Week-13: Decentralized Machine Learning

Systems for data management and data science

19.05.2025 | Akash Dhasade, Anne-Marie Kermarrec

Lecture Outline

- Advent of decentralized ML
- Federated Learning (FL)
  - FedAvg algorithm
- Decentralized Learning (DL)
- Problems with FL and DL
- Reducing communication
- Addressing data heterogeneity
- Addressing systems heterogeneity

Advent of decentralized ML

Federated Learning (FL) and Decentralized Learning (DL)

Traditionally ML models are trained in large data centres.

Users’ data was centrally collected and processed.

Surge in quantity of data

Computational complexity of training

Rising Privacy Concerns

Federated Learning

Decentralized Learning

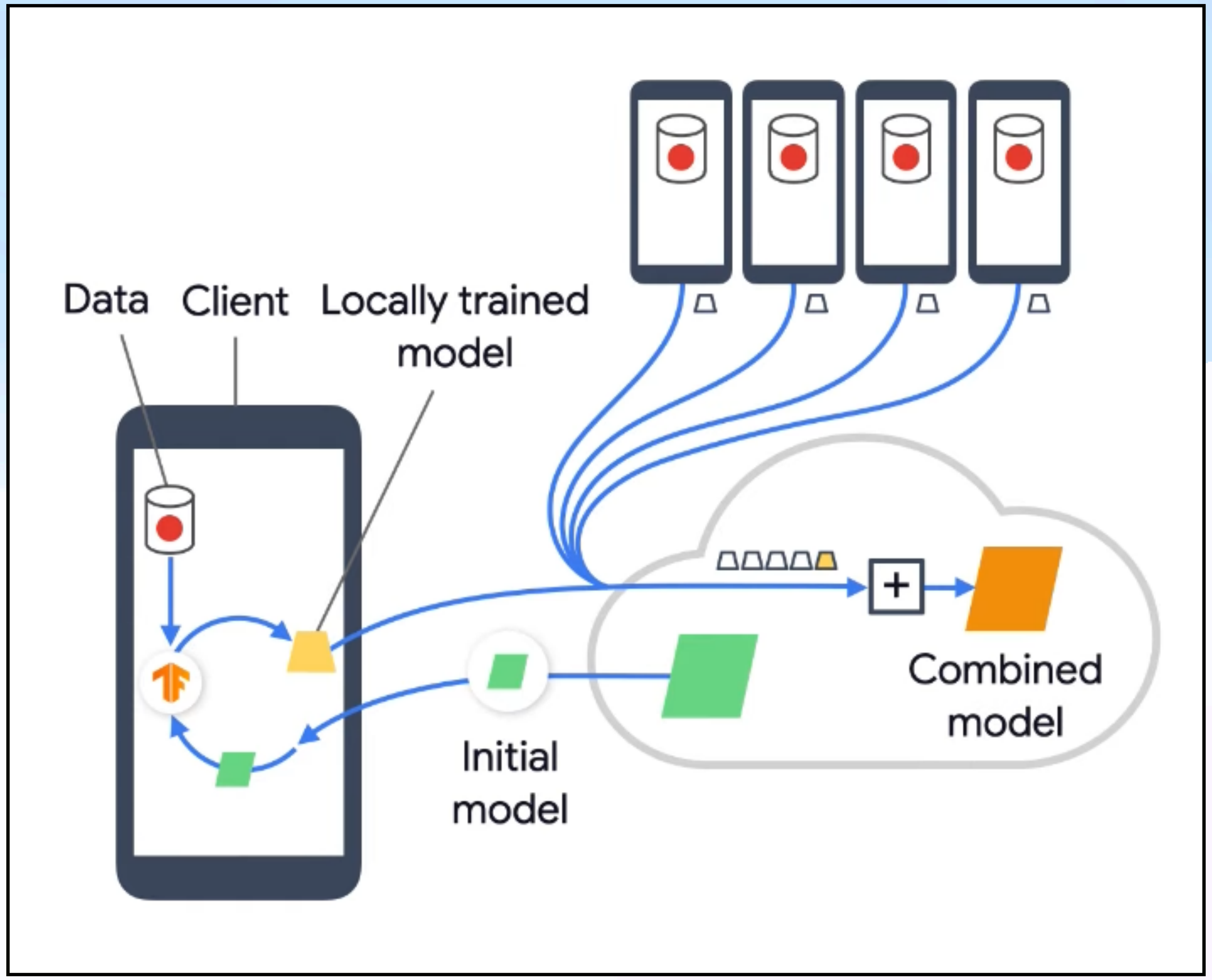
First Principle - Let the data stay where it is, learn by exchanging models

Federated Learning

— Introduced by researchers at Google around 2016-17 [1]

What is the idea ?

- A central server holds a global model
- Broadcasts the model to a set of clients i.e. edge devices and waits for a stipulated time
- Clients train on their local dataset and send the updated model back
- Server aggregates and pushes the new global model to a new set of clients



Repeats until convergence

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

Why did it become so popular ?

Several reasons, most important is privacy

Privacy advantage

Data not shared	Clients never share any data
I(Models) << I(Data)	Information processing inequality
Ephemeral model updates	Individual client models can be deleted as soon as aggregated
No identity information	Client identity information is not required

But who labeled the data ?

Data at edge is often self-labeled

Text entry

Entered text is self-labeled. Eg. When user types messages — next character/word prediction

Image classification

Photo labels can be defined by natural user interaction with the app. e.g. which photos are likely to be viewed multiple times in future.

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

Federated Learning

Formal problem setting

m clients with fixed local dataset $\mathscr{P}_i$

Objective: $$\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x}) = \sum_{i=1}^{m} \frac{n_i}{n} F_i(\boldsymbol{x}) \quad (1)$$

$$F_i(\boldsymbol{x}) = \frac{1}{n_i} \sum_{\xi \sim \mathscr{P}_i} f(\xi; \boldsymbol{x}) \text{ where } |\mathscr{P}_i| = n_i$$

$f(\xi; \boldsymbol{x})$ is generally loss function i.e prediction loss for sample ξ made with model parameters $\boldsymbol{x}$

Centralised Setting: One node, one update step per round

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta g$$

Extension: Include more nodes

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta \sum_{i=1}^{m} \frac{n_i}{n} g_i \quad \text{where} \quad g_i = \nabla F_i(\boldsymbol{x}^{(t)})$$

This algorithm is called `FedSGD` (Federated SGD)

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

The FedAvg Algorithm

Design of `FedAvg`

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} - \eta \sum_{i=1}^{m} \frac{n_i}{n} g_i$$

Reformulating in the following way:

$\boldsymbol{x}_i^{(t+1)} = \boldsymbol{x}_i^{(t)} - \eta g_i$ (On i^{th} client)

$\boldsymbol{x}^{(t+1)} = \sum_{i=1}^{m} \frac{n_i}{n} \boldsymbol{x}_i^{(t+1)}$ (Server)

$$\boldsymbol{x}_i^{(t,k+1)} = \boldsymbol{x}_i^{(t,k)} - \eta g_i^k \quad \text{for} \quad k \in \{0,1,\ldots,\tau_i - 1\}$$

Second index in (t, k) refers to the local step on client i which performs τ_i local steps

$\boldsymbol{x}^{(t+1,0)} = \sum_{i=1}^{m} \frac{n_i}{n} \boldsymbol{x}_i^{(t,\tau_i)}$ E Epochs, B: $\tau_i = \frac{n_i}{B} E$

This is the popular `FedAvg` (Federated Averaging) algorithm.

