Federated Communication-Efficient Multi-Objective Optimization

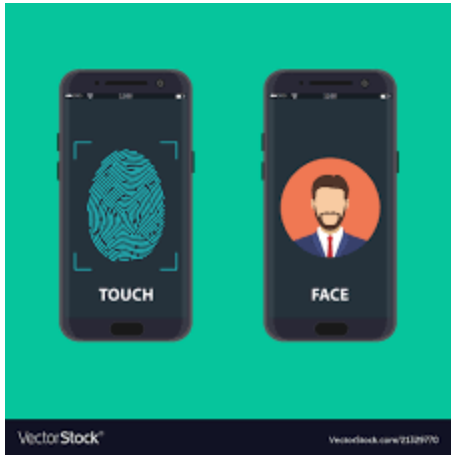
Pranay (pranaysh@iitb.ac.in)
C-MInDS, IIT Bombay

Paper: AISTATS'25 (arxiv.org/pdf/2410.16398)

Work with

Baris Askin, Dr. Gauri Joshi and Dr. Carlee Joe-Wong (ECE, CMU)

Big Data



- Need lots of data

Data is Naturally Distributed!

- Edge-devices collect data

- This data is used to train ML models

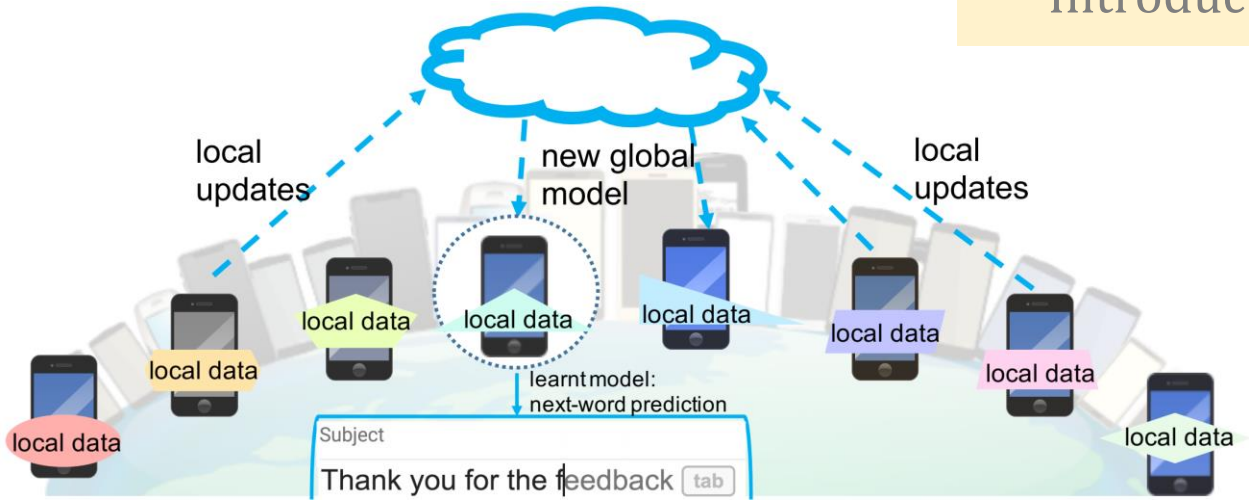
Liu, et al. "Privacy-preserving traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

Liu, et al. "Privacy-preserving traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

Image Classification

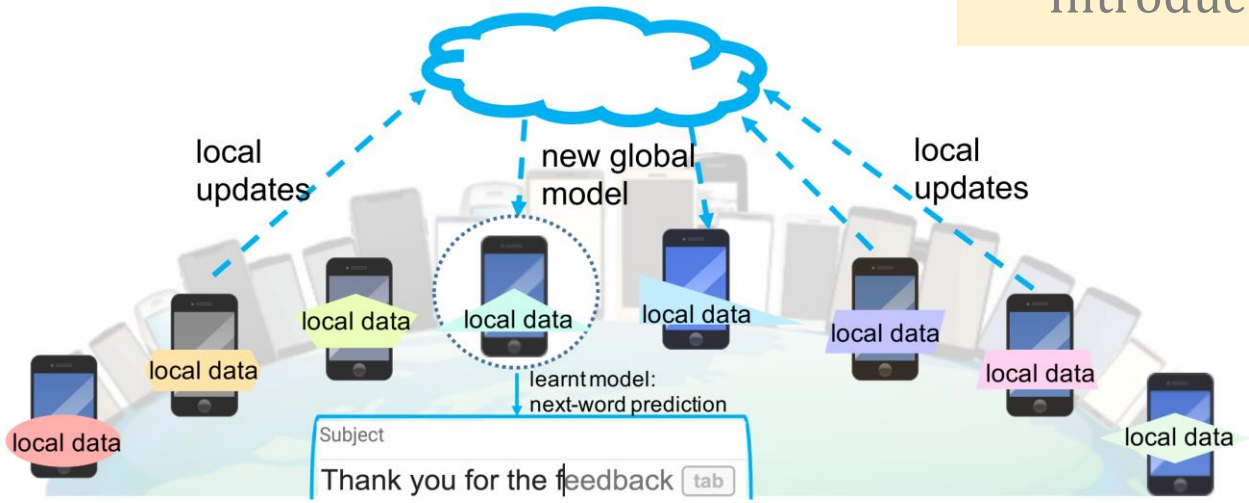
Liu, et al. "Privacy-preserving traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

Image Classification



Liu, et al. "Privacy-preserving traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

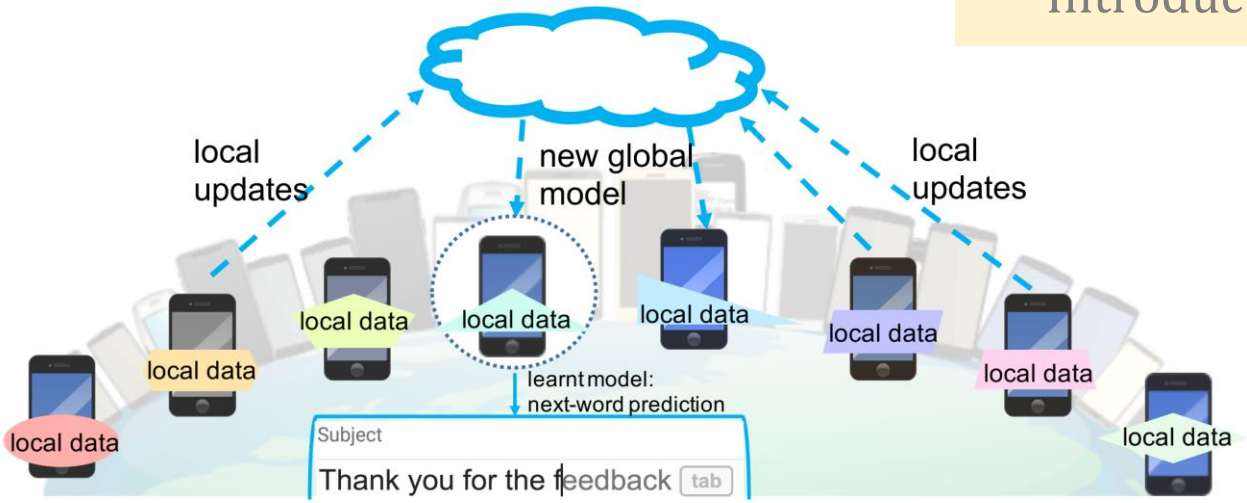
Image Classification



Next-word Prediction

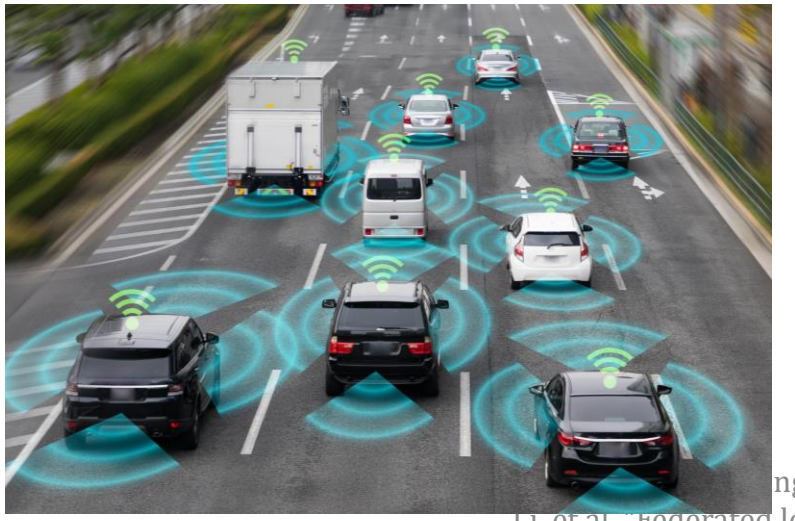
Liu, et al. "Privacy-preserving traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

Image Classification



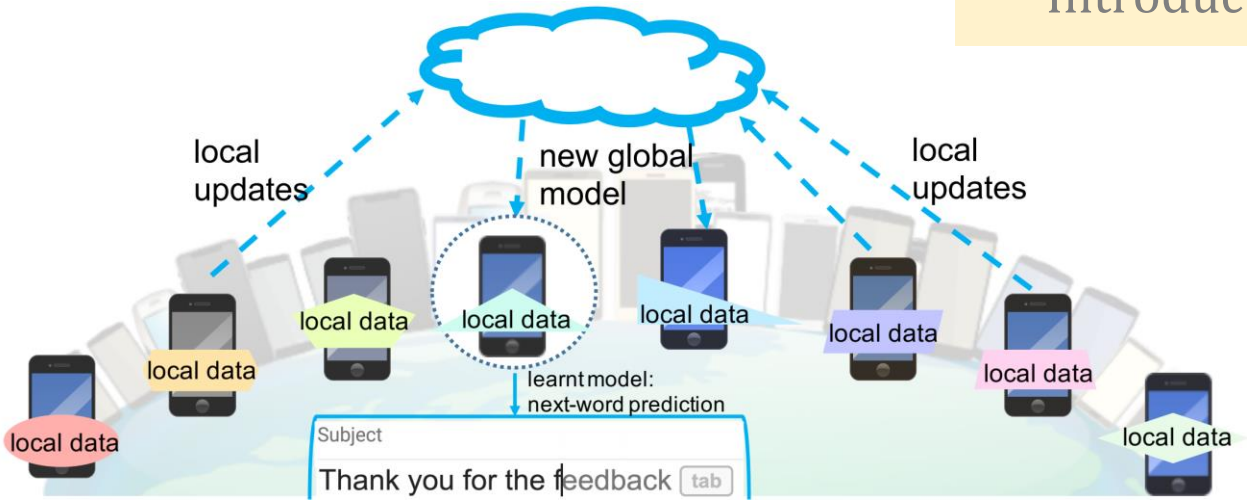
Next-word Prediction

Autonomous Driving



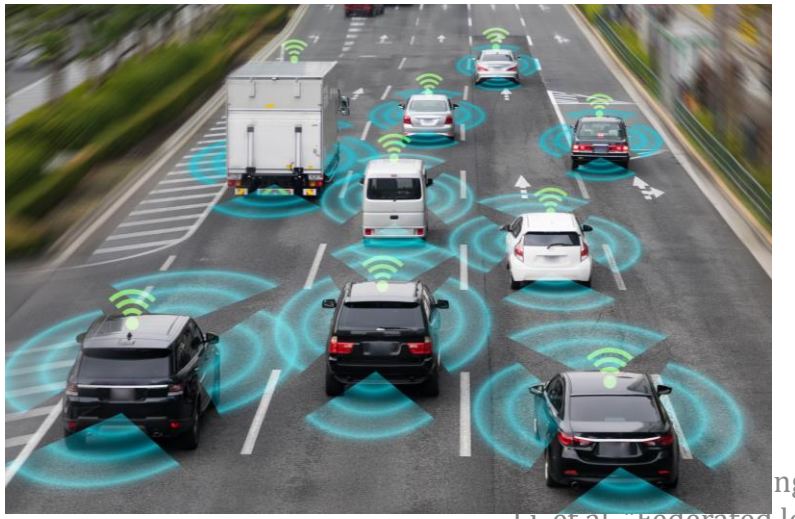
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

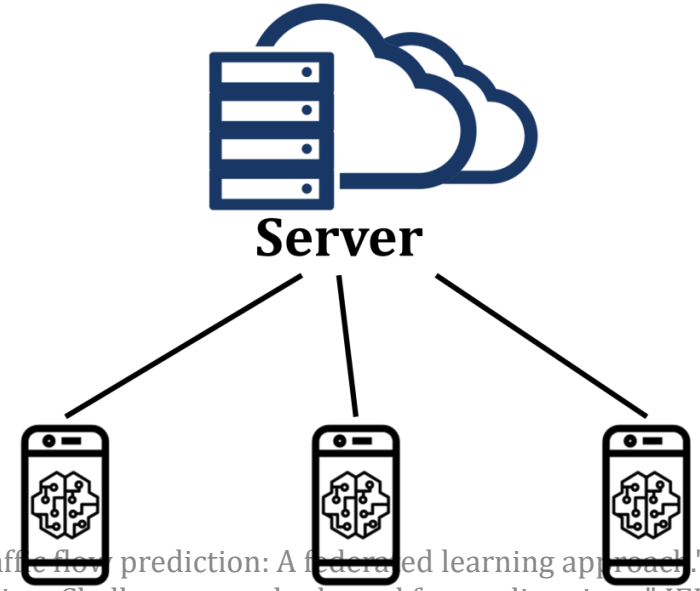
Image Classification



Next-word Prediction

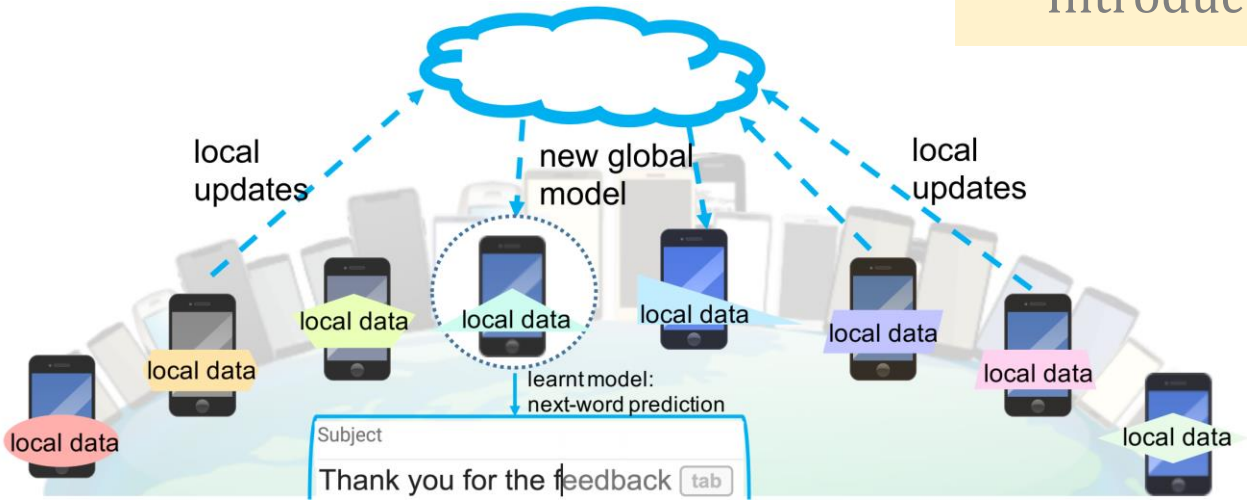
Autonomous Driving





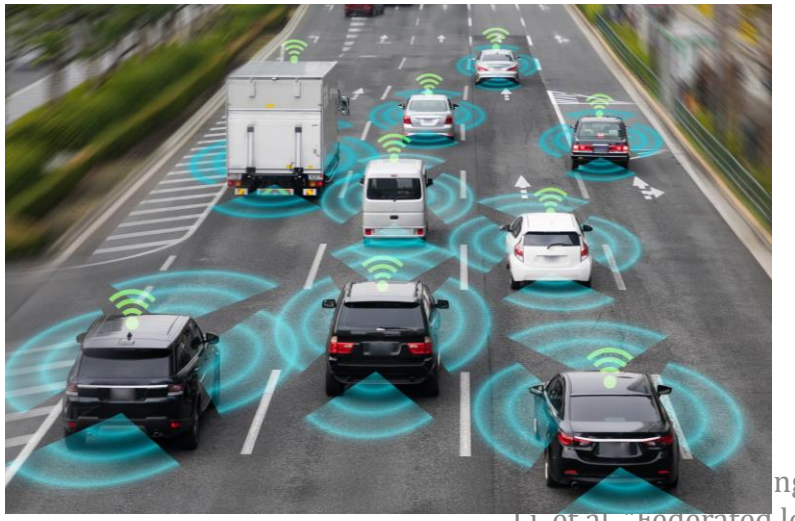
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

