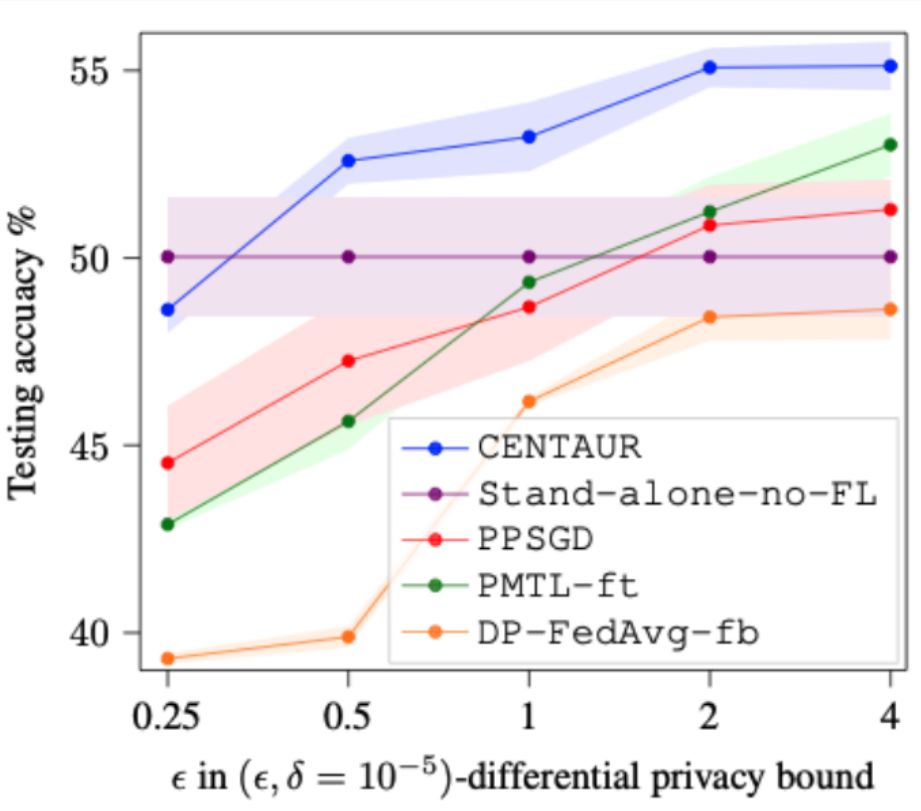
Agreement Metric: cosine similarity between gradient of each client and the average gradient over all clients (*before* clipping)

Improved Privacy Utility Trade-off under Data Heterogeneity

Under fixed privacy budget ($\varepsilon = 1$), **Centaur** consistently enables **higher** utility than **local-only** training across different data heterogeneity levels

Improved Privacy Utility Trade-off Across Privacy Budgets

- Under a fixed level of data heterogeneity, CENTAUR enables **higher** utility than **local-only** training across privacy budgets $0.5 \leq \varepsilon \leq 4$
- Centaur also outperforms DP-FedAvg + finetuning and other DP model personalization algorithms



Conclusion

- Federated learning allows clients to collaboratively train a model without directly sharing their data.
- However, FL isn't inherently "privacy-preserving".
- Adversaries in FL can infer significant information about the private training datasets.
- We can design algorithms that protect privacy via client-level differential privacy, while still benefiting from collaborating.

References i

Bietti, A., Wei, C.-Y., Dudik, M., Langford, J., and Wu, S. (2022). **Personalization improves privacy-accuracy tradeoffs in federated learning.** In International Conference on Machine Learning, pages 1945–1962. PMLR.

Chen, J., Pan, X., Monga, R., Bengio, S., and Jozefowicz, R. (2016). **Revisiting distributed synchronous sgd.** arXiv preprint arXiv:1604.00981.

Chen, M., Mathews, R., Ouyang, T., and Beaufays, F. (2019). **Federated learning of out-of-vocabulary words.** arXiv preprint arXiv:1903.10635.

References ii

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. (2021). **Exploiting shared representations for personalized federated learning.** In International conference on machine learning, pages 2089–2099. PMLR.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. (2022). **Fedavg with fine tuning: Local updates lead to representation learning.** Advances in Neural Information Processing Systems, 35:10572–10586.

References iii

Fowl, L., Geiping, J., Czaja, W., Goldblum, M., and Goldstein, T. (2021).

Robbing the fed: Directly obtaining private data in federated learning with modified models.

arXiv preprint arXiv:2110.13057.

Jain, P., Rush, J., Smith, A., Song, S., and Guha Thakurta, A. (2021).

Differentially private model personalization.

Advances in Neural Information Processing Systems, 34:29723–29735.

References iv

Kariyappa, S., Guo, C., Maeng, K., Xiong, W., Suh, G. E., Qureshi, M. K., and Lee, H.-H. S. (2023).

Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis.

In International Conference on Machine Learning, pages 15884–15899. PMLR.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017a).

Communication-efficient learning of deep networks from decentralized data.

In Artificial intelligence and statistics, pages 1273–1282. PMLR.

References v


McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2017b). **Learning differentially private recurrent language models.** arXiv preprint arXiv:1710.06963.


Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. (2019). **Exploiting unintended feature leakage in collaborative learning.** In 2019 IEEE symposium on security and privacy (SP), pages 691–706. IEEE.


Nasr, M., Shokri, R., and Houmansadr, A. (2019). **Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning.**

References vi

In 2019 IEEE symposium on security and privacy (SP), pages 739–753. IEEE.

Neyshabur, B., Sedghi, H., and Zhang, C. (2020).

What is being transferred in transfer learning?

Advances in neural information processing systems, 33:512–523.

Ramaswamy, S., Mathews, R., Rao, K., and Beaufays, F. (2019).

Federated learning for emoji prediction in a mobile keyboard.

arXiv preprint arXiv:1906.04329.

References vii

Shen, Z., Ye, J., Kang, A., Hassani, H., and Shokri, R. (2023).
Share your representation only: Guaranteed improvement of the privacy-utility tradeoff in federated learning.
In The Eleventh International Conference on Learning Representations.

Shokri, R. and Shmatikov, V. (2015).
Privacy-preserving deep learning.
In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pages 1310–1321.

References viii


Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., and Beaufays, F. (2018).
Applied federated learning: Improving google keyboard query suggestions.
arXiv preprint arXiv:1812.02903.


Yu, T., Bagdasaryan, E., and Shmatikov, V. (2020).
Salvaging federated learning by local adaptation.
arXiv preprint arXiv:2002.04758.


Zhu, L., Liu, Z., and Han, S. (2019).
Deep leakage from gradients.
Advances in Neural Information Processing Systems, 32.