Another Detail: Select C fraction out of m clients

(Empirical results show diminishing returns)

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

The FedAvg Algorithm

Pseudo Code

Server executes

Initialise $\boldsymbol{x}^{(0,0)}$

For each round $t = 1,2,\ldots$ **do**

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

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

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

$\boldsymbol{x}_k^{(t,\tau_k)} \leftarrow \text{ClientUpdate}(k, \boldsymbol{x}^{(t,0)})$

$$\boldsymbol{x}^{(t+1,0)} = \sum_{k=1}^{K} \frac{n_k}{n} \boldsymbol{x}_k^{(t,\tau_k)}$$

Parameters

local mini-batch size B	Number of local epochs E
Client fraction C	Learning rate η

ClientUpdate($k, \boldsymbol{x}$): // Run on client k

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

For each local epoch e from 1 to E **do**

For each $b \in \mathscr{B}$ **do**

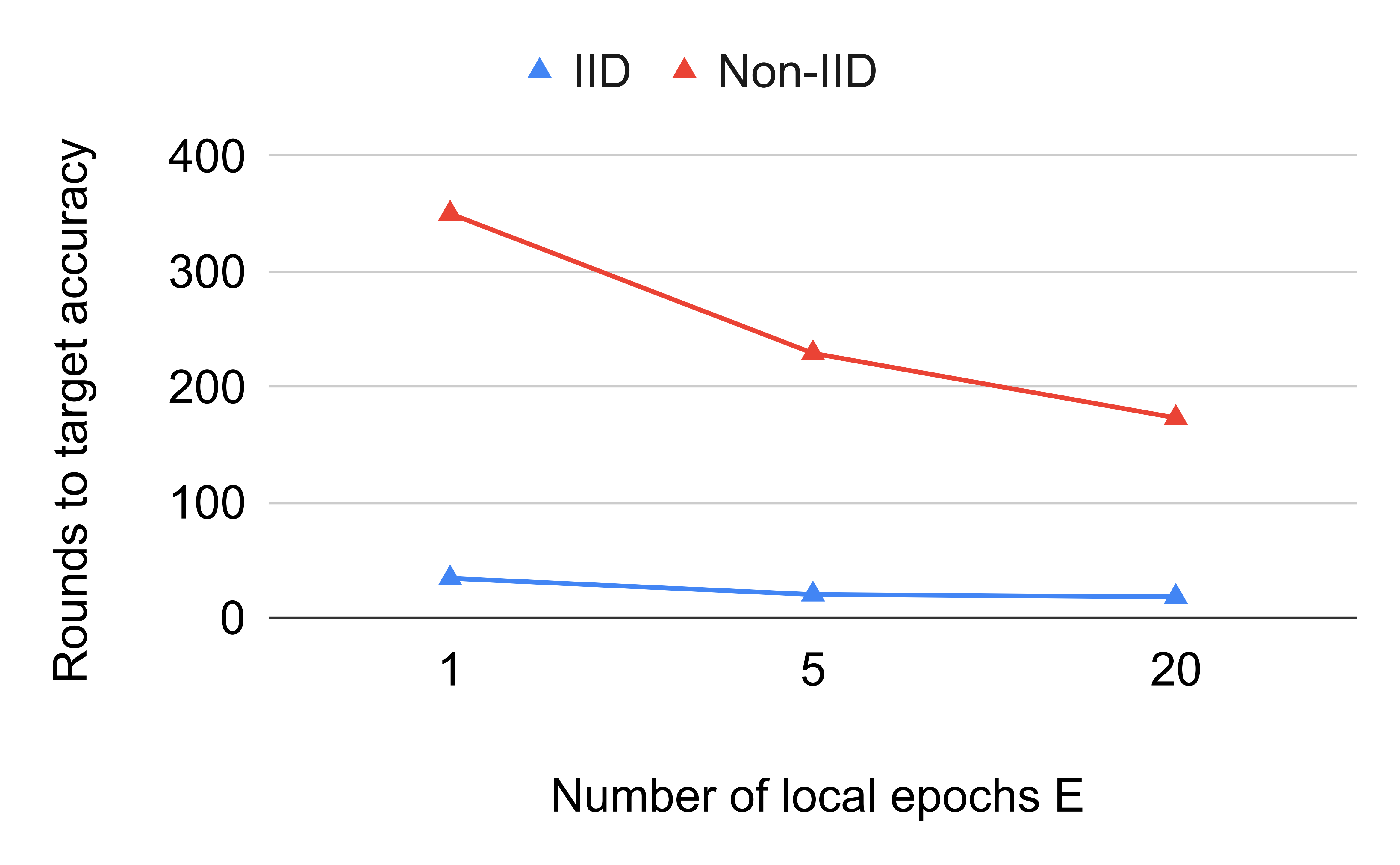
$\boldsymbol{x} \leftarrow \boldsymbol{x} - \eta \nabla f(\boldsymbol{x}; b)$

return $\boldsymbol{x}$ to the server

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

Some results

FedAvg algorithm



MNIST dataset

Target accuracy — 99%
Batch size — 10
Model — CNN with 2 convolution layers
m = 100
C = 0.1

FedAvg is communication-efficient. Why ?

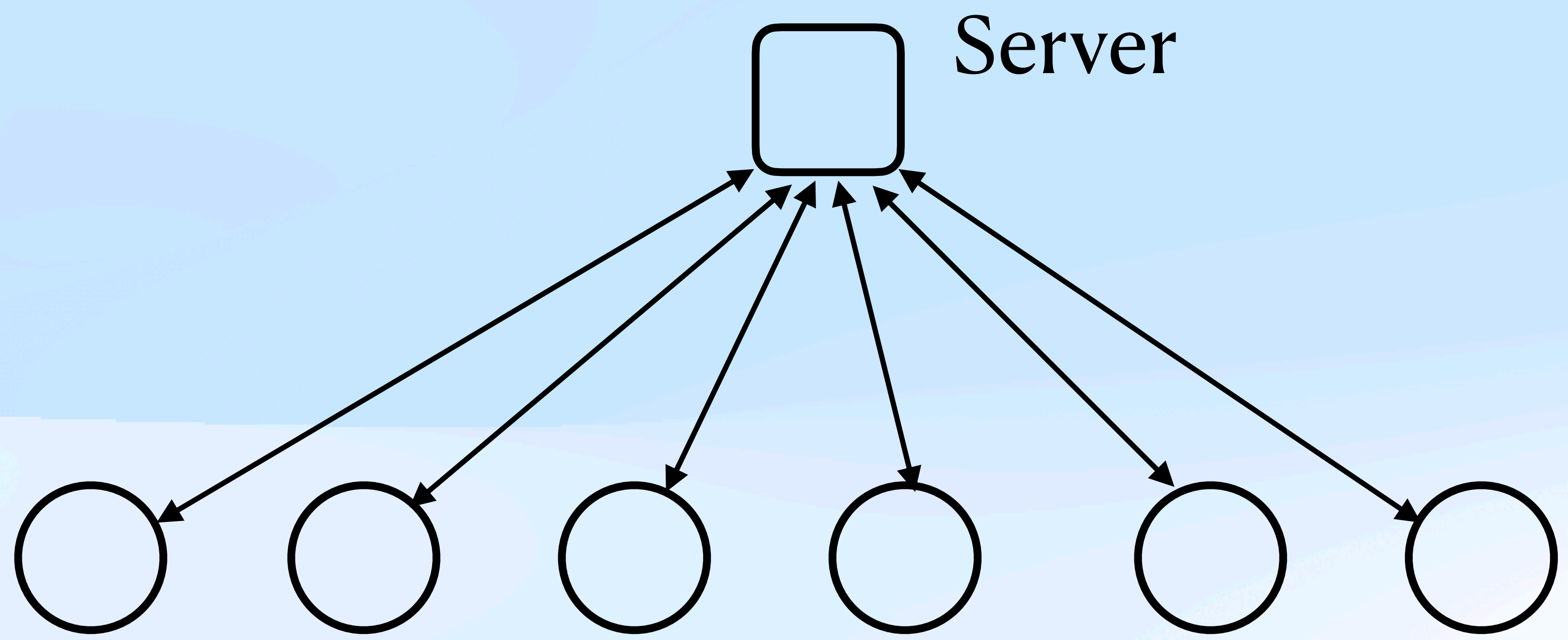
- Each round entails significant communication costs
- Multiple local updates reduce total rounds to convergence and save communication