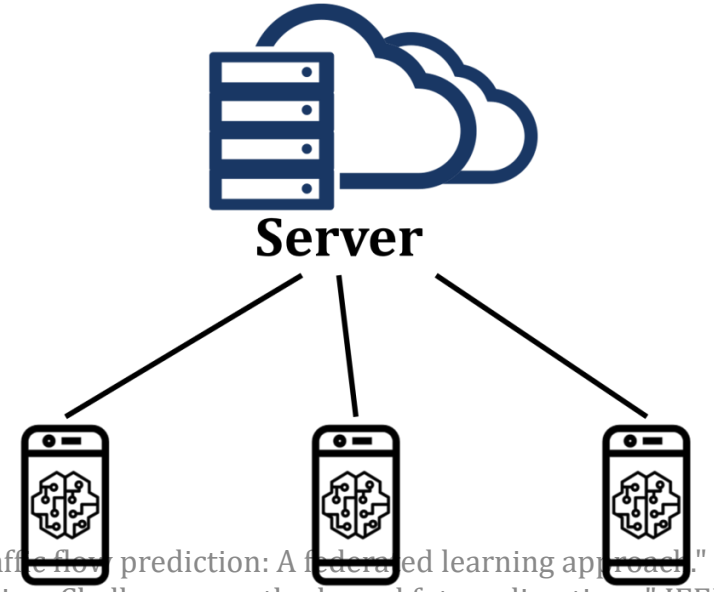
Image Classification



Next-word Prediction

Autonomous Driving

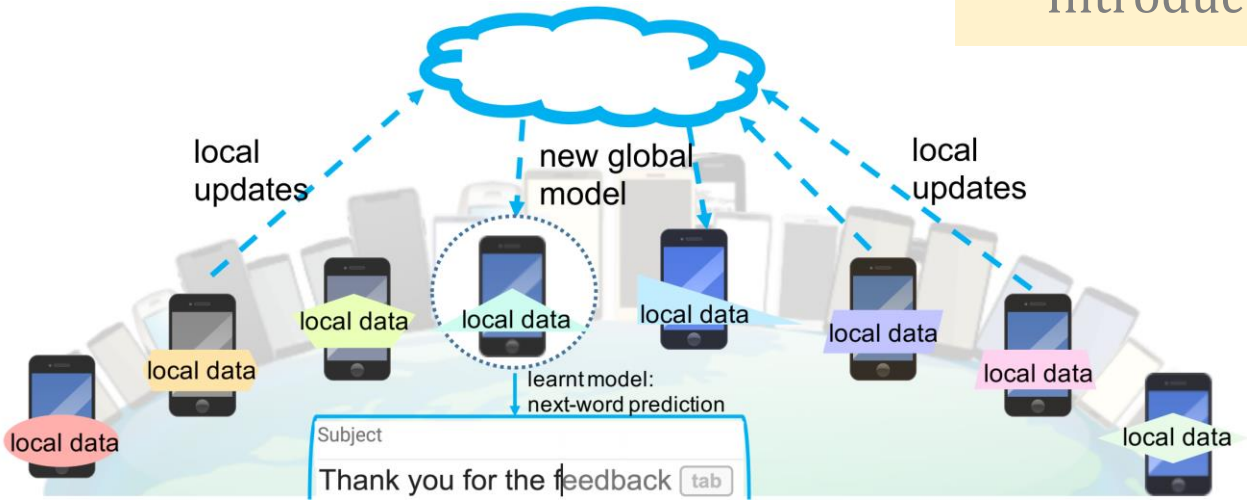




- Communication
  - Device ↔ Server
  - Device ↔ Device

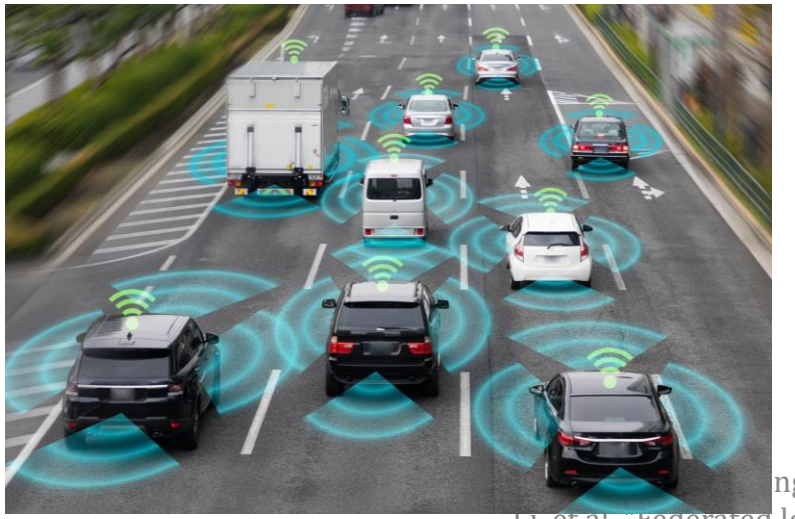
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

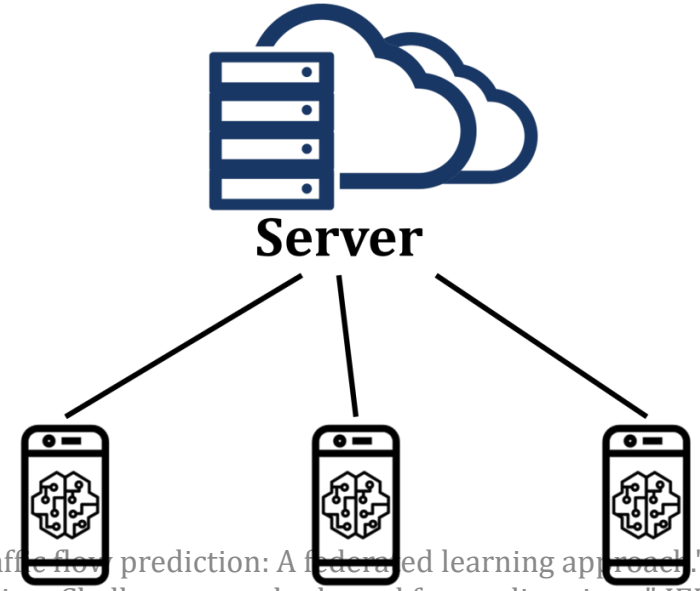
Image Classification



Next-word Prediction

Autonomous Driving

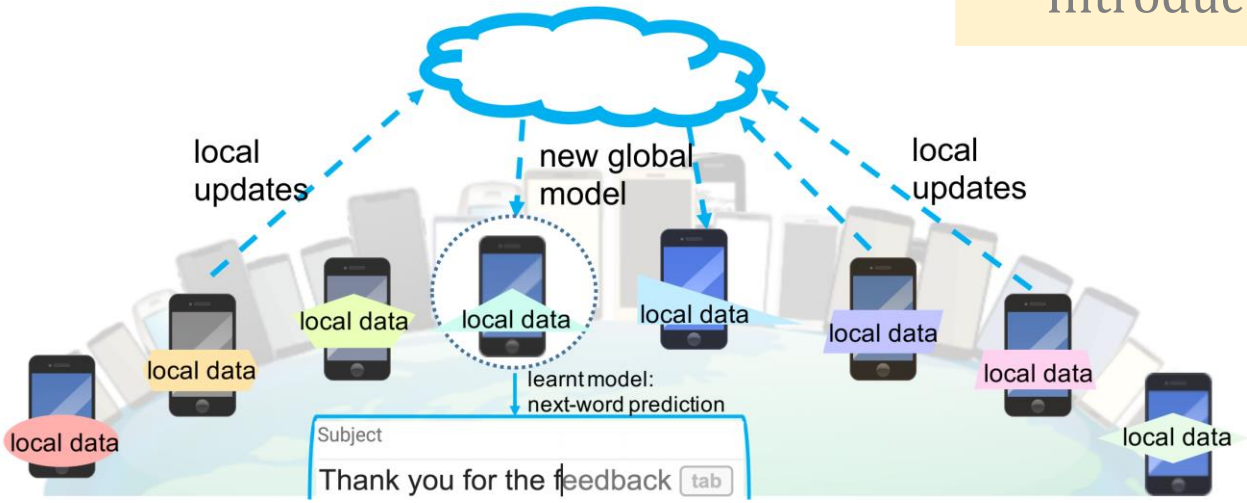




- Communication
  - Device ↔ Server
  - Device ↔ Device

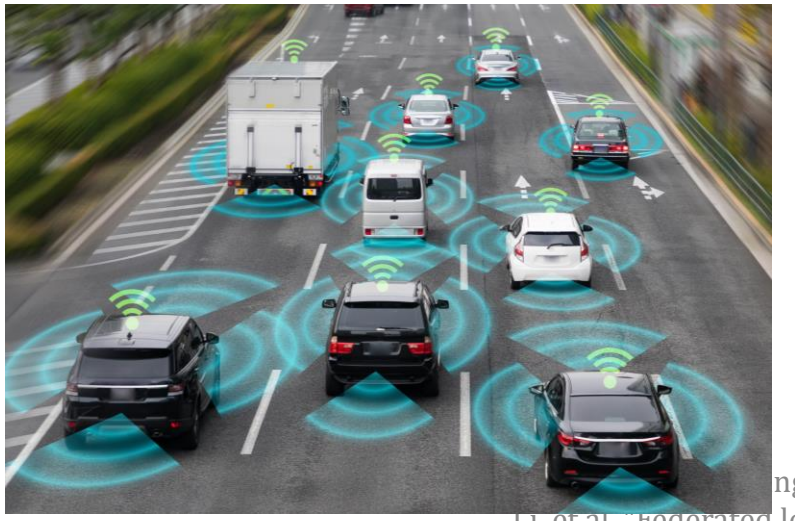
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

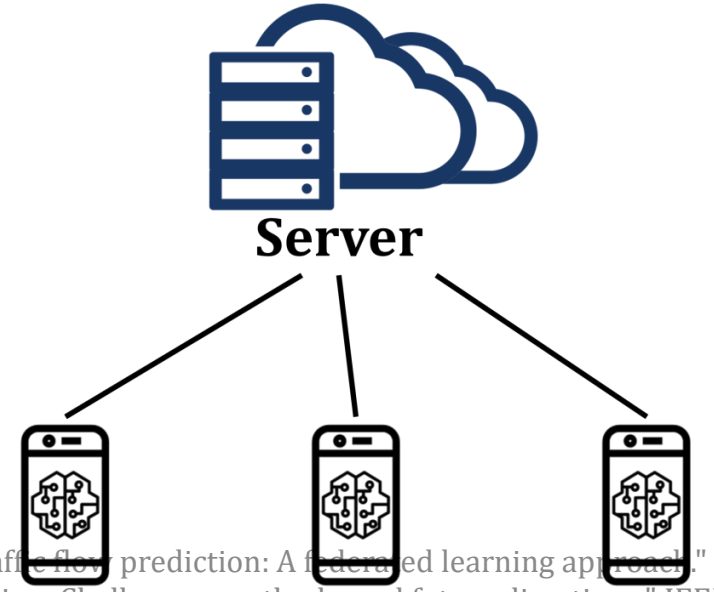
Image Classification



Next-word Prediction

Autonomous Driving

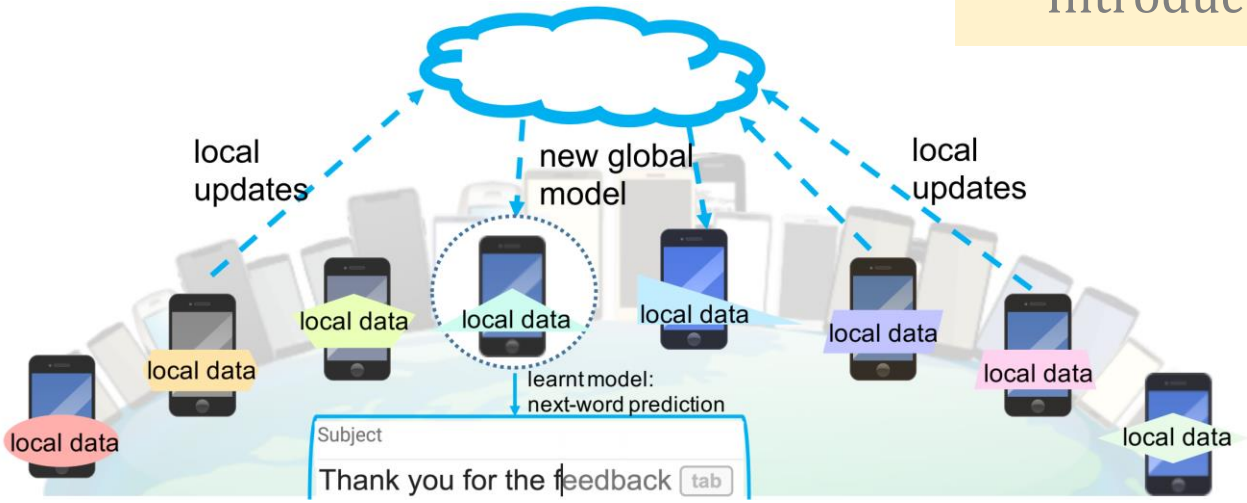




- Communication
  - Device ↔ Server
  - Device ↔ Device

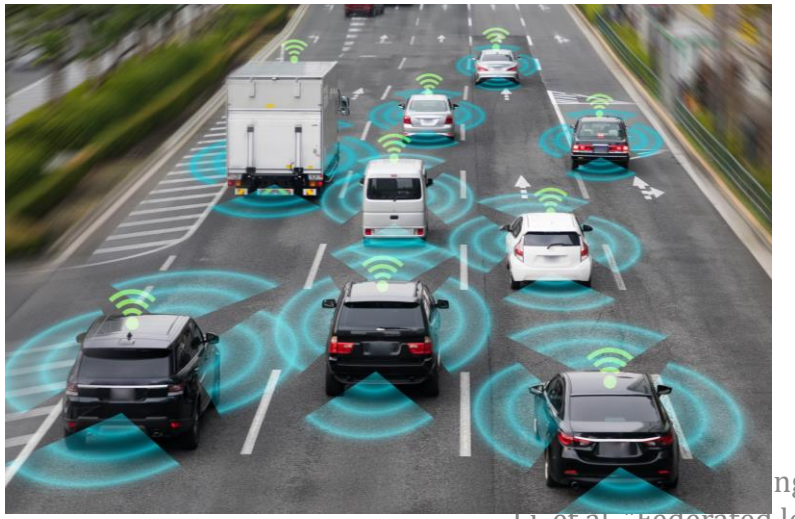
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

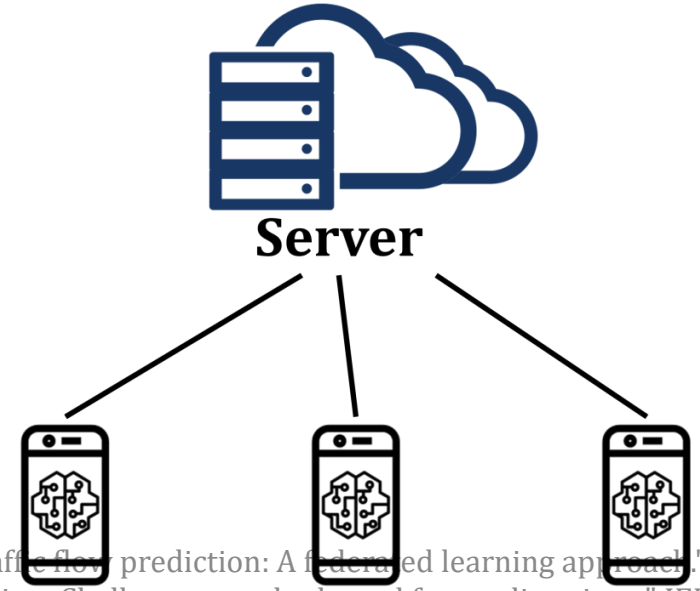
Image Classification



Next-word Prediction

Autonomous Driving

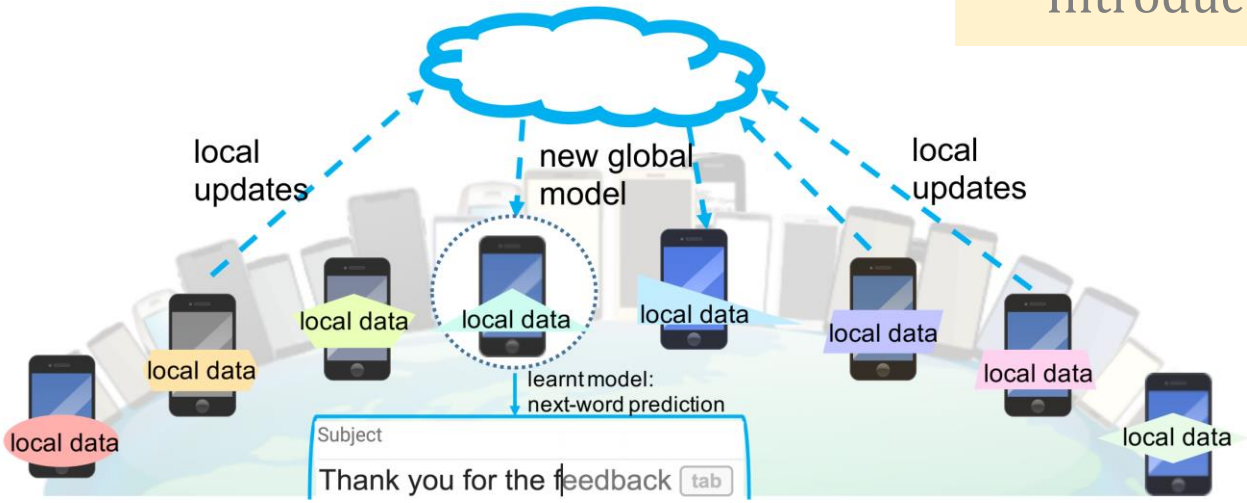




- Communication
  - Device ↔ Server
  - Device ↔ Device
- "Device-server" setting