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

Decentralized Learning (DL)

Introduction

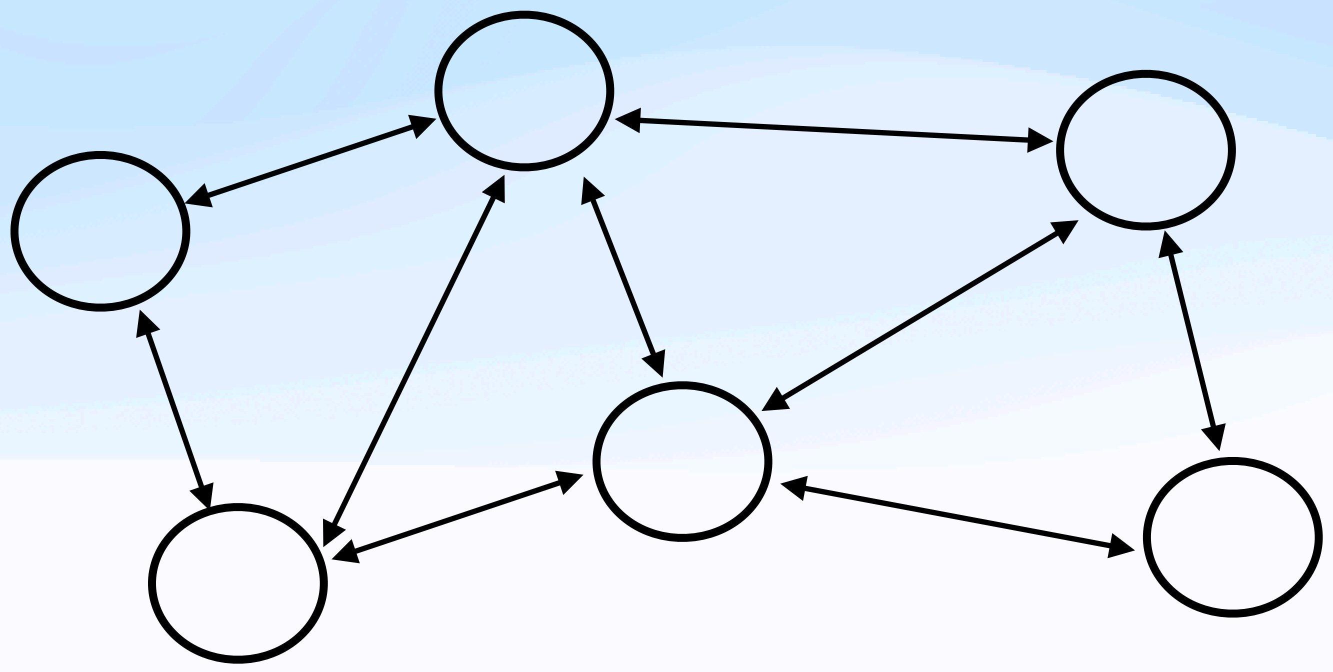
Federated Learning (FL) [1]



Workers

$$\boldsymbol{x}^{(t+1,0)} = \sum_{i=1}^{m} \frac{n_i}{n} \boldsymbol{x}_i^{(t,\tau_i)}$$

Decentralized Learning (DL) [2]



Workers

$$\boldsymbol{x}_i^{(t+1,0)} = \sum_{j \in N_i \cup \{i\}} \boldsymbol{W}_{ji} \boldsymbol{x}_j^{(t,\tau_j)}$$

[1] Brendan McMahan, Edier Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273-1282. PMLR, 2017.

[2] Lian, Xiangru, et al. "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent." Advances in neural information processing systems 30 (2017).

Decentralized Learning (DL)

Introduction

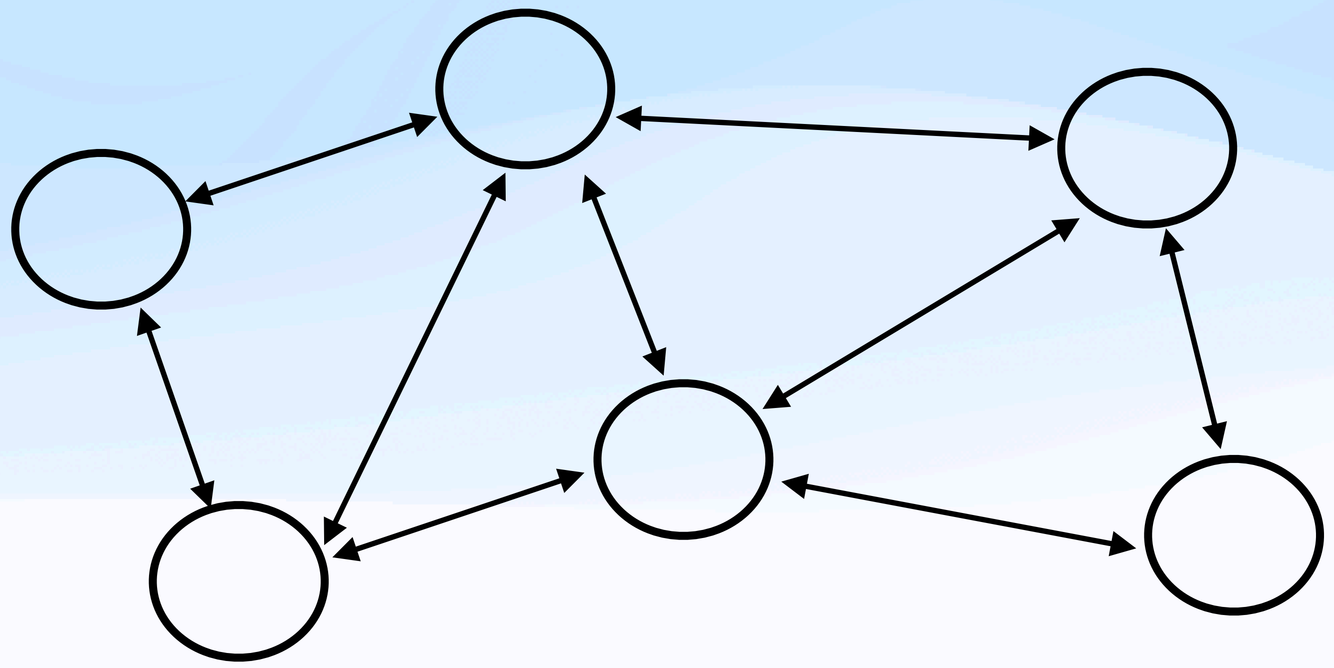
Benefits of DL

- **Scalability**: Remove bandwidth bottleneck on server
- **Privacy**: Remove a central monitoring point
- **Fault-tolerance**: Remove the need for a highly available server for coordination

Potential drawbacks

- **(Possible) Lower Convergence Speed**: Higher variance between individual models may slow down convergence
- **Topology affects convergence**

Decentralized Learning (DL)



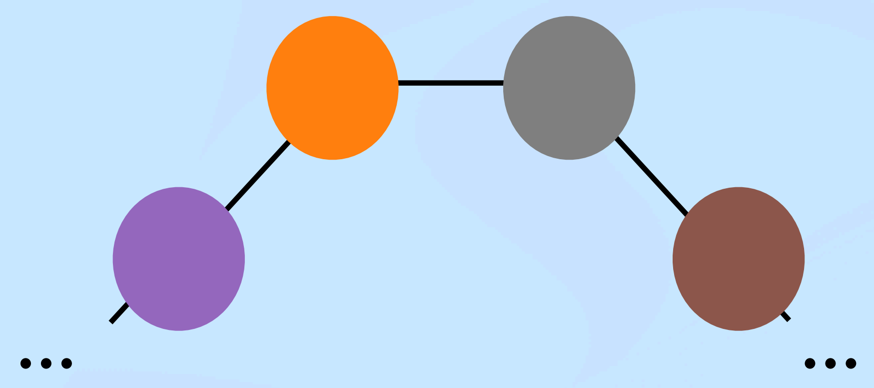
Workers

$$\boldsymbol{x}_i^{(t+1,0)} = \sum_{j \in N_i \cup \{i\}} \boldsymbol{W}_{ji} \boldsymbol{x}_j^{(t,\tau_j)}$$