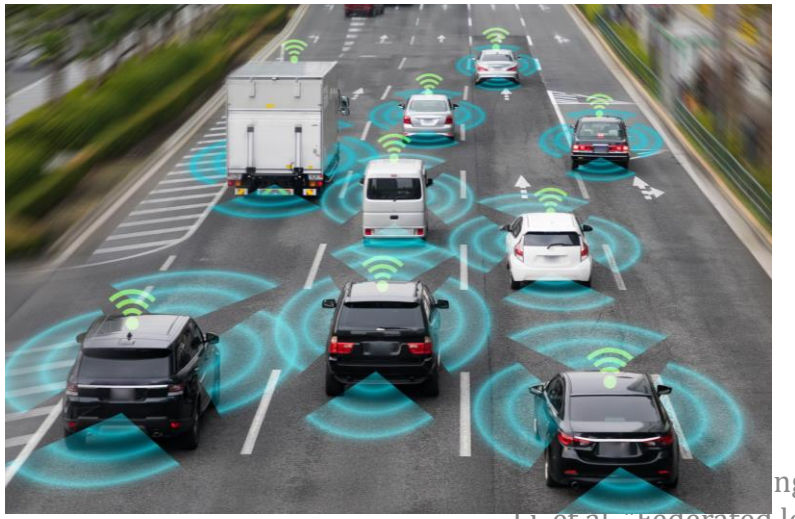
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

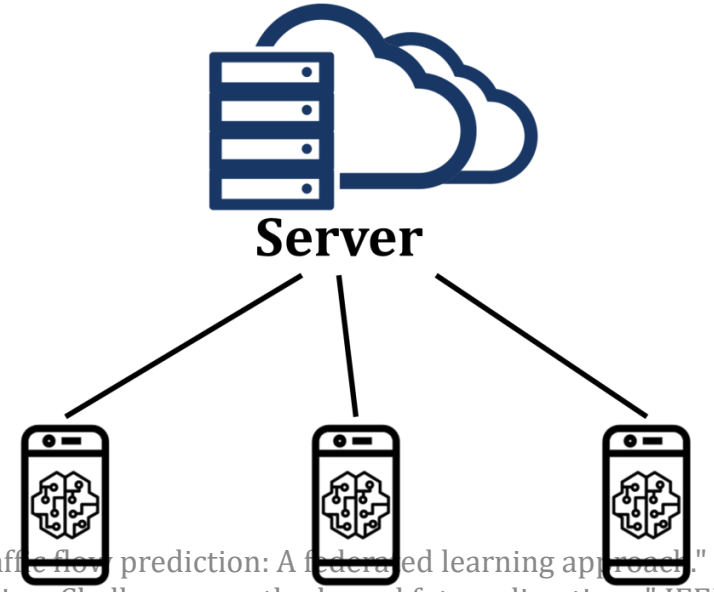
Image Classification



Next-word Prediction

Autonomous Driving



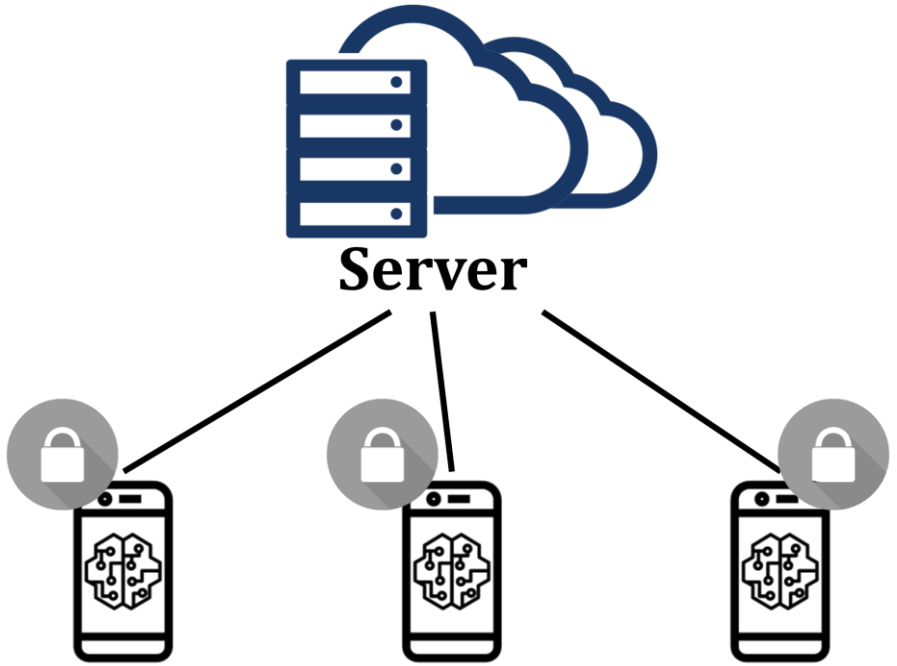


- Communication
  - Device ↔ Server
  - Device ↔ Device
- "Device-server" setting
- Challenges?

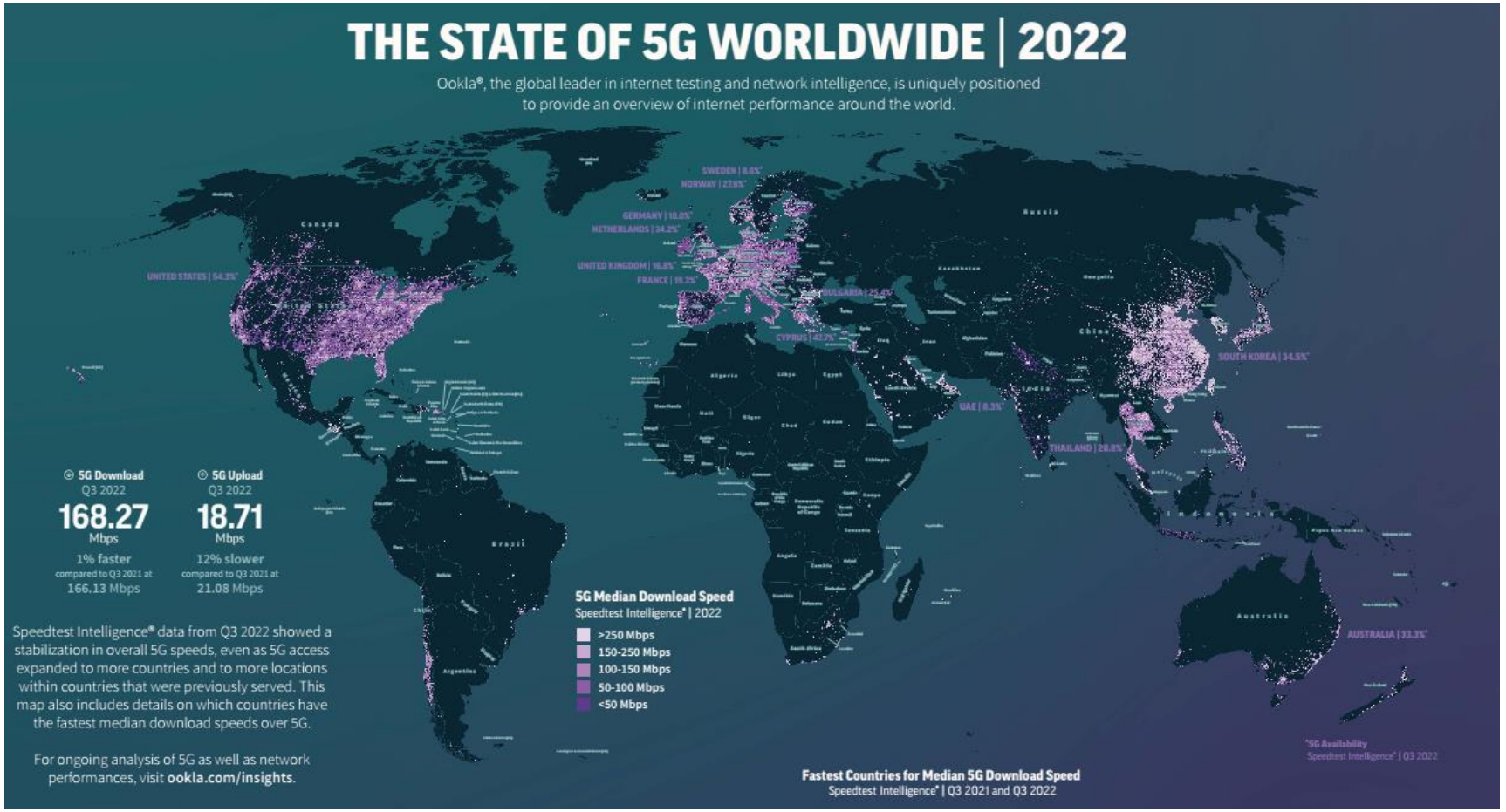
ng traffic flow prediction: A federated learning approach." IEEE IoT Journal 2020.
Li, et al. "Federated learning: Challenges, methods, and future directions." IEEE SPMag, 2020.
Hard, et al. "Federated learning for mobile keyboard prediction." arXiv:1811.03604.
Image curtsey: https://envuetelematics.com/

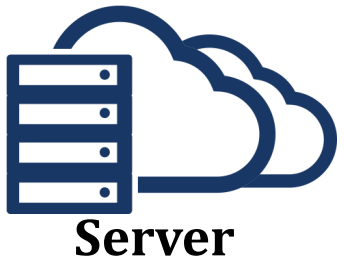
Privacy

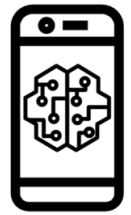
Individual user data should not be leaked



Network constraints







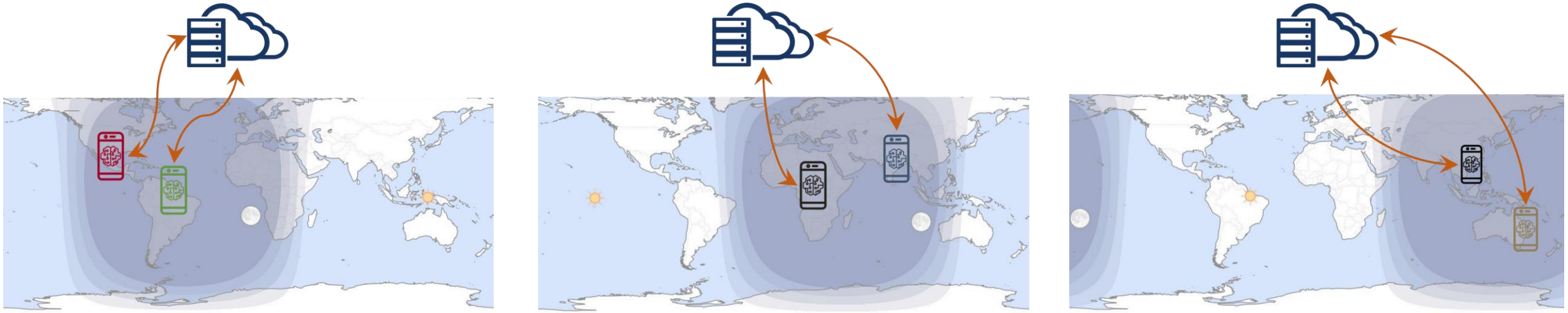
Communication is expensive

Image from www.ookla.com

Intermittent Availability

Devices are available when

- Idle, plugged-in and on wifi



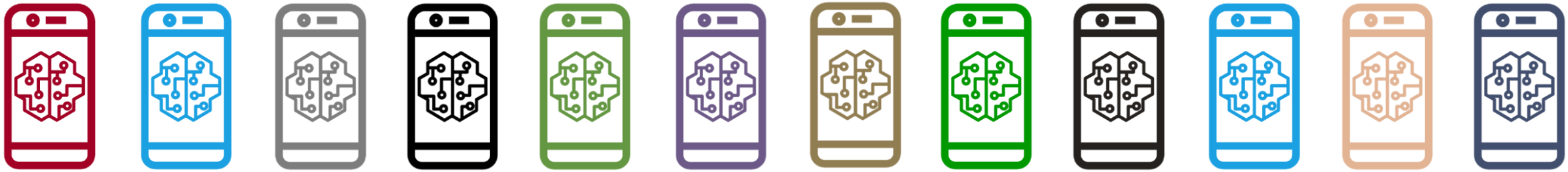
Distributed learning under

- Privacy concerns
  Devices don’t send raw data*
- Network constraints
  Infrequent communication with server
- Intermittent client availability

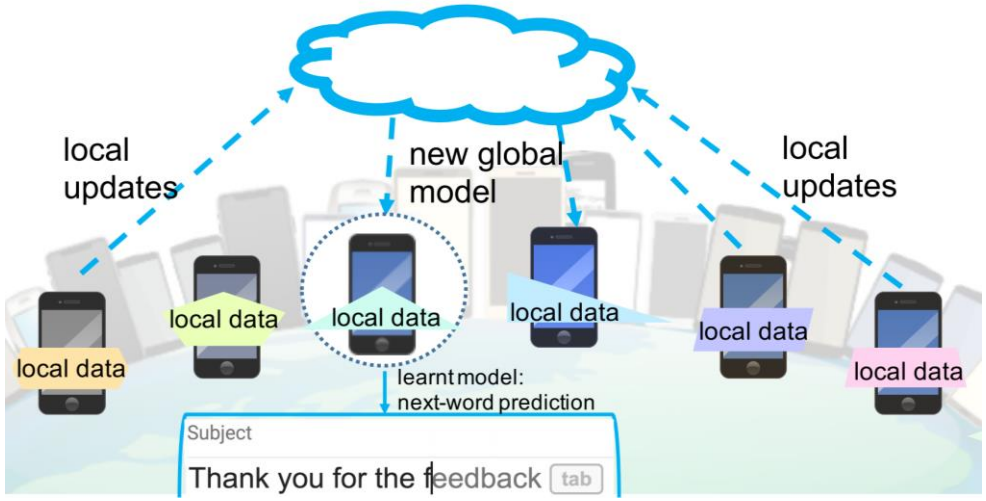
  (partial participation)

Federated Learning (FL)

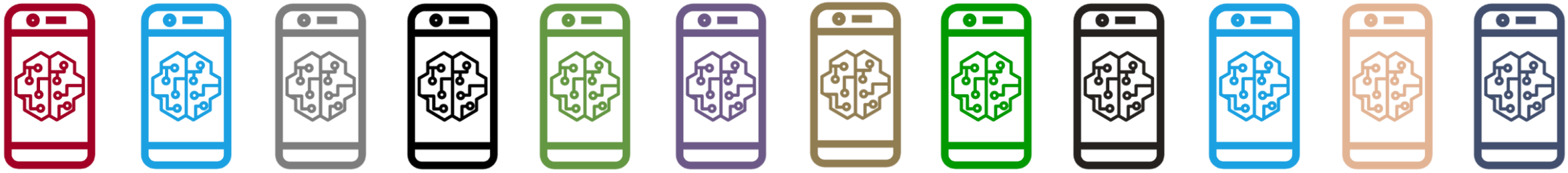
Server



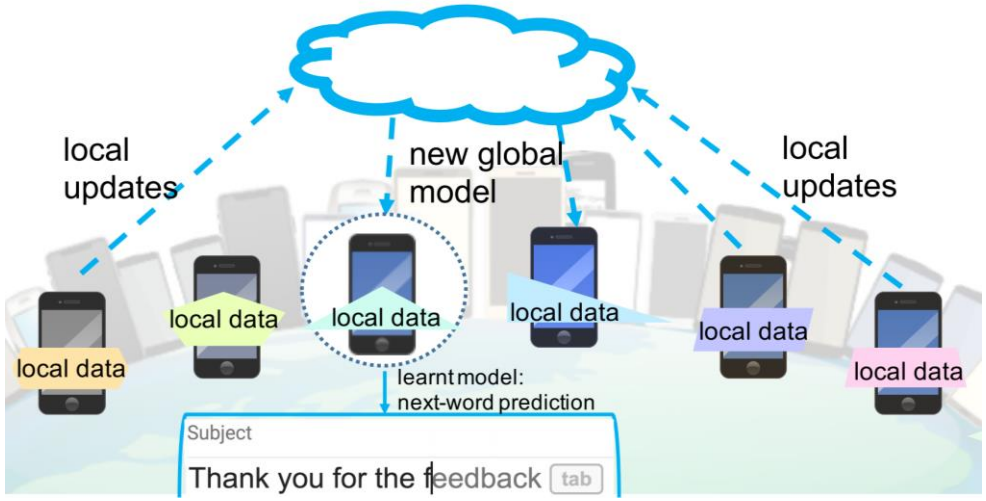
1 2 3 ... n


local updates
new global model
local updates
local data
local data
local data
local data
local data
local data
learnt model:
next-word prediction
Subject
Thank you for the feedback
tab