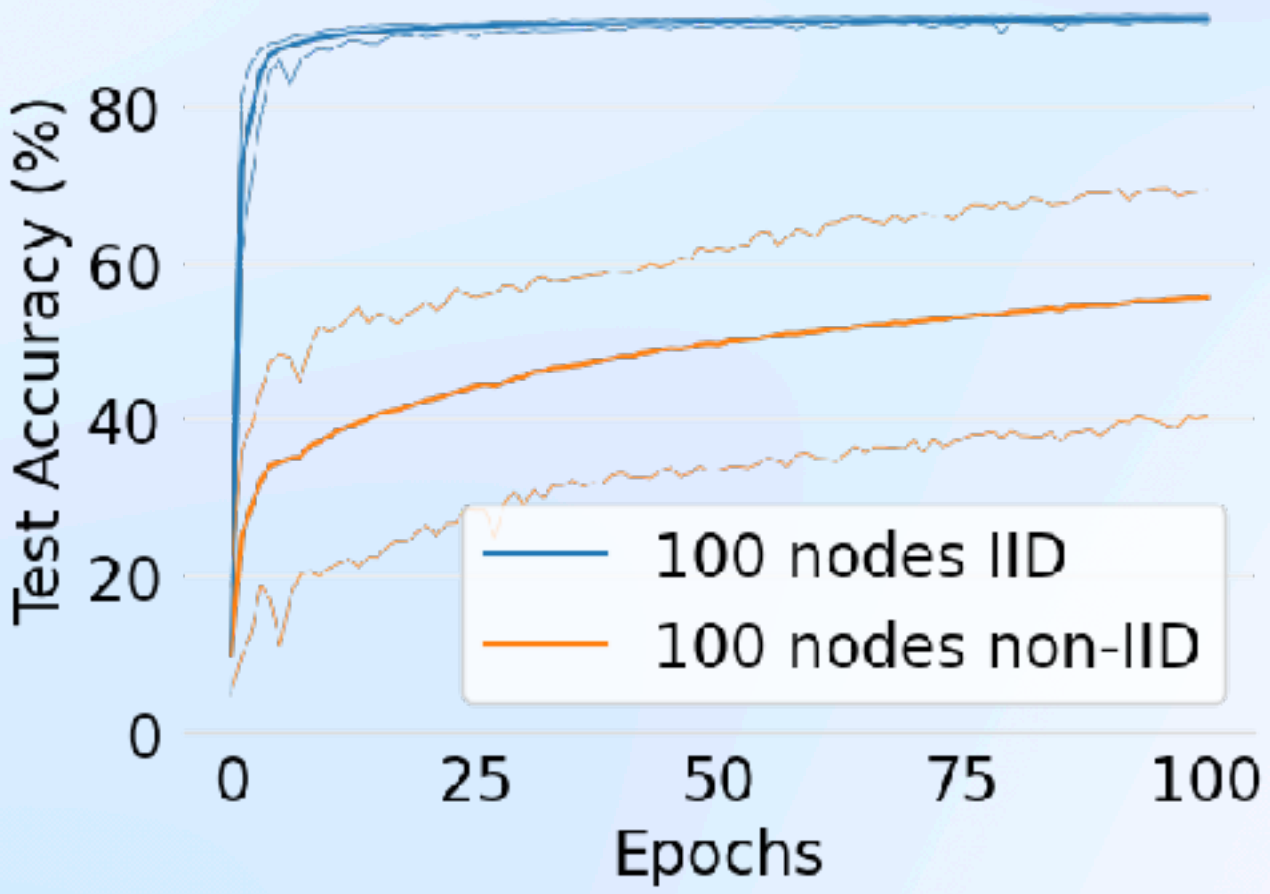
Some results: Decentralized-SGD

MNIST with linear classifier

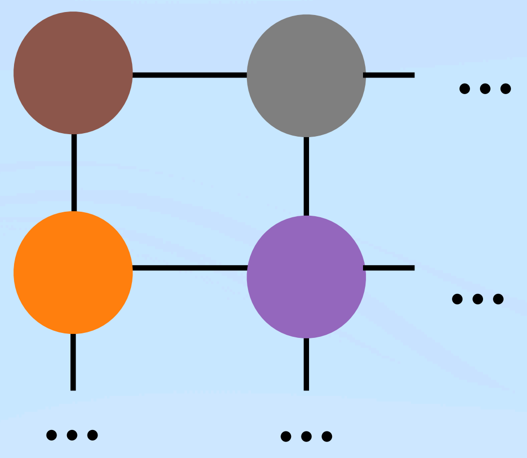
Ring



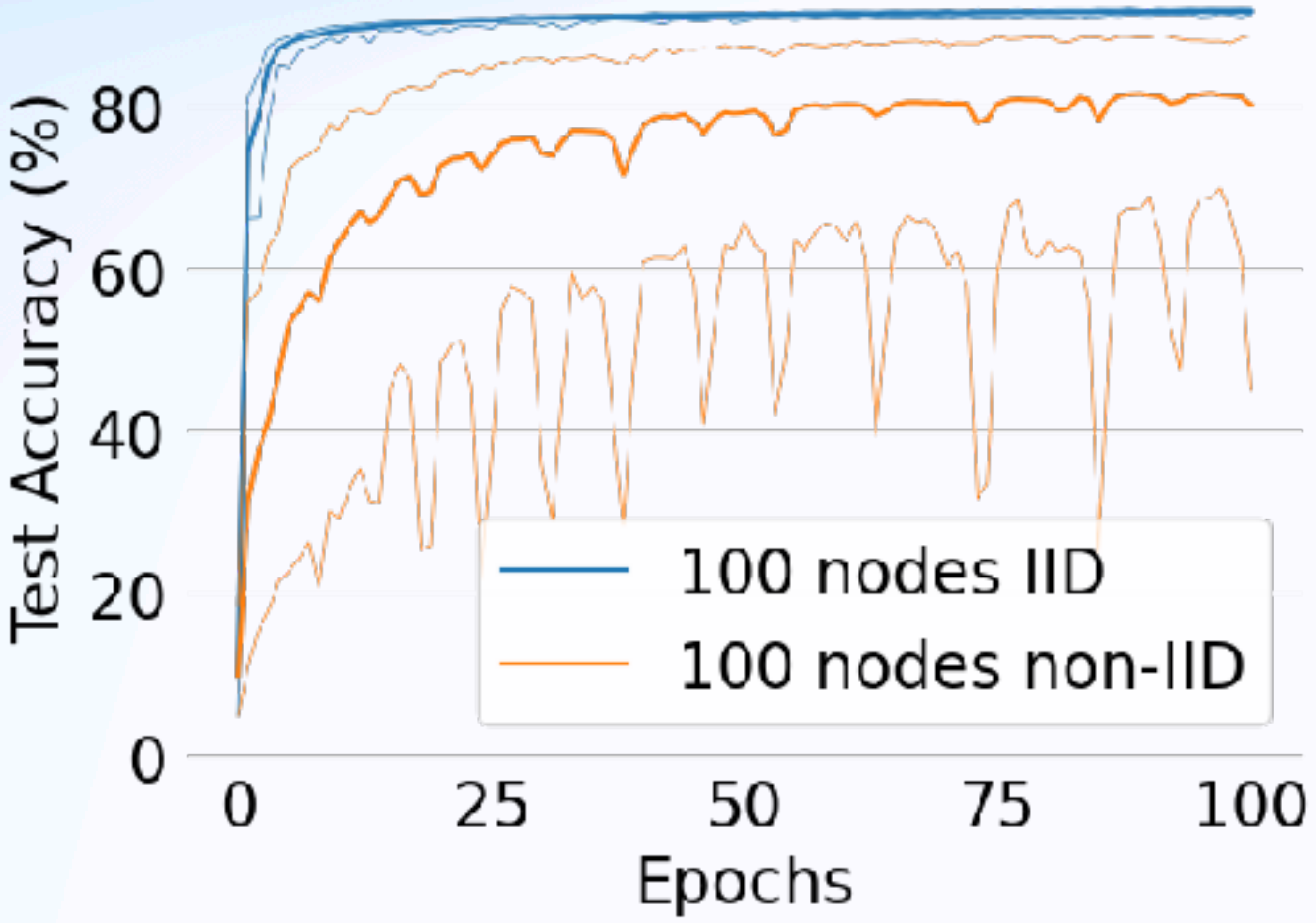
Minimal symmetric topology



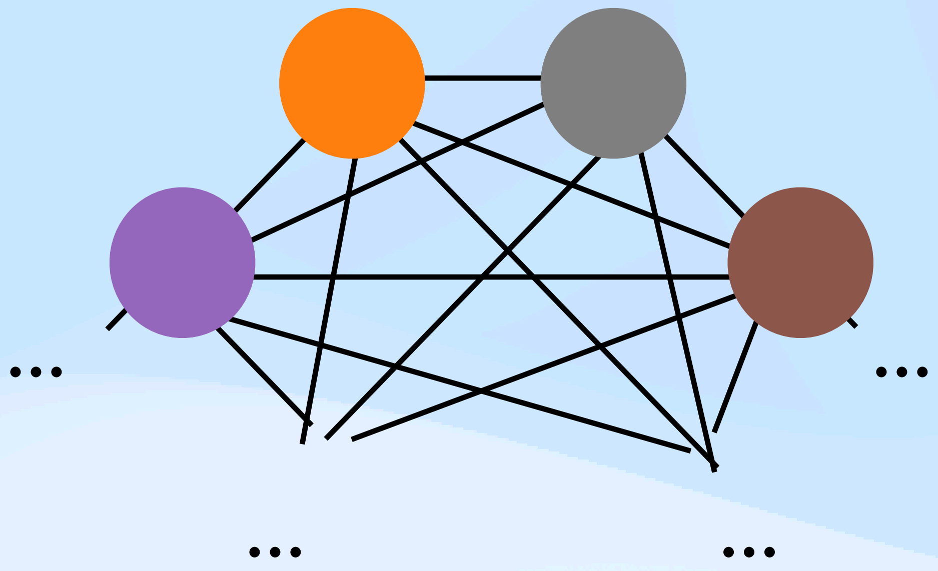
Grid



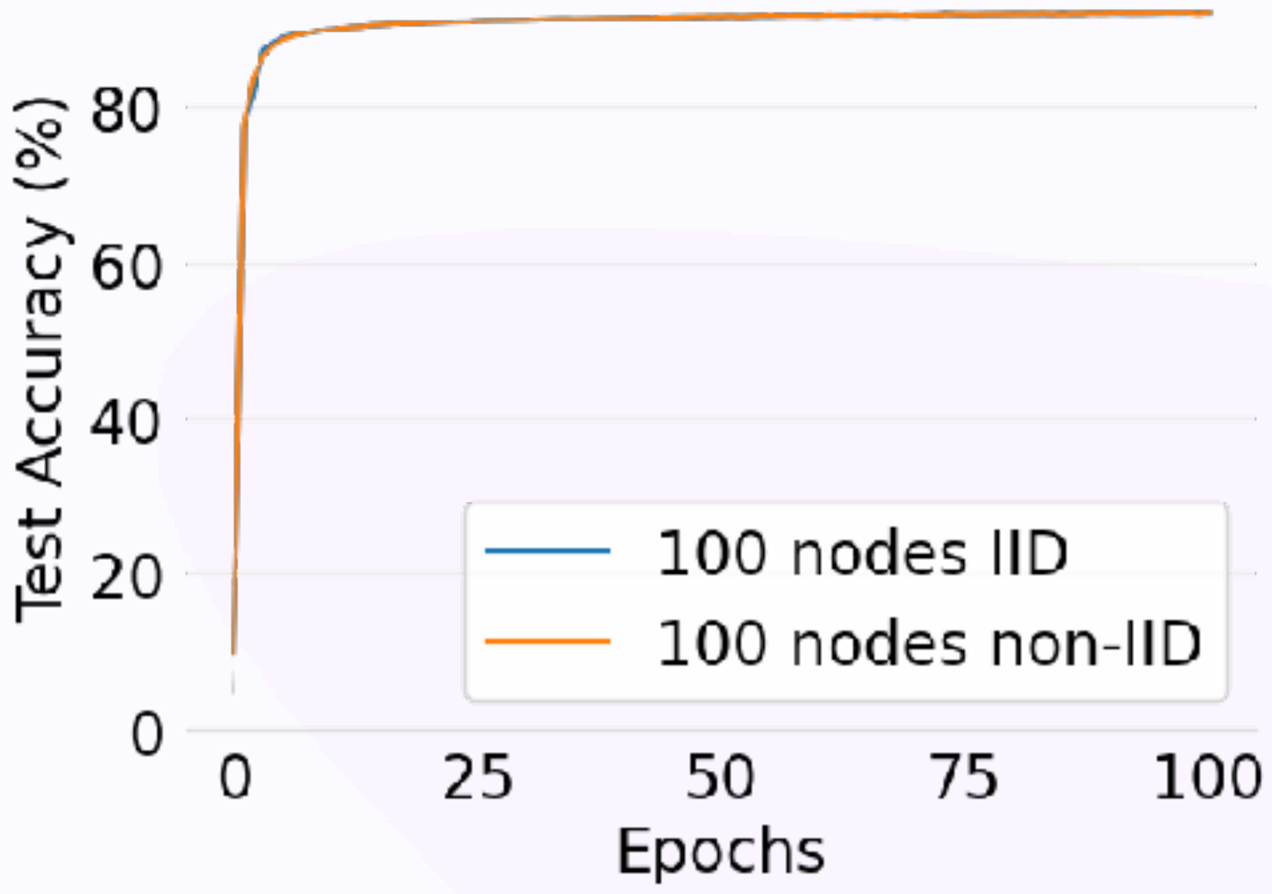
Intermediate



Fully-Connected



Maximally connected topology

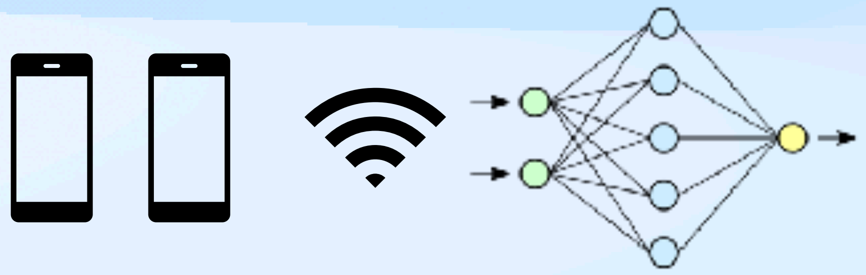


Results from — [1] Bellet, Aurélien, Anne-Marie Kermarrec, and Erick Lavoie. "D-cliques: Compensating for data heterogeneity with topology in decentralized federated learning." 2022 41st International Symposium on Reliable Distributed Systems (SRDS). IEEE, 2022.

Problems with FL and DL

These approaches have issues

Expensive Communication	Data Heterogeneity	Systems Heterogeneity	
Low end participating devices must upload/ download deep neural network models	Local data distributions of clients could be arbitrarily different from global distributions.	Clients differ in their processor, memory, network capabilities, etc.	. . . .



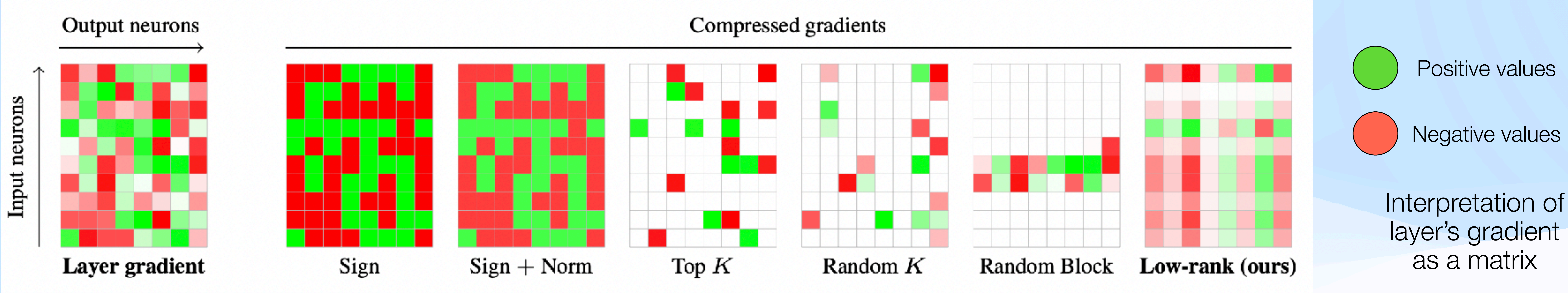
....

Data Center

High bandwidth links connecting clusters in data centres	IID data fully available for training.	All nodes are similarly equipped.

How to reduce communication ?

Some ideas



Interpretation of layer's gradient as a matrix

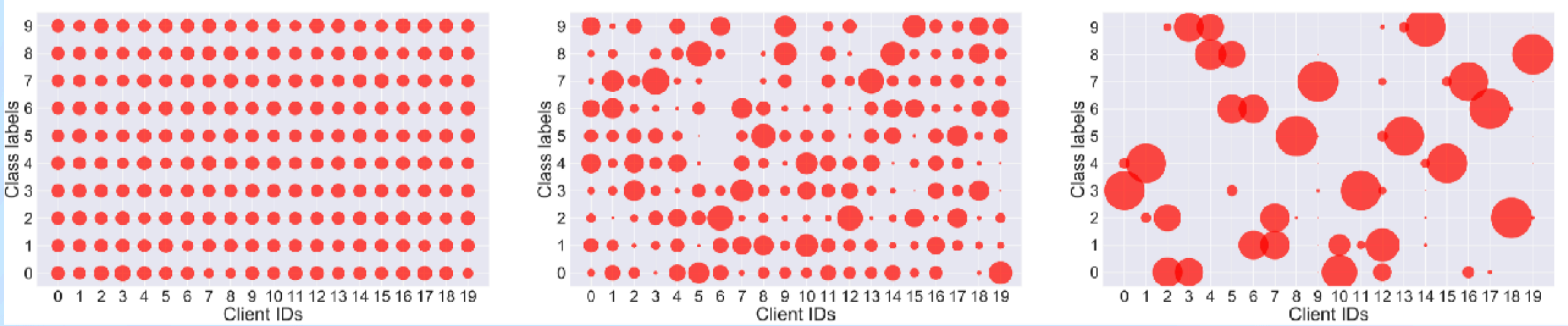
- Do multiple local updates (FedAvg algorithm)
- Model/gradient compression using
  - Quantization (to 1-bit) — i.e use sign — $32\times$ reduction
  - Top k = 1 % of the entries — $100\times$ reduction
  - Low rank approximation [1]
- One-shot Federated Learning [2]

Figure credits — [1] Vogels, Thijs, Sai Praneeth Karimireddy, and Martin Jaggi. "PowerSGD: Practical low-rank gradient compression for distributed optimization." In *NeurIPS 2019*.

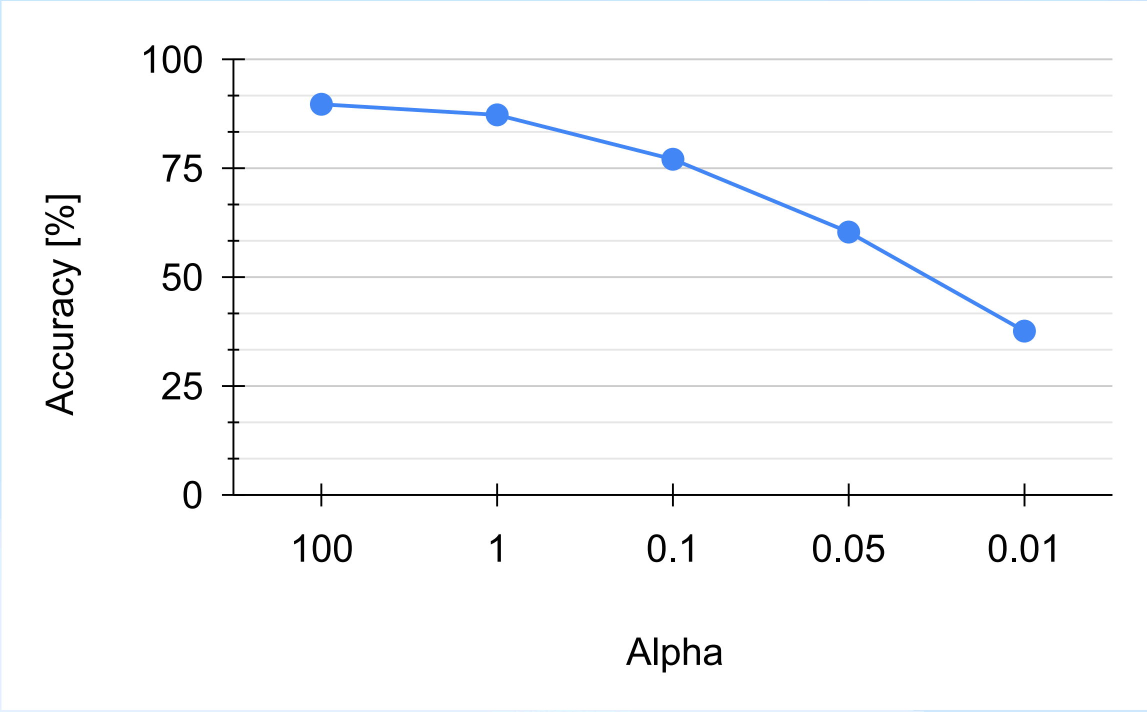
[2] Allouah, Youssef, Akash Dhasade, Rachid Guerraoui, Nirupam Gupta, Anne-Marie Kermarrec, Rafael Pinot, Rafael Pires, and Rishi Sharma. "Revisiting Ensembling in One-Shot Federated Learning." In *NeurIPS 2024*.