Server

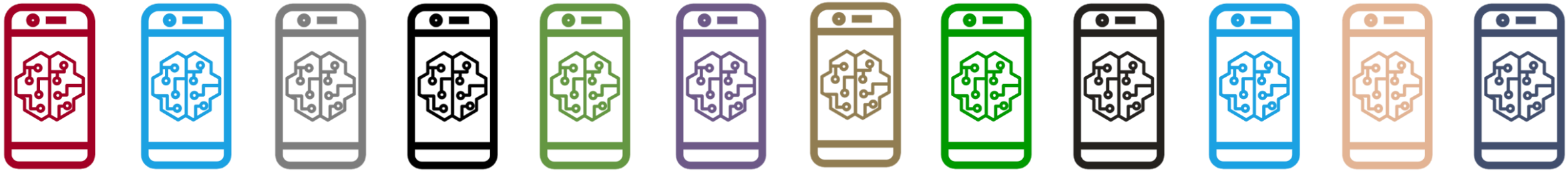


1 2 3 ... n

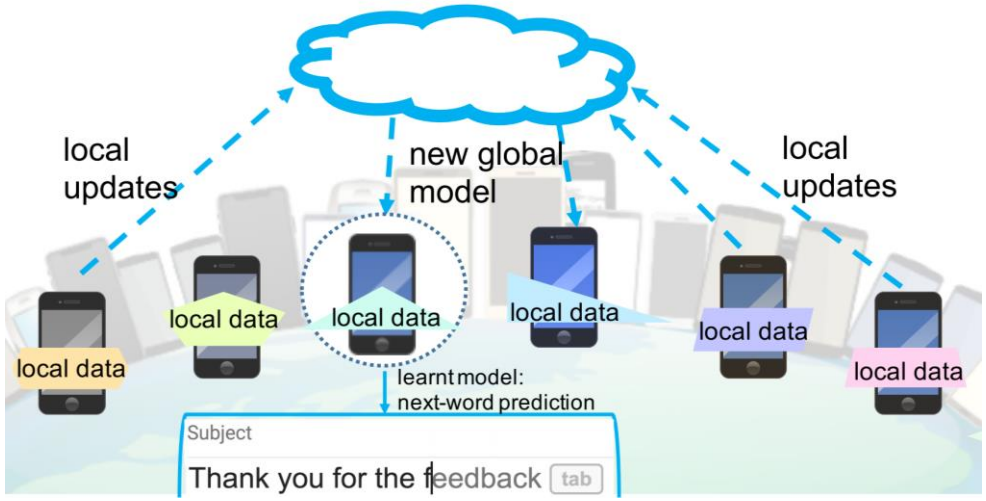


Server

In a small town a greengrocer had opene
was located above a deep cellar. Every n
in droves out of this cellar into the shop.



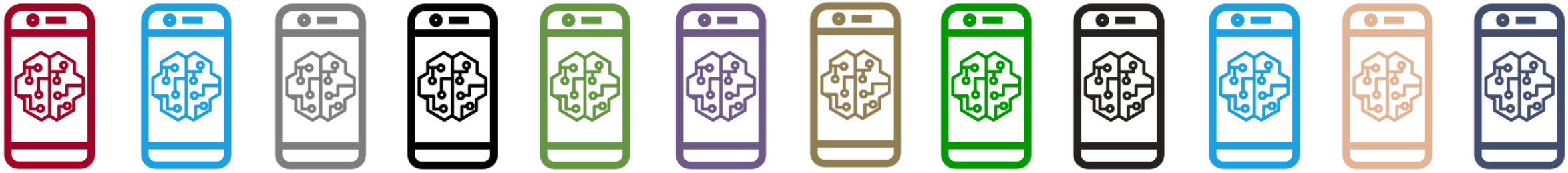
1 2 3 ... n



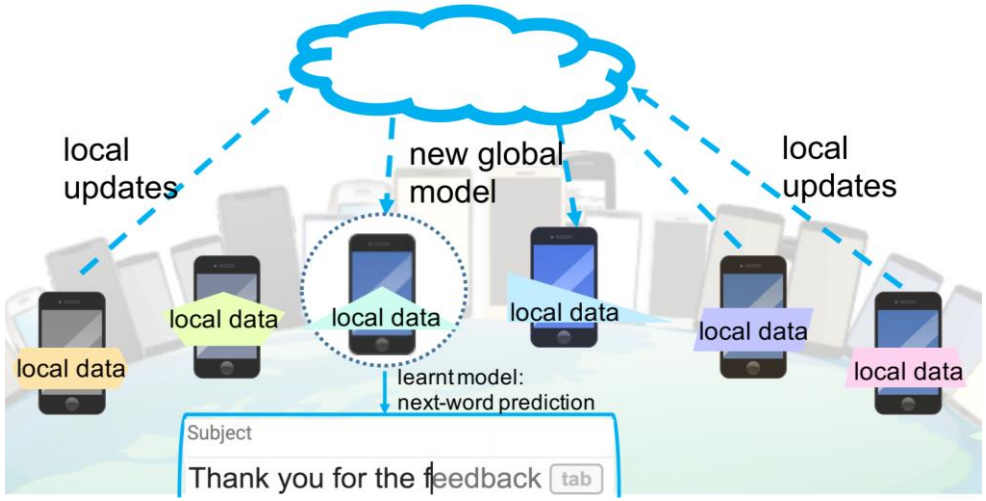
Server

In a small town a greengrocer had opene
was located above a deep cellar. Every n
in droves out of this cellar into the shop.

Where the mind is without fear
and the head is held high
Where knowledge is free
Where the world has not been broken
up into fragments by narrow domestic walls;



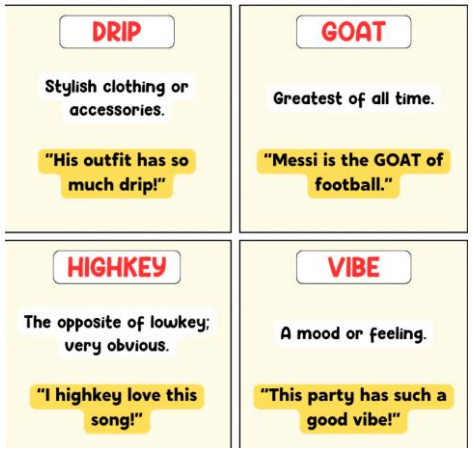
1 2 3 ... n

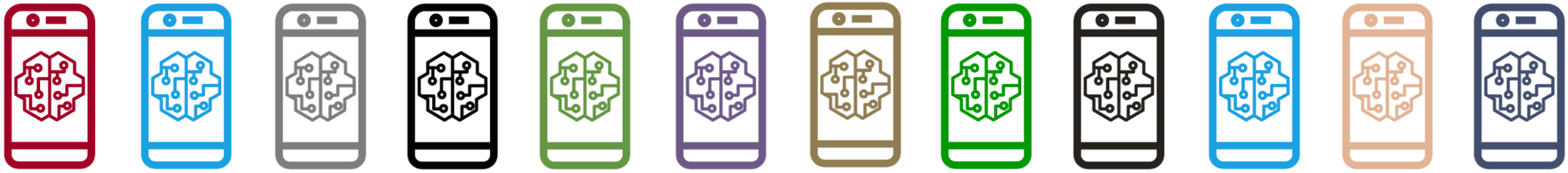


Server

In a small town a greengrocer had opene
was located above a deep cellar. Every n
in droves out of this cellar into the shop.

Where the mind is without fear
and the head is held high
Where knowledge is free
Where the world has not been broken
up into fragments by narrow domestic walls;