How to address data heterogeneity ?

Some methods



$\alpha = 100$ $\qquad$ $\alpha = 1$ $\qquad$ $\alpha = 0.01$



FedAvg performance deteriorates

Advanced algorithms

- FedProx [2] — Modifies client local loss with the proximal term

$$F_i(\boldsymbol{x}^{(t,k)}) + \frac{\mu}{2}||\boldsymbol{x}^{(t,0)} - \boldsymbol{x}^{(t,k)}||$$

- Scaffold [3] — Client drift correction
- Federated Adaptive Optimization [4] — FedAdam, FedAdagrad

Heterogeneity simulated Dirichlet distribution $\alpha \in (0,\infty)$

Lower $\alpha \rightarrow$ higher heterogeneity

α tending to $\infty \rightarrow$ IID data

Figure from [1] Lin, Tao, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. "Ensemble distillation for robust model fusion in federated learning." In *NeurIPS 2020.*

[2] Li, Tian, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. "Federated optimization in heterogeneous networks." In *MLSys 2020.*

[3] Karimireddy, Sai Praneeth, et al. "Scaffold: Stochastic controlled averaging for federated learning." *International conference on machine learning*. PMLR, 2020.

[4] Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., ... & McMahan, H. B. Adaptive Federated Optimization. In *ICLR 2021*.

How to address systems heterogeneity ?

Some ideas

- Let each client use a different quantisation of the same model: 2-bit, 3-bit, 8-bit, etc. depending on the system speed [1]
- Let each client use a different model suitable to its hardware [2]
  - Then how do we aggregate at the server ? New aggregation schemes ?
- Let clients submit model updates even after the reporting deadline [3]
  - Aggregate while accounting for the staleness factor — Asynchronous FL
- Clever participation selection — choose clients that allow both good and fast learning [4]
  - Does it introduce bias ?

---

[1] Abdelmoniem, Ahmed M., and Marco Canini. "Towards mitigating device heterogeneity in federated learning via adaptive model quantization." In Proceedings of the 1st Workshop on Machine Learning and Systems, pp. 96-103. 2021.

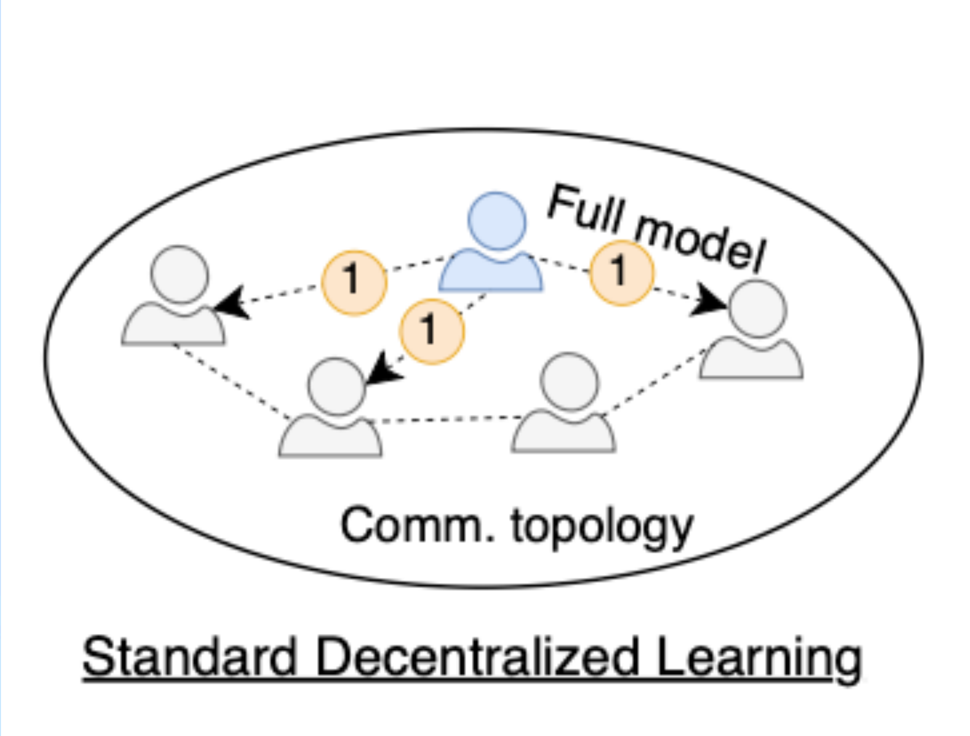
[2] Diao, Enmao, Jie Ding, and Vahid Tarokh. "HeteroFL: Computation and communication efficient federated learning for heterogeneous clients." arXiv preprint arXiv:2010.01264 (2020)

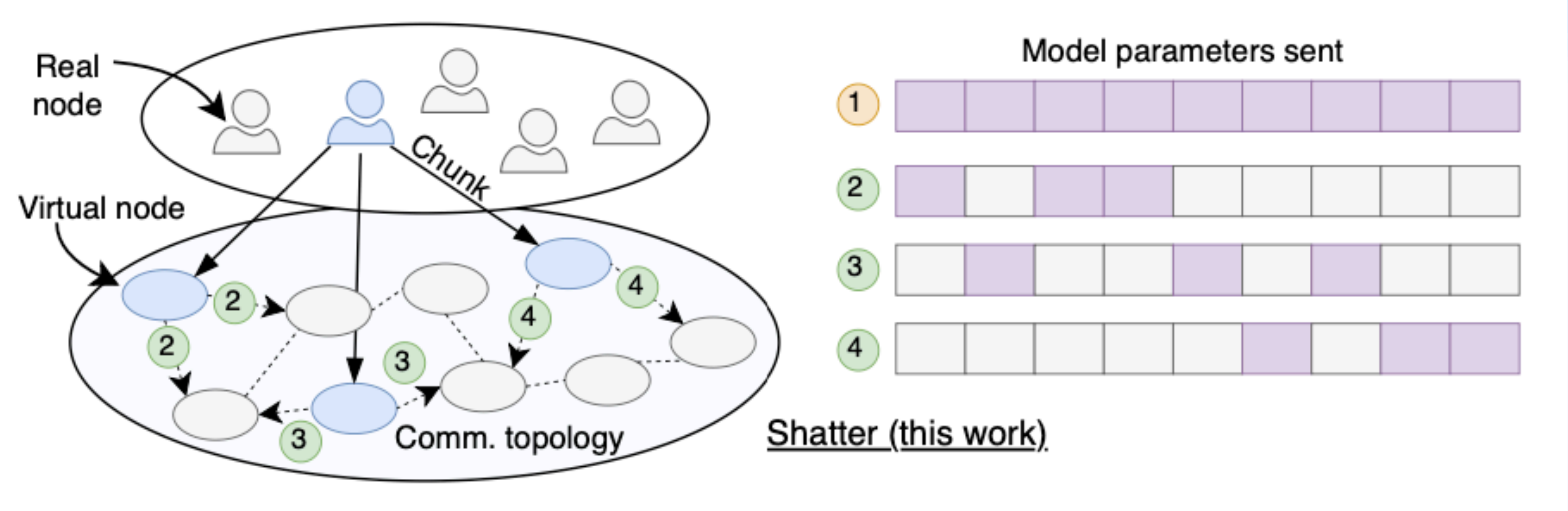
[3] Huba, Dzmitry, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu et al. "Papaya: Practical, private, and scalable federated learning." Proceedings of Machine Learning and Systems 4 (2022): 814-832