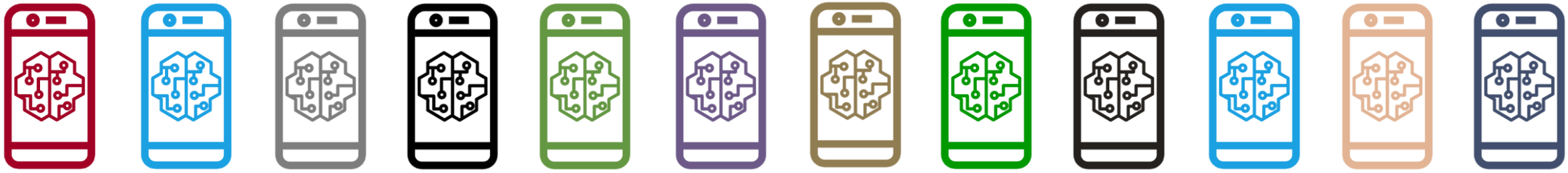




1 2 3 ... n

Data heterogeneity

Server



1 2 3 ... n

Data heterogeneity

Server

Server

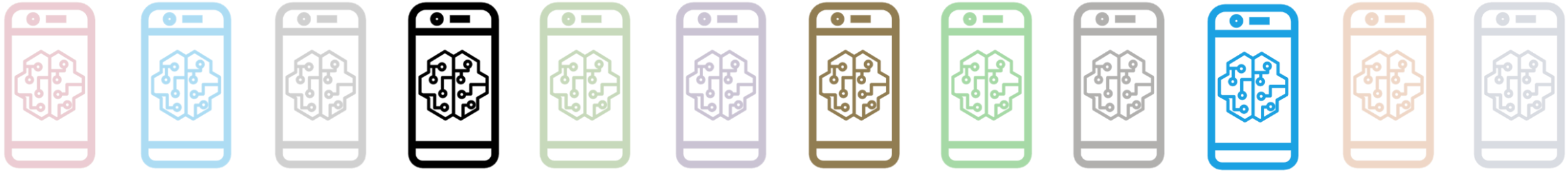
Data heterogeneity

System heterogeneity

Round t begins

- P (out of n) clients

Server



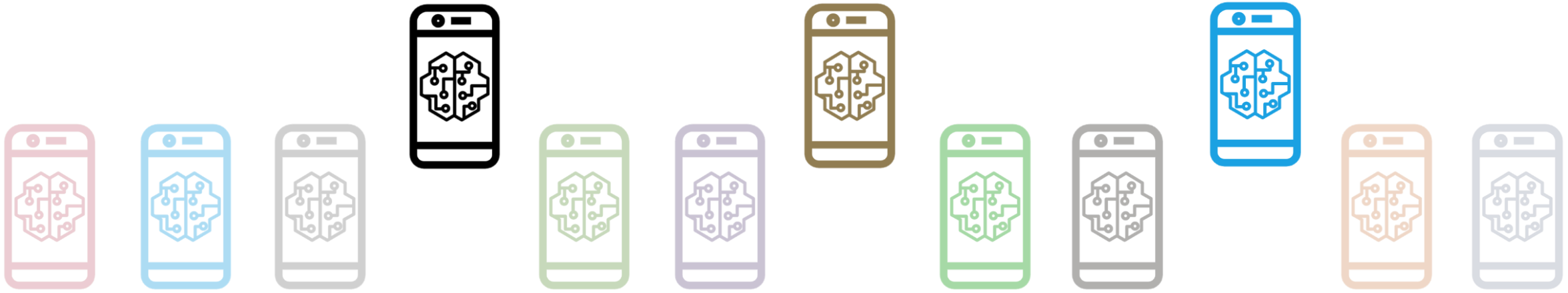
1 2 3 n

Clients

Round t begins

- P (out of n) clients

Server



Clients

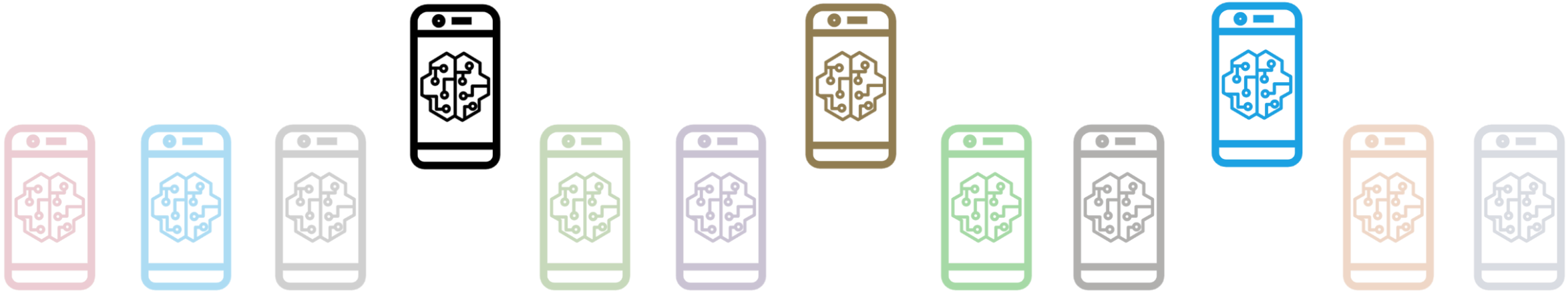
Round t begins

- P (out of n) clients

Server

Partial Client Participation

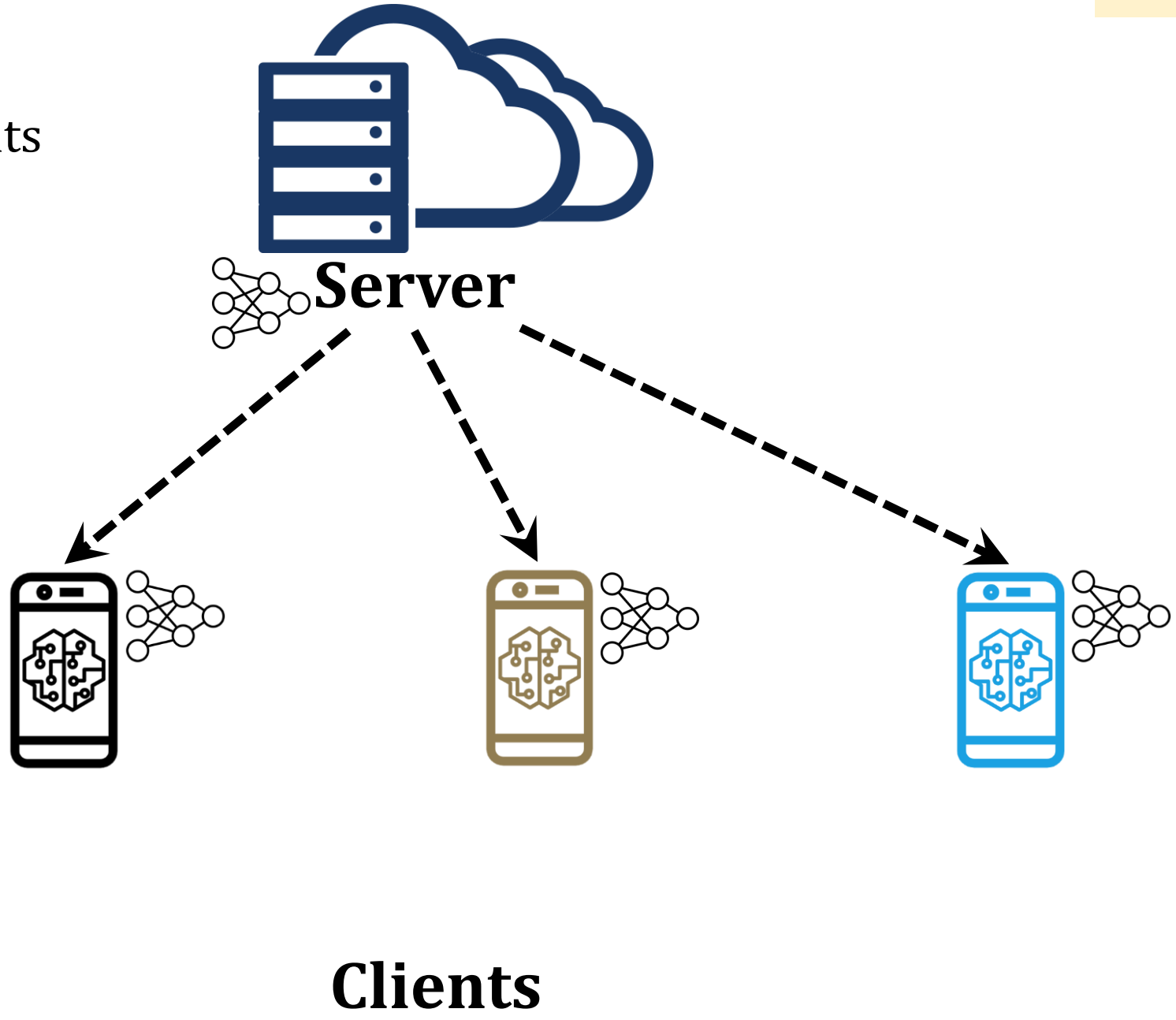
Small fraction of clients available



Clients

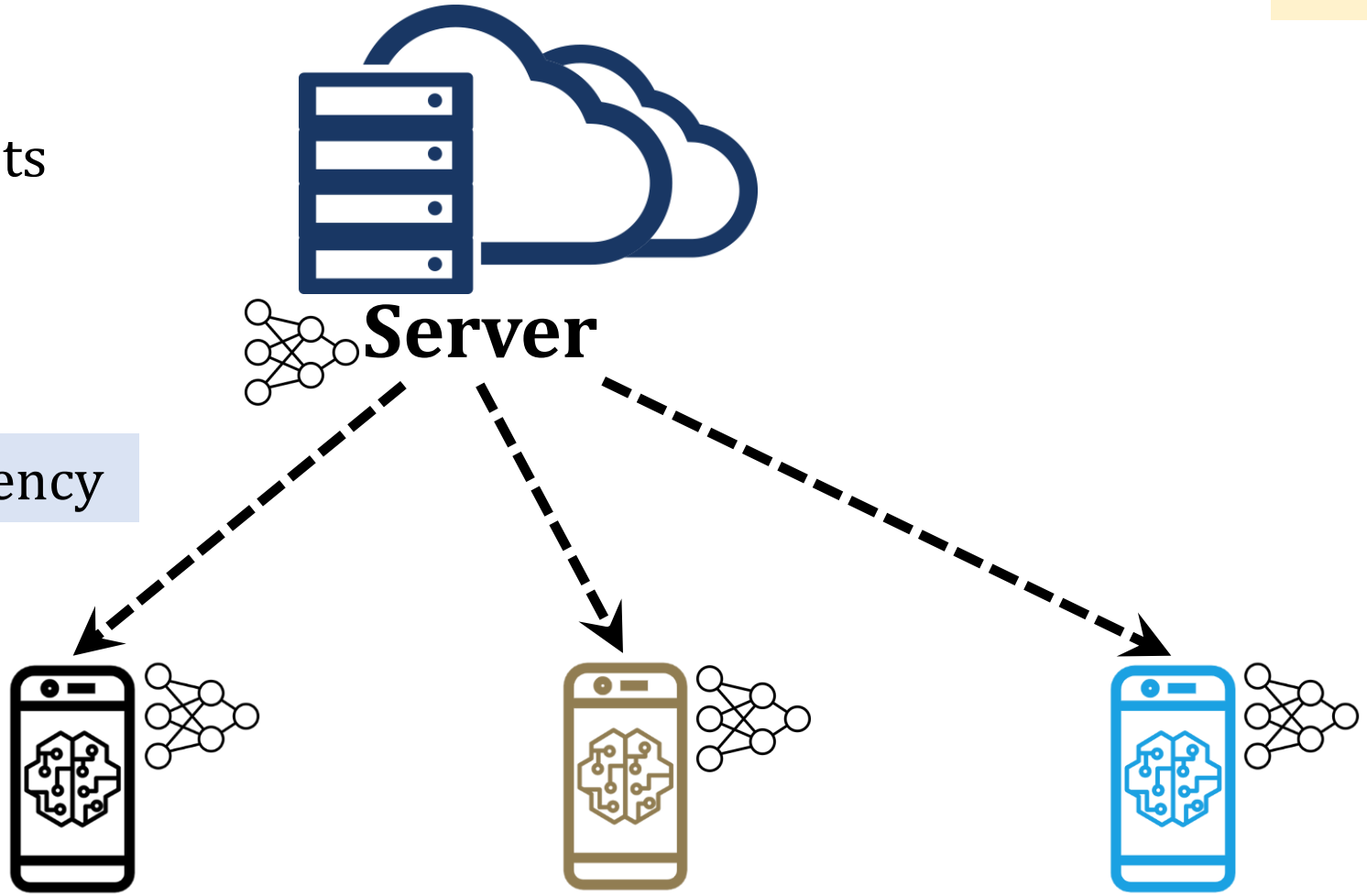
Round t

- Global model → clients



Round t

- Global model → clients



Communication Efficiency

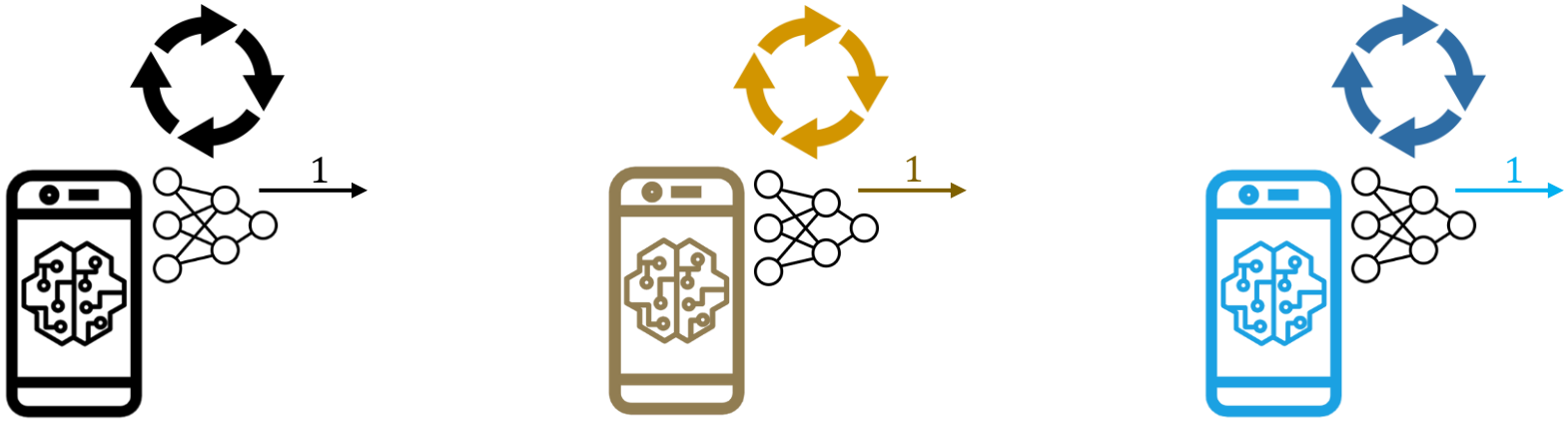
Network constraints

Infrequent

Round t

- Clients run $\tau \geq 1$ local steps

Server

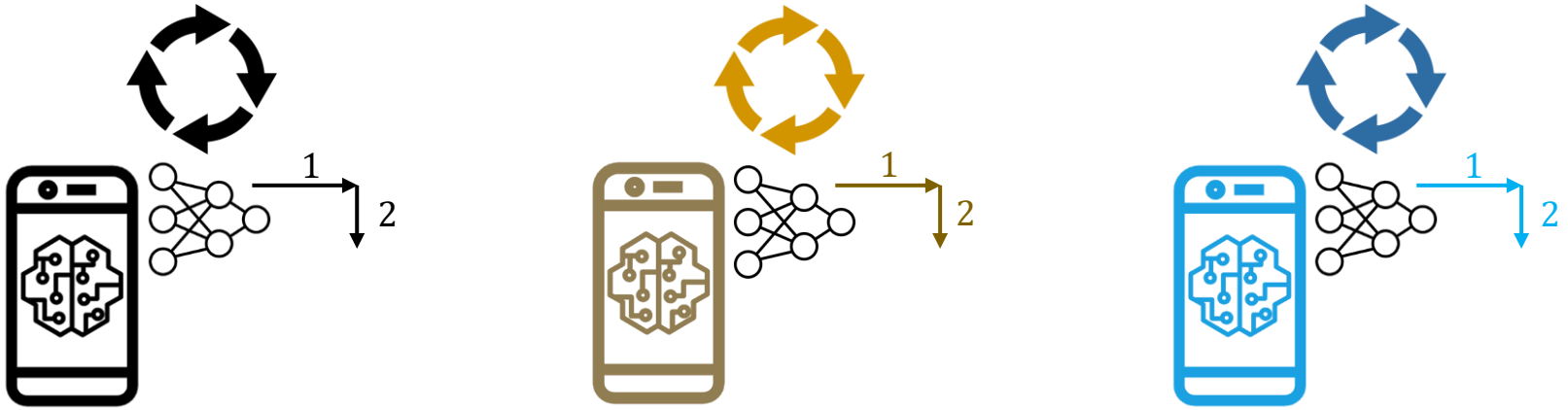


Clients

Round t

- Clients run $\tau \geq 1$ local steps

Server

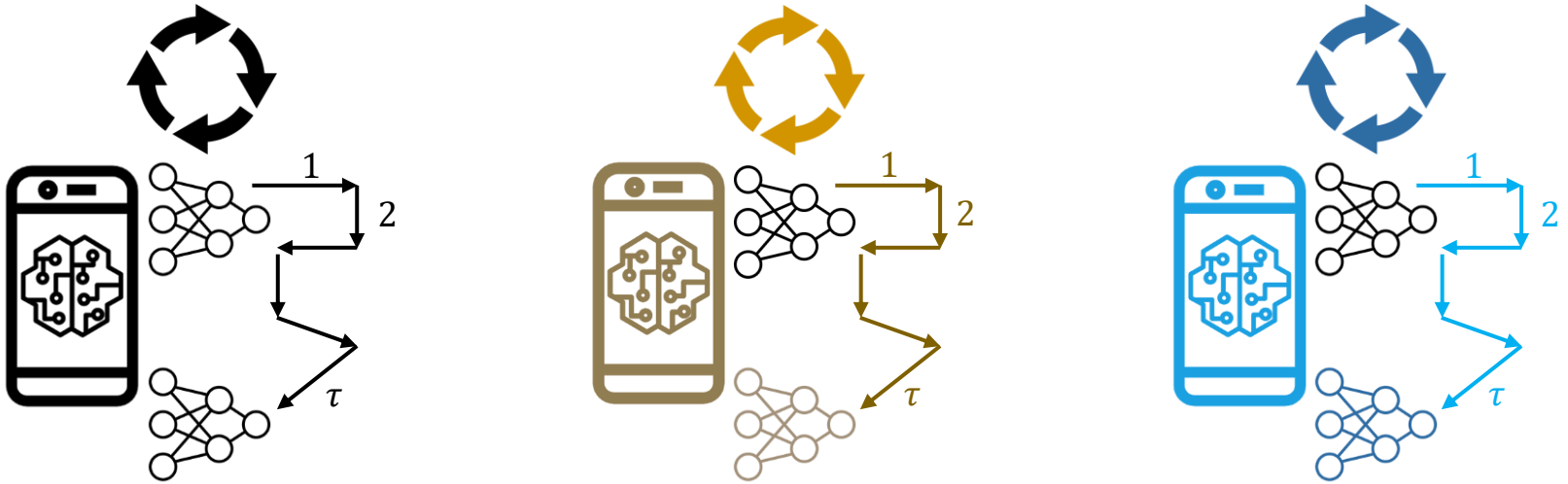


Clients

Round t

- Clients run $\tau \geq 1$ local steps

Server

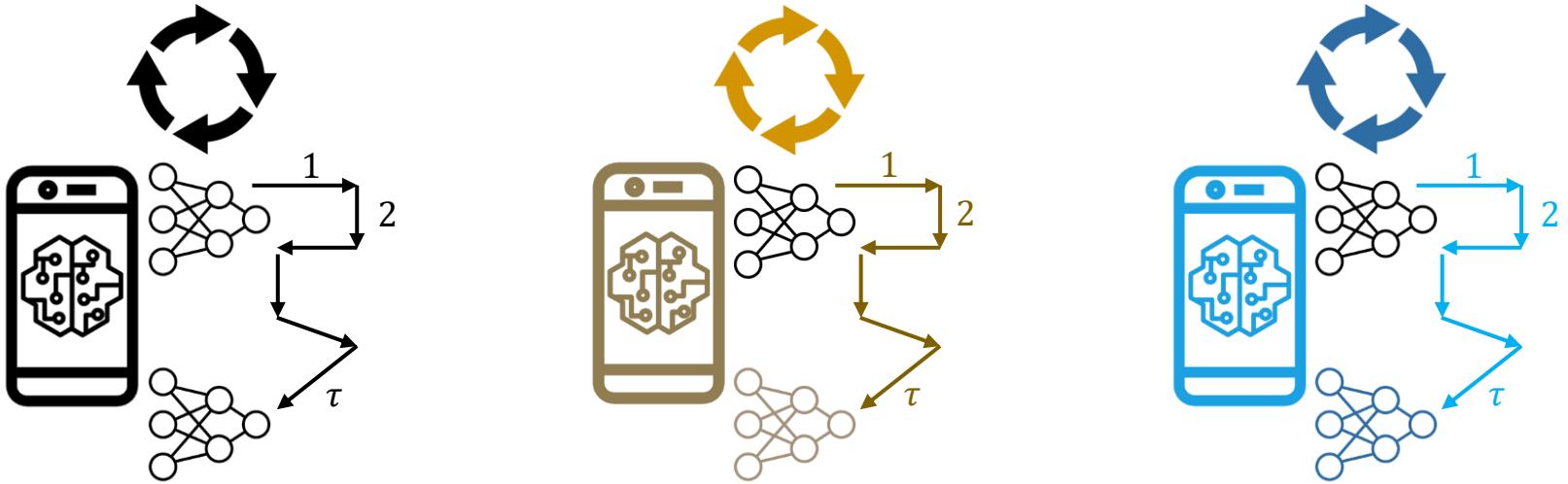


Clients

Round t

- Clients run $\tau \geq 1$ local steps
- Saves communication cost

Server



Clients

Round t

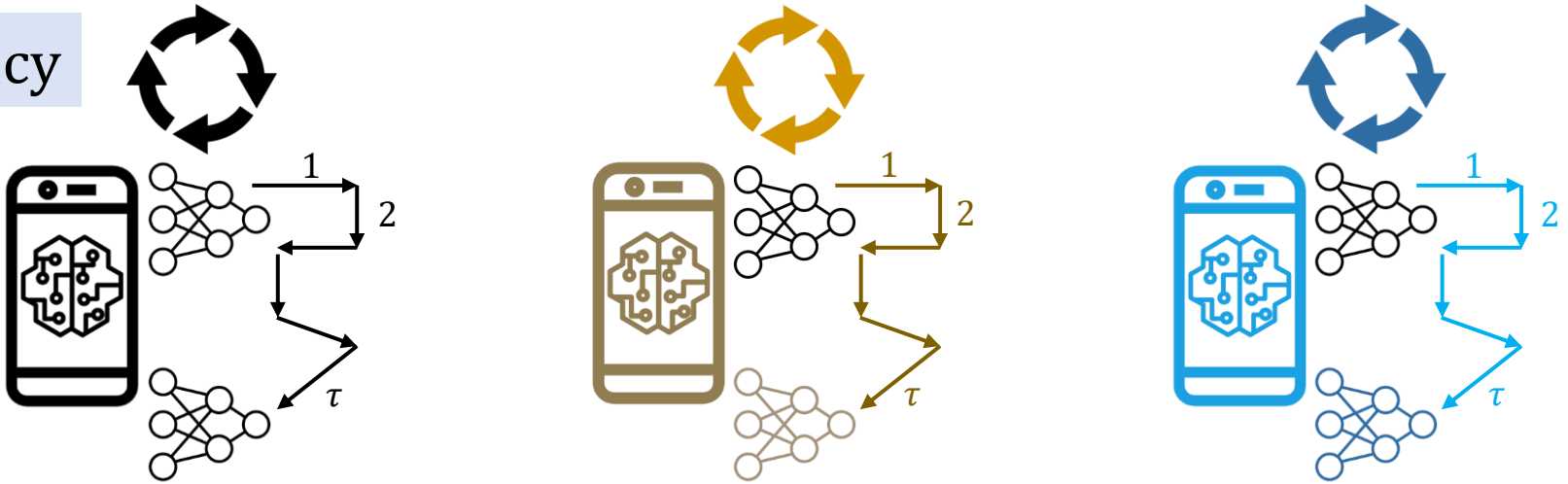
- Clients run $\tau \geq 1$ local steps
- Saves communication cost