[4] Lai, Fan, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. "Oort: Efficient Federated Learning via Guided Participant Selection." In OSDI, pp. 19-35. 2021.

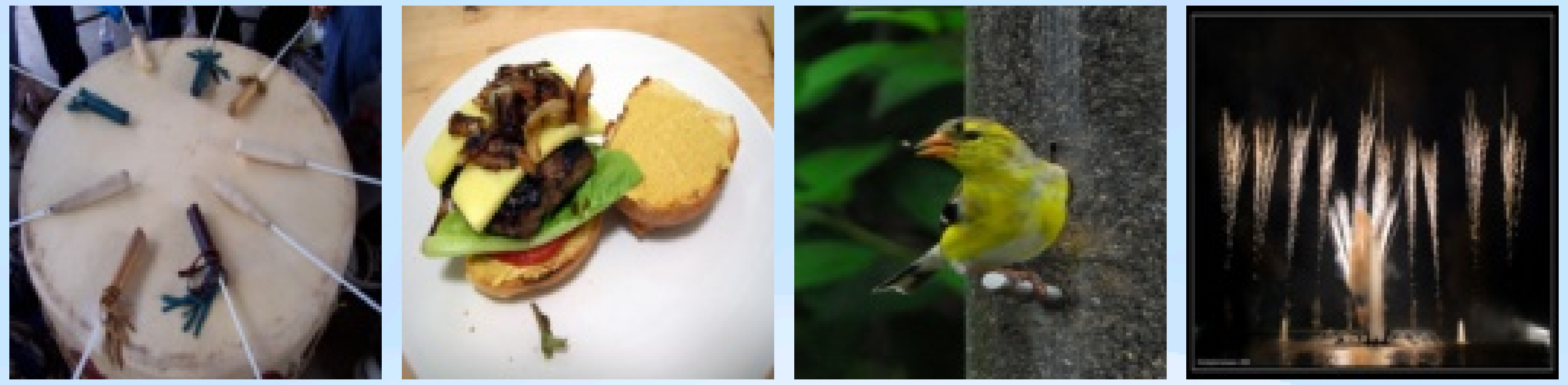
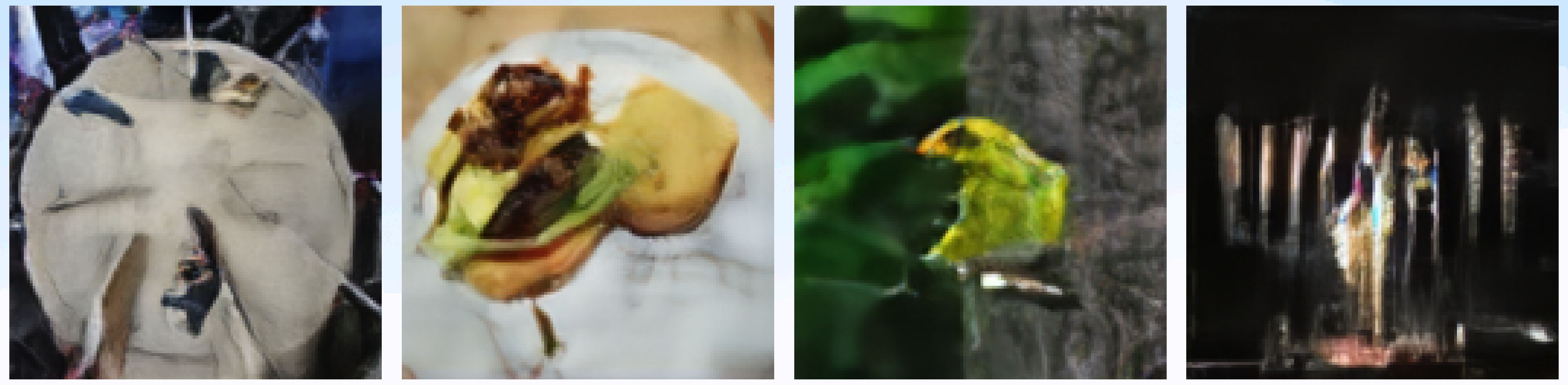
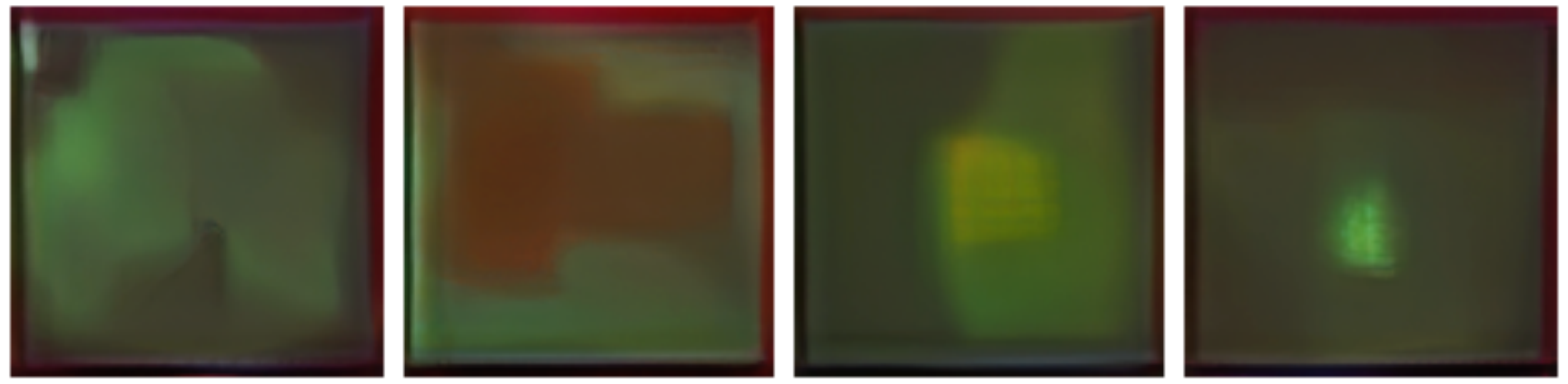
On the privacy of gradients

Gradient Inversion Attack





Shatter

Approach	Images	LPIPS Score
Original		-
Standard DL		0.266
Random 1/8 chunk		0.781

LPIPS: lower means more similar

[1] Biswas, Sayan, Mathieu Even, Anne-Marie Kermarrec, Laurent Massoulié, Rafael Pires, Rishi Sharma, and Martijn de Vos. "Noiseless Privacy-Preserving Decentralized Learning." *Proceedings on Privacy Enhancing Technologies* (2025).