Server

Computation Efficiency

Low-power devices

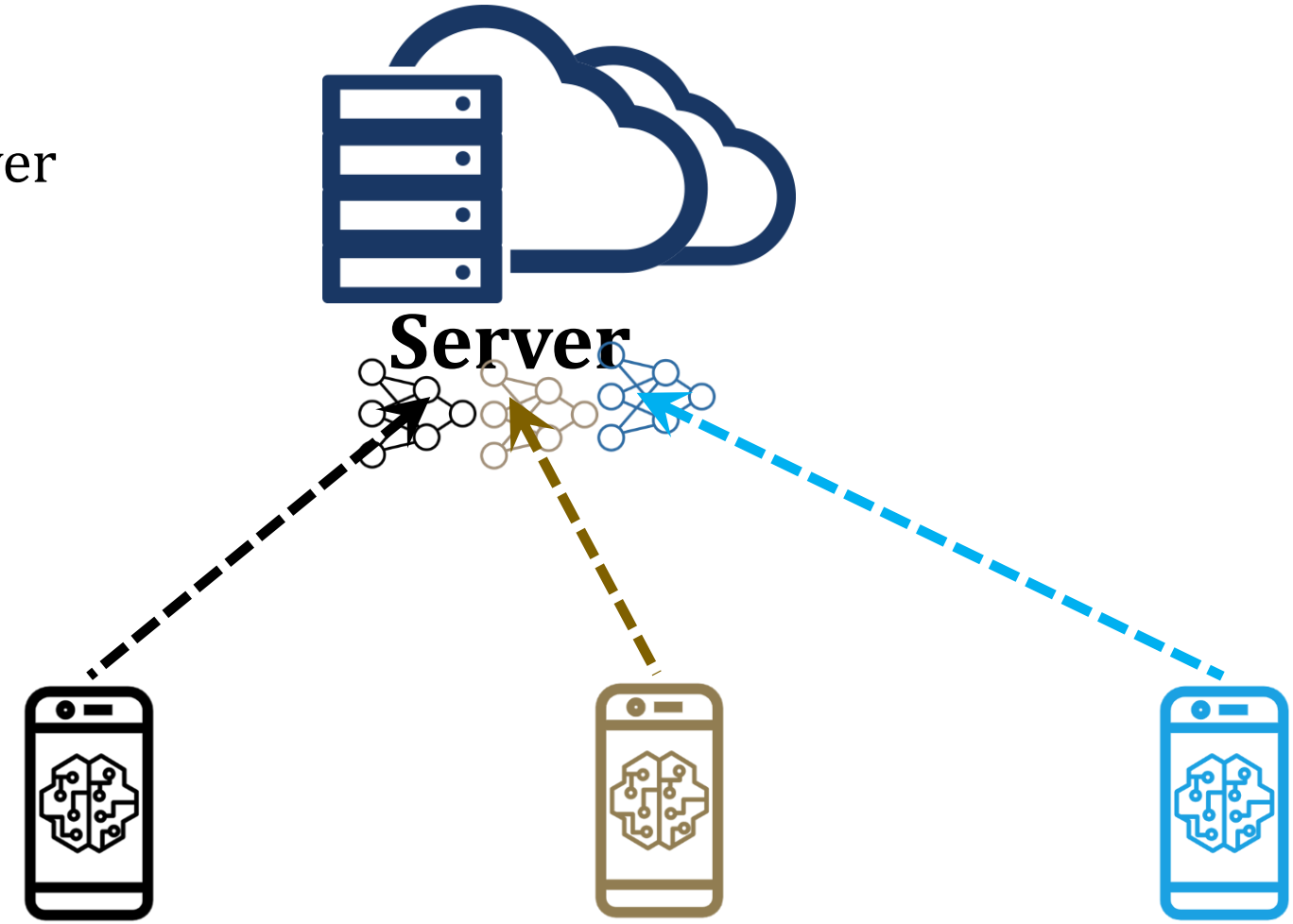
First-order methods



Clients

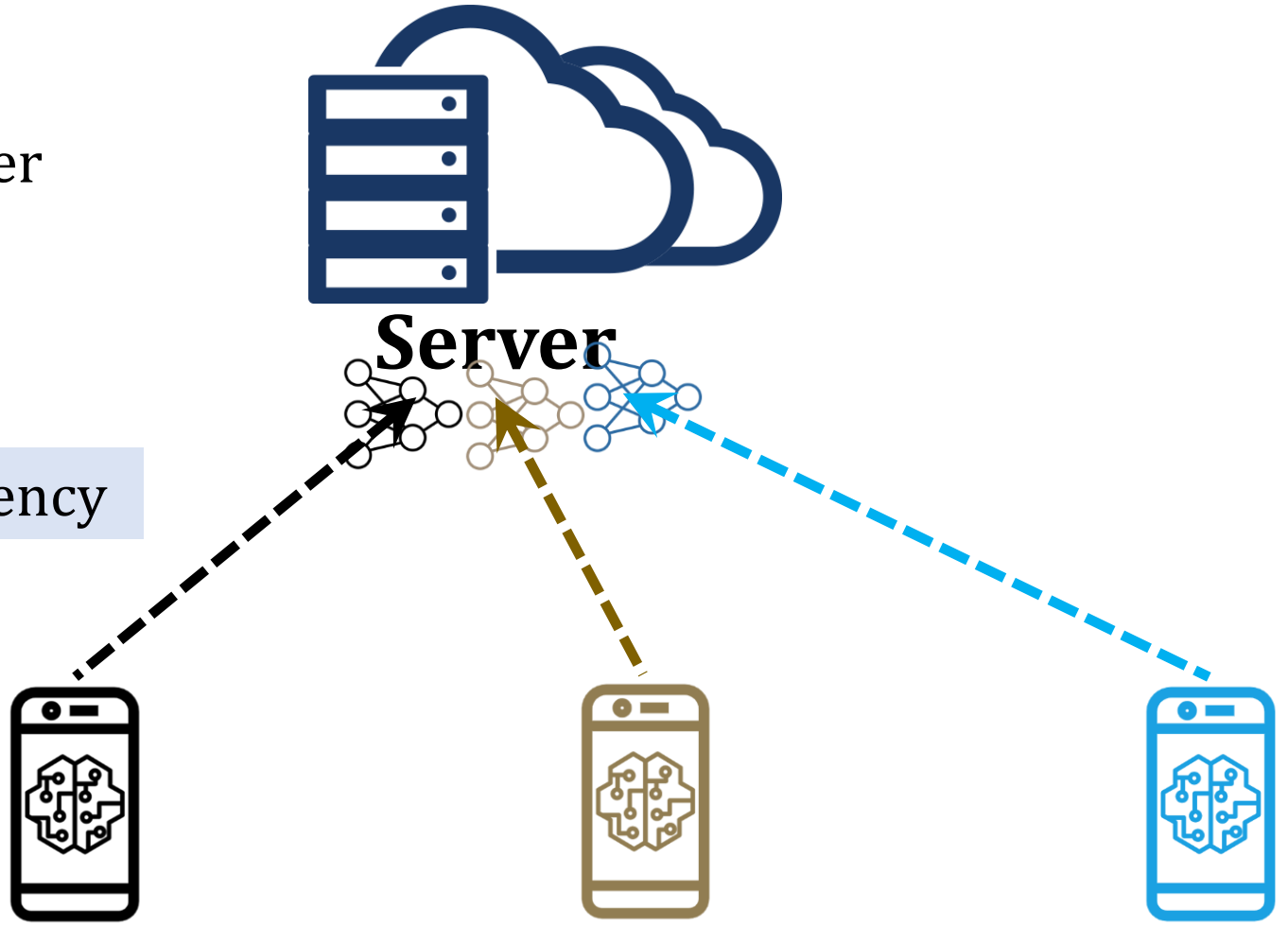
Round t

- Local models → server



Round t

- Local models → server



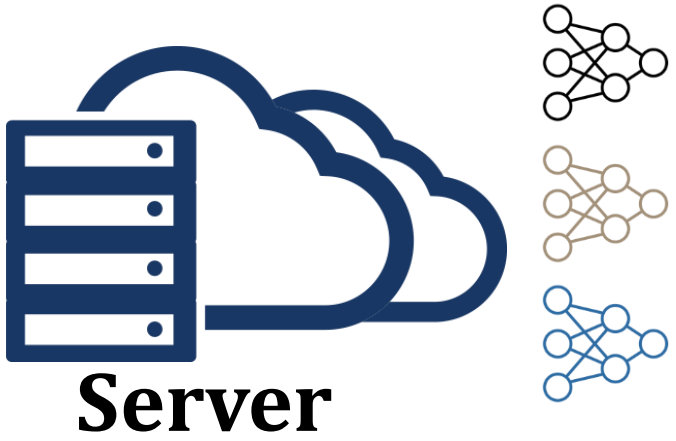
Communication Efficiency

Network constraints

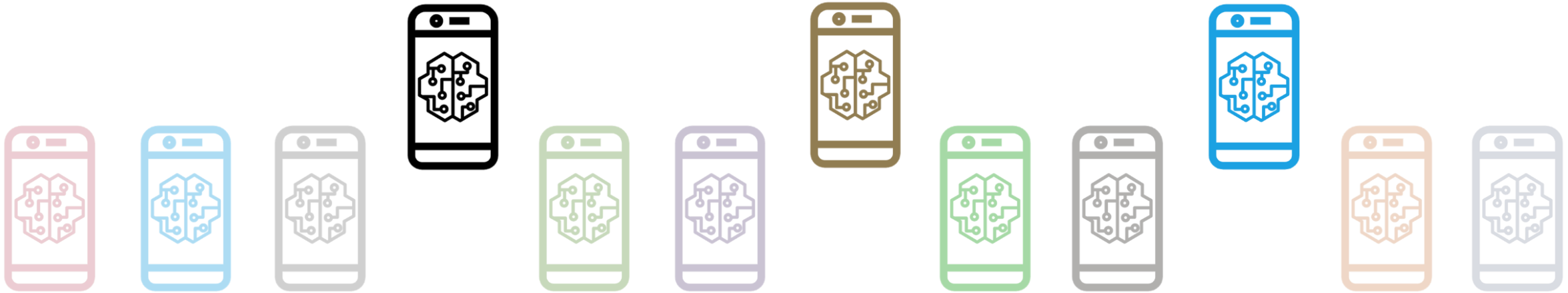
Infrequent

Round t

- Server aggregates local models



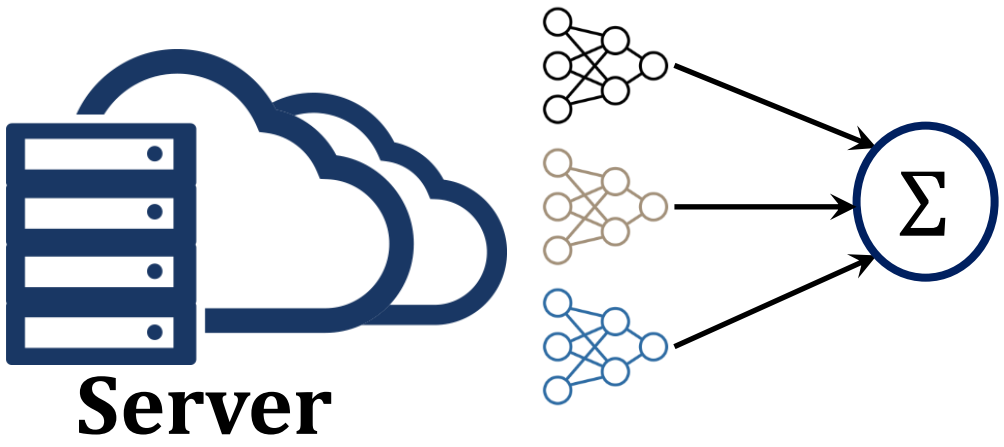
Server



Clients

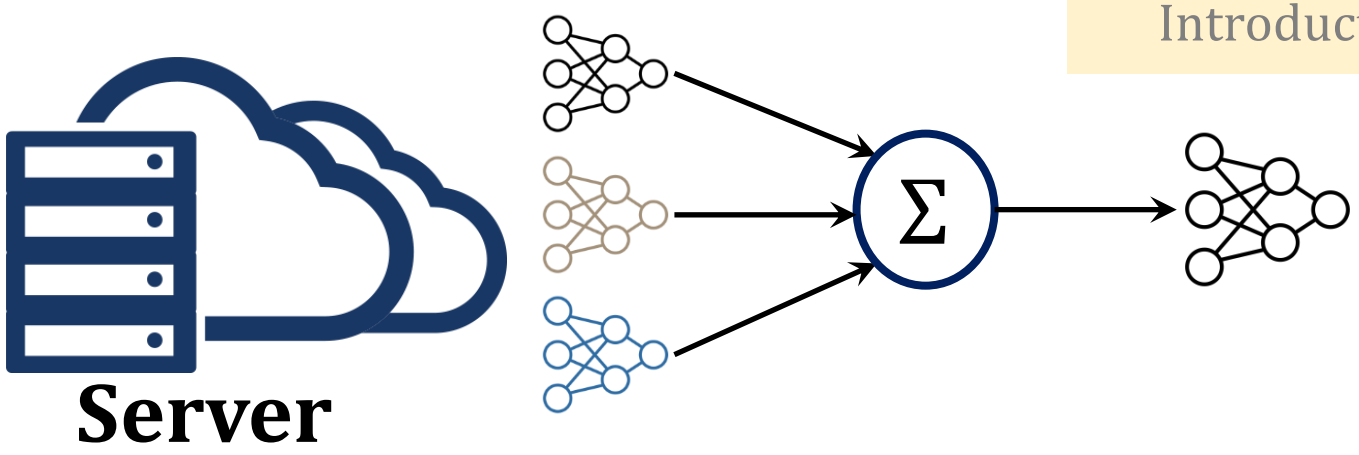
Round t

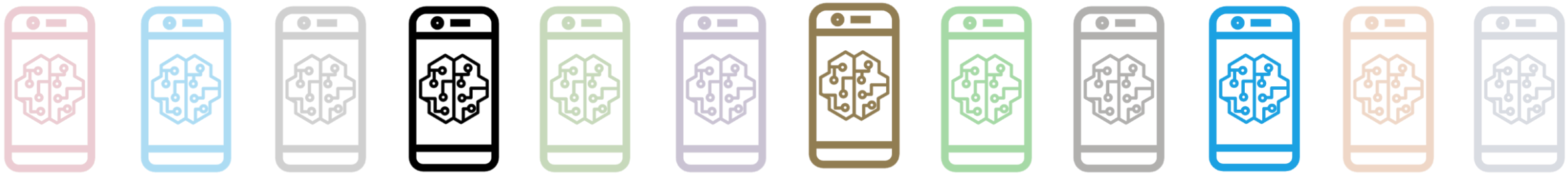
- Server aggregates local models



Round t

- Server aggregates local models

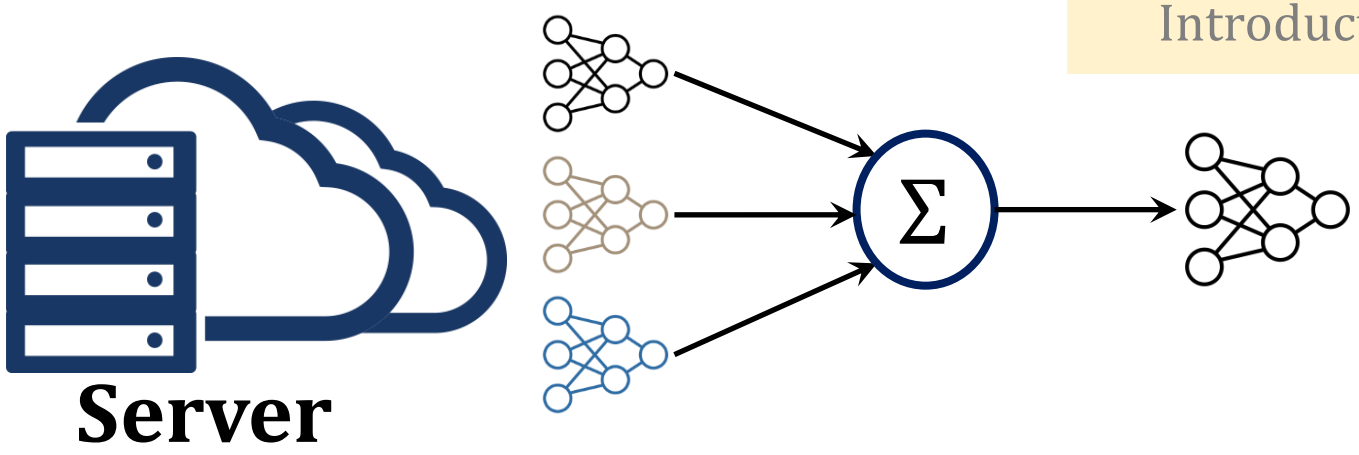


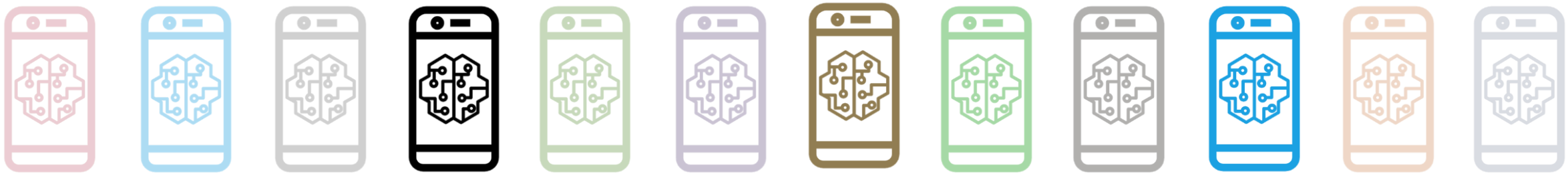


Round t

- Server aggregates local models

Round t ends

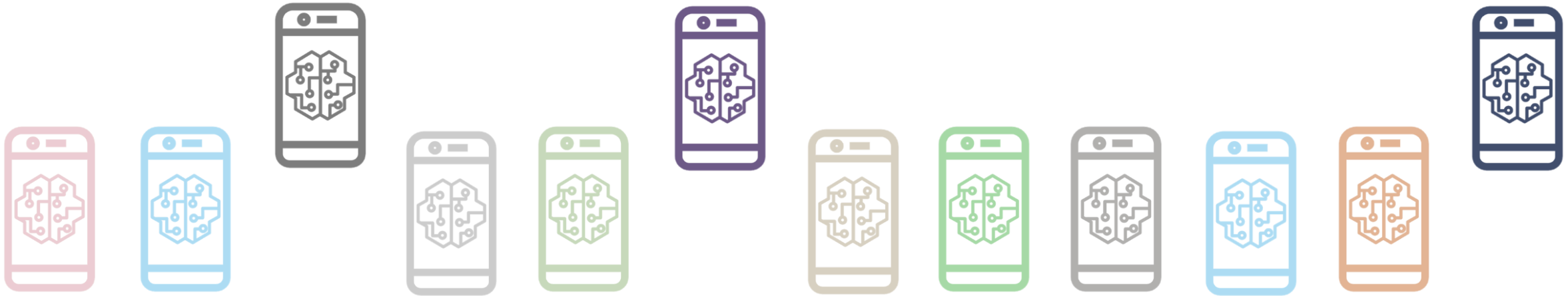




Round $t + 1$ begins

- P (out of n) clients

Server



1 2 3 n

Clients

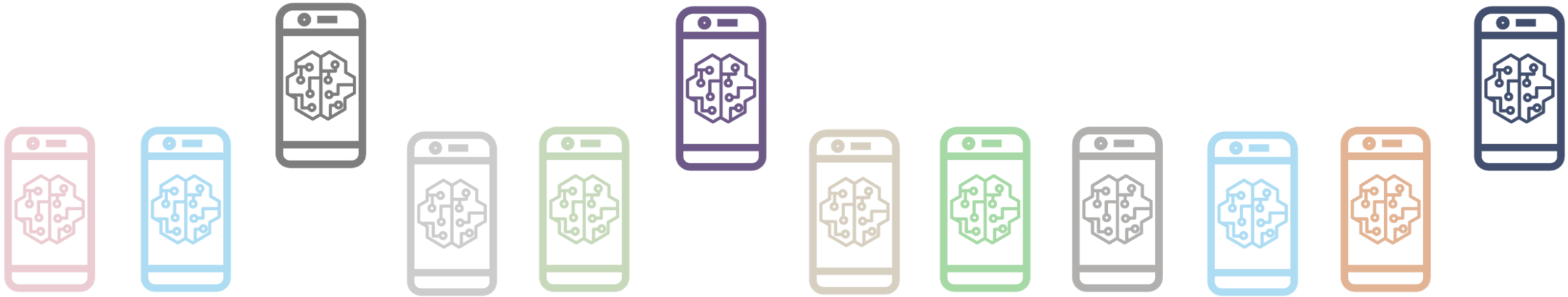
Round $t + 1$ begins

- P (out of n) clients

Server

Partial Client Participation

Small fraction of clients available

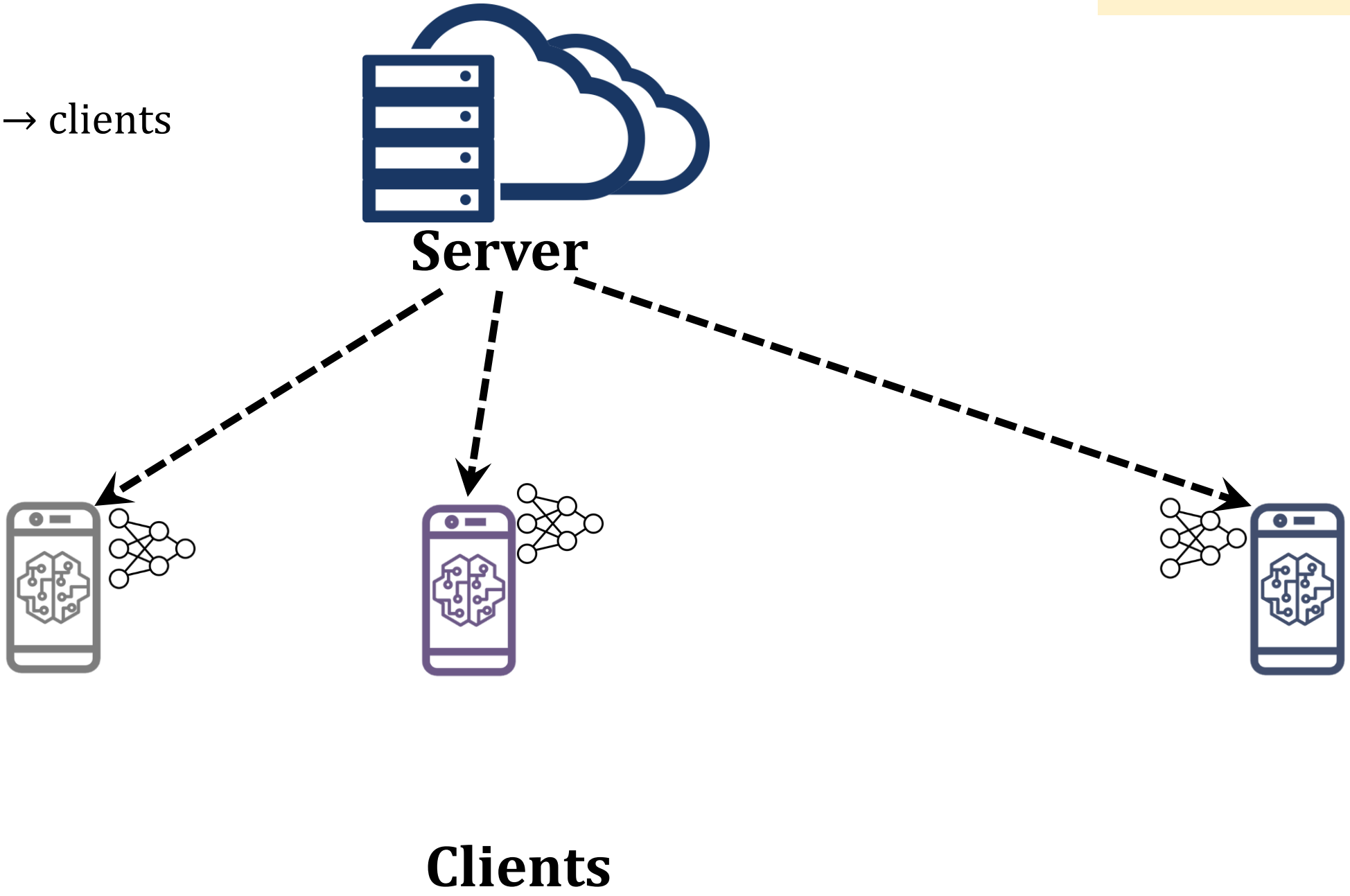


1 2 3 n

Clients

Round $t + 1$

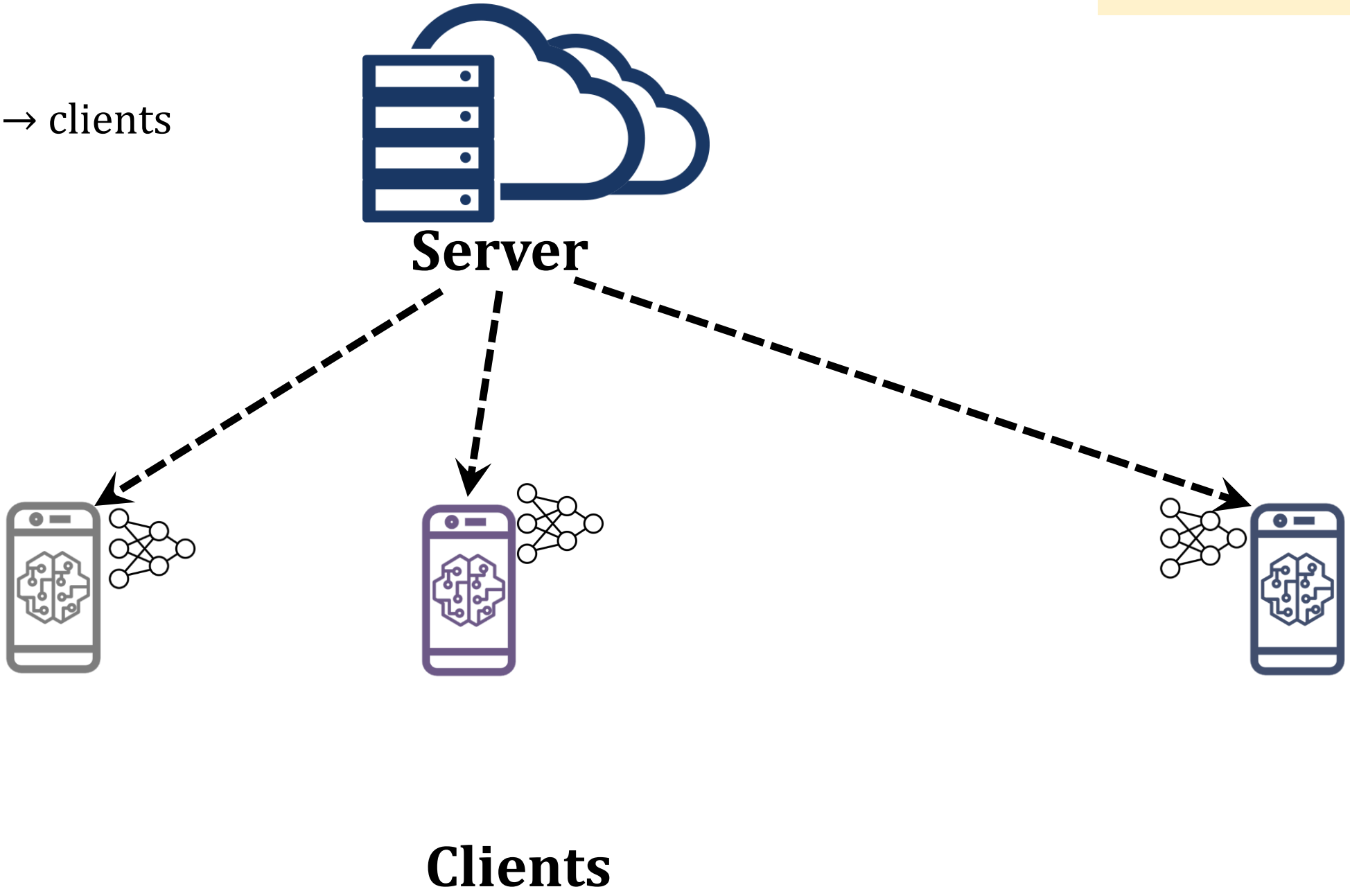
- Global model → clients



Round $t + 1$

- Global model → clients

Total T rounds



ChatBots

Multiple Objectives

- Accuracy/Relevance
- Engagement
- Politeness/Safety
- Personalization

Trade-offs: accuracy $\leftrightarrow$ personalization

Recommender Systems*

Image modified from MLArchive

Multiple Users

- User Diversity
- Data Privacy

Multiple Objectives

- Accuracy/Relevance
- Diversity
- Novelty
- Contextual Relevance

Trade-offs: accuracy $\leftrightarrow$ diversity

*Sun, et al. "A survey on federated recommendation systems." *IEEE TNNLS'24*.

Personalized Medicine*

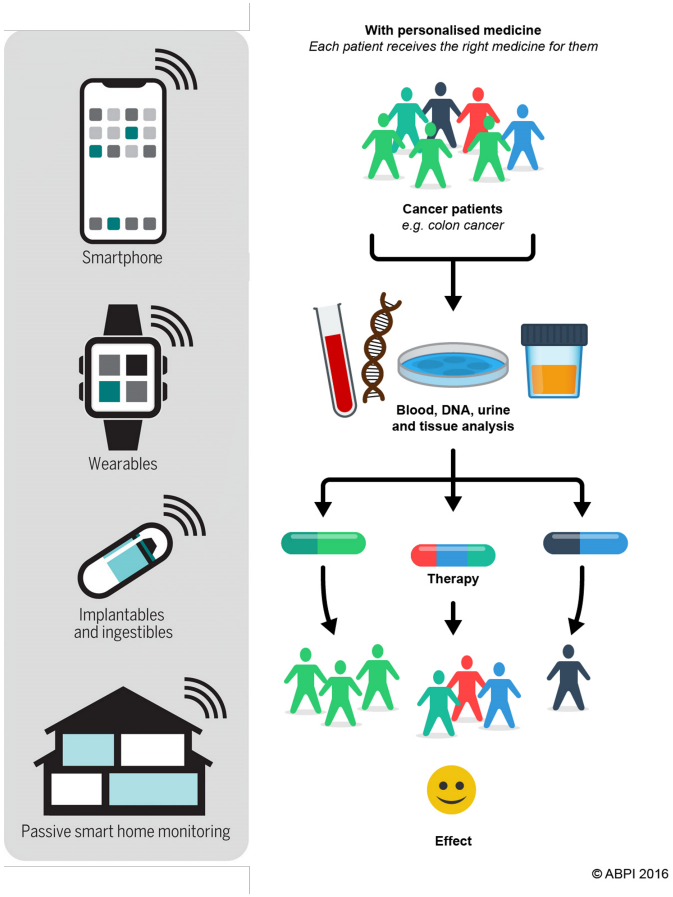


Image from science.org and eupati.eu

Multiple Patients

- Patient diversity
- Data confidentiality

Multiple Objectives

- Diagnostic Precision
- Cost-Effectiveness
- Minimizing Side-Effects
- Privacy and Data Security
- Equity in Access

Trade-offs:

Precision $\leftrightarrow$ Cost-Effectiveness

*Sheller, et al. "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data." *Scientific Reports'20*.

Goal

Using data from multiple (**distributed**) users/patients/clients,
learn a model
that **simultaneously** optimizes multiple objectives

In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**

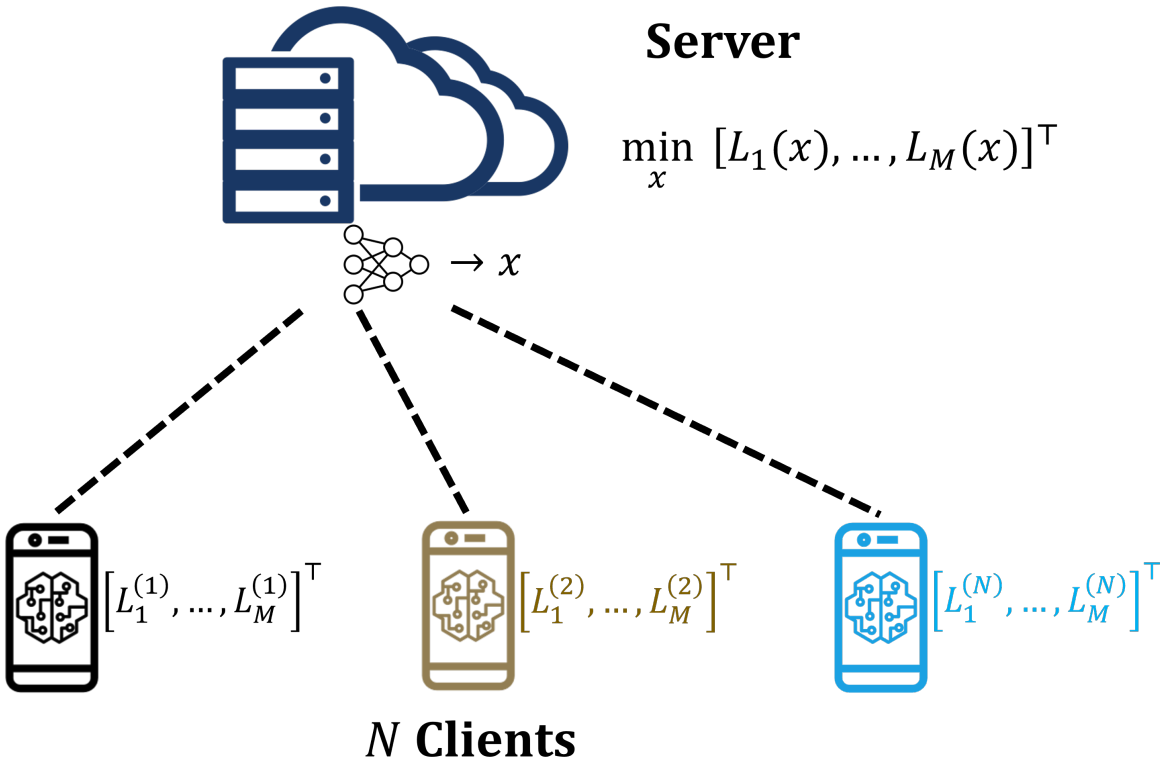
In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**



In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**

- Communication-efficient algorithm FedCMOO

In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**

- Communication-efficient algorithm FedCMOO
  - Communication cost does not scale with number of objectives M

In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**

- Communication-efficient algorithm FedCMOO
  - Communication cost does not scale with number of objectives M
- Convergence to *Pareto-stationary point*

In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**

- Communication-efficient algorithm FedCMOO
  - Communication cost does not scale with number of objectives M
- Convergence to *Pareto-stationary point*
  - Improved sample complexity - under weaker assumptions

In this talk

Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$L_k^{(i)}$ - contribution of i-th client to k-th objective

M objectives, N users/patients/**clients**

- Communication-efficient algorithm FedCMOO
  - Communication cost does not scale with number of objectives M
- Convergence to *Pareto-stationary point*
  - Improved sample complexity - under weaker assumptions
- Improved empirical performance

Outline

1. Background: MOO and MGDA
2. Federated MGDA and its Drawbacks
3. FedCMOO: Improving Federated MGDA
4. Experiment Results
5. Conclusion

Background: MOO and MGDA

Multi-Objective Optimization (MOO)*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

Why do we need it? Why not just solve $\min_{\mathbf{x}} \sum_{k=1}^{M} \alpha_k L_k(\mathbf{x})$?

---

†Hu, et al. "Revisiting Scalarization in Multi-Task Learning: A Theoretical Perspective," *NeurIPS'23*.

*Sener and Koltun, "Multi-task learning as multi-objective optimization," *NeurIPS'18*.

Multi-Objective Optimization (MOO)*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

Why do we need it? Why not just solve $\min_{\mathbf{x}} \sum_{k=1}^{M} \alpha_k L_k(\mathbf{x})$?

- How do we decide the weights $\{\alpha_k\}_{k=1}^{M}$?

---

†Hu, et al. "Revisiting Scalarization in Multi-Task Learning: A Theoretical Perspective," *NeurIPS'23*.

*Sener and Koltun, "Multi-task learning as multi-objective optimization," *NeurIPS'18*.

Multi-Objective Optimization (MOO)*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

Why do we need it? Why not just solve $\min_{\mathbf{x}} \sum_{k=1}^{M} \alpha_k L_k(\mathbf{x})$?

- How do we decide the weights $\{\alpha_k\}_{k=1}^{M}$?
- A subset of tasks might be **severely under-optimized**

---

†Hu, et al. "Revisiting Scalarization in Multi-Task Learning: A Theoretical Perspective," *NeurIPS'23*.

*Sener and Koltun, "Multi-task learning as multi-objective optimization," *NeurIPS'18*.

Multi-Objective Optimization (MOO)*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

Why do we need it? Why not just solve $\min_{\mathbf{x}} \sum_{k=1}^{M} \alpha_k L_k(\mathbf{x})$?

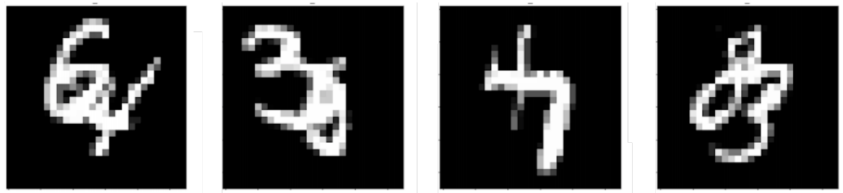
- How do we decide the weights $\{\alpha_k\}_{k=1}^{M}$?
- A subset of tasks might be **severely under-optimized**
- Scalar losses do not explore the set of all possible solutions†

†Hu, et al. "Revisiting Scalarization in Multi-Task Learning: A Theoretical Perspective," *NeurIPS'23*.

*Sener and Koltun, "Multi-task learning as multi-objective optimization," *NeurIPS'18*.

MGDA with Multi-MNIST Dataset*

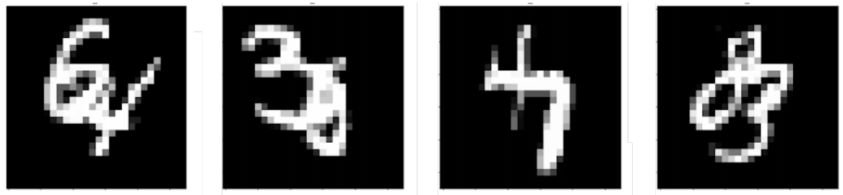
Two digits - one in top-left, the other in bottom-right



*Sener and Koltun, "Multi-task learning as multi-objective optimization," *NeurIPS'18*.

MGDA with Multi-MNIST Dataset*

Two digits - one in top-left, the other in bottom-right



Task-L/R: classifying the digit on top-left/bottom-right

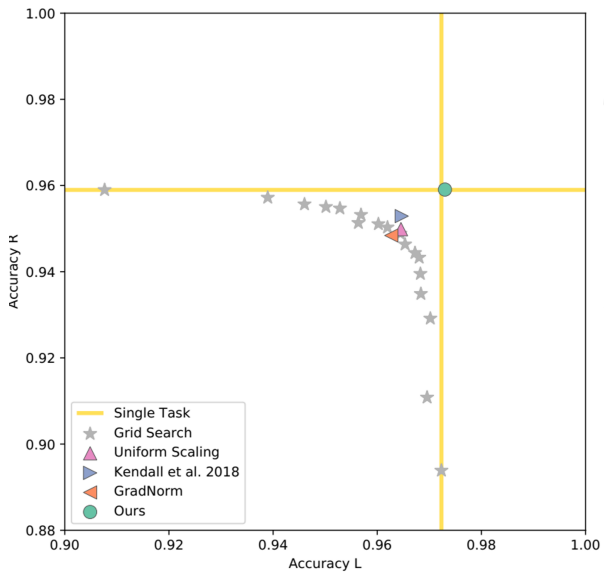


Table 3: Performance of MTL algorithms on MultiMNIST. Single-task baselines solve tasks separately, with dedicated models, but are shown in the same row for clarity.

	Left digit accuracy [%]	Right digit accuracy [%]
Single task	**97.23**	**95.90**
Uniform scaling	96.46	94.99
Kendall et al. 2018	96.47	95.29
GradNorm	96.27	94.84
Ours	**97.26**	**95.90**

*Sener and Koltun, "Multi-task learning as multi-objective optimization," *NeurIPS'18*.

Pareto Optimality/Stationarity

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

- At **Pareto-optimal** point - cannot reduce any loss L_k without also increasing some other $L_{k'}$

Pareto Optimality/Stationarity

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

- At **Pareto-optimal** point - cannot reduce any loss L_k without also increasing some other $L_{k'}$
- At **Pareto-stationary** point $\bar{\mathbf{x}}$ - "some" convex combination of gradient vectors is zero, i.e., there exist $w_1, \ldots, w_M$ (all ≥ 0) s.t.

$$\sum_{i=1}^{M} w_k = 1 \quad \text{and} \quad \sum_{i=1}^{M} w_k \nabla L_k(\bar{\mathbf{x}}) = \mathbf{0}$$

  i.e., no common descent direction for all objectives

Pareto Optimality/Stationarity

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

- At **Pareto-optimal** point - cannot reduce any loss L_k without also increasing some other $L_{k'}$
- At **Pareto-stationary** point $\bar{\mathbf{x}}$ - "some" convex combination of gradient vectors is zero, i.e., there exist $w_1, \ldots, w_M$ (all ≥ 0) s.t.

$$\sum_{i=1}^{M} w_k = 1 \quad \text{and} \quad \sum_{i=1}^{M} w_k \nabla L_k(\bar{\mathbf{x}}) = \mathbf{0}$$

  i.e., no common descent direction for all objectives
- Pareto-optimality $\Rightarrow$ Pareto-stationarity

Pareto Optimality/Stationarity

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top$$

- At **Pareto-optimal** point - cannot reduce any loss L_k without also increasing some other $L_{k'}$
- At **Pareto-stationary** point $\bar{\mathbf{x}}$ - "some" convex combination of gradient vectors is zero, i.e., there exist $w_1, \ldots, w_M$ (all ≥ 0) s.t.

$$\sum_{i=1}^{M} w_k = 1 \quad \text{and} \quad \sum_{i=1}^{M} w_k \nabla L_k(\bar{\mathbf{x}}) = \mathbf{0}$$

  i.e., no common descent direction for all objectives
- Pareto-optimality $\Rightarrow$ Pareto-stationarity
- $\therefore \bar{\mathbf{x}}$ is not Pareto-stationary $\Rightarrow$ can find descent directions to simultaneously reduce all $\{L_k\}_{k=1}^M$

Multiple Gradient Descent

Multiple Gradient Descent Algorithm (MGDA)* solves

$$\min_{w_1,\ldots,w_M} \left\| \sum_{i=1}^{M} w_k \nabla L_k(\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \sum_{i=1}^{M} w_k = 1, w_k \geq 0, \text{ for all } k \in [M]$$

*Désidéri, "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization," *Comptes Rendus Mathematique, 2012*.

Multiple Gradient Descent

Multiple Gradient Descent Algorithm (MGDA)* solves

$$\min_{w_1,\ldots,w_M} \left\| \sum_{i=1}^{M} w_k \nabla L_k(\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \sum_{i=1}^{M} w_k = 1, w_k \geq 0, \text{ for all } k \in [M]$$

- Either the minimum value is 0,

*Désidéri, "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization," *Comptes Rendus Mathematique, 2012*.

Multiple Gradient Descent

Multiple Gradient Descent Algorithm (MGDA)* solves

$$\min_{w_1,\ldots,w_M} \left\| \sum_{i=1}^{M} w_k \nabla L_k(\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \sum_{i=1}^{M} w_k = 1, w_k \geq 0, \text{ for all } k \in [M]$$

- Either the minimum value is 0,
- Or the solution gives $\sum_{i=1}^{M} w_k^* \nabla L_k(\mathbf{x})$ that reduces all objectives

*Désidéri, "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization," *Comptes Rendus Mathematique, 2012*.

Outline

1. Background: MOO and MGDA
2. Federated MGDA and its Drawbacks
3. FedCMOO: Improving Federated MGDA
4. Experiment Results
5. Conclusion

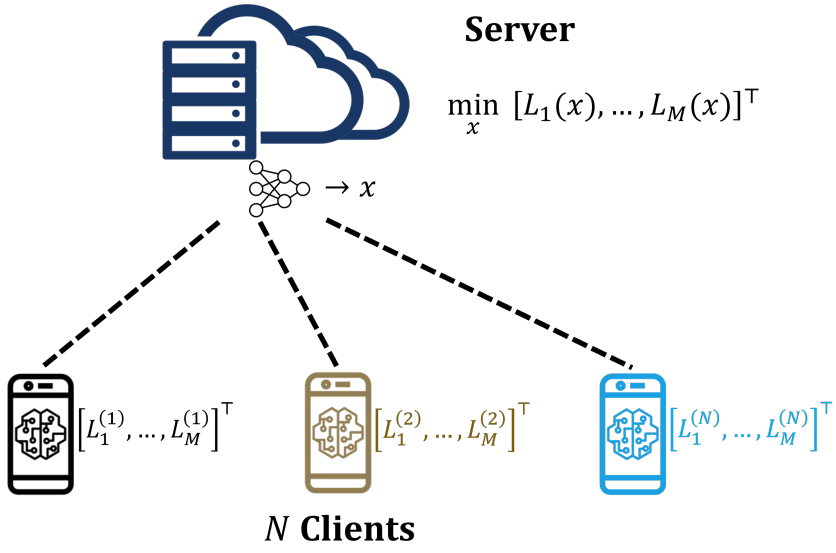
Federated MGDA and its Drawbacks

Problem - Federated Multi-Objective Optimization

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top$$

$$L_k(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- M objectives, N clients



First Attempt - Federated MGDA*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

M objectives, N clients

Recall MGDA - $\min_{\mathbf{w}} \left\| \sum_{i=1}^{M} w_k \nabla L_k(\mathbf{x}) \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0}$, $\mathbf{w}^\top \mathbf{1} = 1$

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

First Attempt - Federated MGDA*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

M objectives, N clients

Recall MGDA - $\min_{\mathbf{w}} \left\| \sum_{i=1}^{M} w_k \nabla L_k(\mathbf{x}) \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0}$, $\mathbf{w}^\top \mathbf{1} = 1$

- Client i computes $\approx \left[\nabla L_1^{(i)}(\mathbf{x}), \ldots, \nabla L_M^{(i)}(\mathbf{x})\right]^\top$ (local gradients)

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

First Attempt - Federated MGDA*

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) = \min_{\mathbf{x}} \left(L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})\right)^\top, \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

M objectives, N clients

Recall MGDA - $\min_{\mathbf{w}} \left\| \sum_{i=1}^{M} w_k \nabla L_k(\mathbf{x}) \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0}$, $\mathbf{w}^\top \mathbf{1} = 1$

- Client i computes $\approx \left[\nabla L_1^{(i)}(\mathbf{x}), \ldots, \nabla L_M^{(i)}(\mathbf{x})\right]^\top$ (local gradients)
- Server computes $\approx \left[\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})\right]^\top$, where $\nabla L_k(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \nabla L_k^{(i)}(\mathbf{x})$

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$ (SGD/GD/etc.)

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$ (SGD/GD/etc.)
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$ (SGD/GD/etc.)
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$ (SGD/GD/etc.)
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k

At the server

- Average client updates $\Delta_k = \frac{1}{N} \sum_i \Delta_k^{(i)}$, for each objective $k \in M$

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$ (SGD/GD/etc.)
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k

At the server

- Average client updates $\Delta_k = \frac{1}{N} \sum_i \Delta_k^{(i)}$, for each objective $k \in M$
- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \Delta_k \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Federated MGDA*

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$ (SGD/GD/etc.)
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k

At the server

- Average client updates $\Delta_k = \frac{1}{N} \sum_i \Delta_k^{(i)}$, for each objective $k \in M$
- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \Delta_k \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$
- Update global model $\mathbf{x} \leftarrow \mathbf{x} - \eta_s \sum_k w_k^* \Delta_k$

*Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Challenges of Federated MGDA

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each task k

At the server

- Average client updates $\Delta_k = \frac{1}{N} \sum_i \Delta_k^{(i)}$, for each task $k \in M$
- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \| \sum_k w_k \Delta_k \|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0}$, $\mathbf{w}^\top \mathbf{1} = 1$
- Update global model $\mathbf{x} \leftarrow \mathbf{x} - \eta_s \sum_k w_k^* \Delta_k$

Challenges of Federated MGDA

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each task k
  - $M\times$ communication

At the server

- Average client updates $\Delta_k = \frac{1}{N} \sum_i \Delta_k^{(i)}$, for each task $k \in M$
- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \Delta_k \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$
- Update global model $\mathbf{x} \leftarrow \mathbf{x} - \eta_s \sum_k w_k^* \Delta_k$

Challenges of Federated MGDA

At each client i

- Synchronize local models $\mathbf{x}_k^{(i)} = \mathbf{x}$ for objective k and client i
- Multiple local updates $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Local drift across tasks $\Rightarrow$ worse performance
- Return updates to server $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each task k
  - $M\times$ communication

At the server

- Average client updates $\Delta_k = \frac{1}{N} \sum_i \Delta_k^{(i)}$, for each task $k \in M$
- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \Delta_k \right\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$
- Update global model $\mathbf{x} \leftarrow \mathbf{x} - \eta_s \sum_k w_k^* \Delta_k$

FedCMOO: Improving Federated MGDA

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\nabla^{(i)} = \frac{\mathbf{x} - \mathbf{x}^{(i)}}{\eta_c}$

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\nabla^{(i)} = \frac{\mathbf{x} - \mathbf{x}^{(i)}}{\eta_c}$
  - Instead of task-wise updates $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\nabla^{(i)} = \frac{\mathbf{x} - \mathbf{x}^{(i)}}{\eta_c}$
  - Instead of task-wise updates $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k
- $M\times$ reduction in communication; reduces drift across tasks

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\nabla^{(i)} = \frac{\mathbf{x} - \mathbf{x}^{(i)}}{\eta_c}$
  - Instead of task-wise updates $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k
- $M\times$ reduction in communication; reduces drift across tasks

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\nabla^{(i)} = \frac{\mathbf{x} - \mathbf{x}^{(i)}}{\eta_c}$
  - Instead of task-wise updates $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k
- $M\times$ reduction in communication; reduces drift across tasks

At the server

- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \|\sum_k w_k \Delta_k\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$
- Update global model $\mathbf{x} \leftarrow \mathbf{x} - \eta_s \sum_k w_k^* \Delta_k$

Repeat

Improving Federated MGDA

If we knew the task weights $\{w_k\}$

At each client i - aggregated local updates, starting with $\mathbf{x}$

- Multiple local updates: $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
  - Instead of $\mathbf{x}_k^{(i)} \leftarrow \mathbf{x}_k^{(i)} - \eta_c \widetilde{\nabla} L_k^{(i)}(\mathbf{x}_k^{(i)})$
- Return updates to server $\nabla^{(i)} = \frac{\mathbf{x} - \mathbf{x}^{(i)}}{\eta_c}$
  - Instead of task-wise updates $\Delta_k^{(i)} = \frac{\mathbf{x} - \mathbf{x}_k^{(i)}}{\eta_c}$ for each k
- $M\times$ reduction in communication; reduces drift across tasks

At the server

- Solve $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \|\sum_k w_k \Delta_k\|_2^2$ s.t. $\mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$
- Update global model $\mathbf{x} \leftarrow \mathbf{x} - \eta_s \sum_k w_k^* \Delta_k$

Repeat

How do we update $\mathbf{w}$ with $\{\nabla^{(i)}\}$ instead of $\{\Delta_k\}$?

Why do we need $M\times$ communication?

At the server - need to solve

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \nabla L_k(\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$= \text{argmin}_{\mathbf{w}} \left\| \underbrace{[\nabla L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]}_{\triangleq \nabla \mathbf{L}(\mathbf{x}) \in \mathbb{R}^{d \times M}} \mathbf{w} \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$\mathbf{w}^* \approx \text{Proj}_{\mathcal{S}_M} \left(\mathbf{w} - \eta_w \nabla \mathbf{L}(\mathbf{x})^\top \nabla \mathbf{L}(\mathbf{x}) \mathbf{w} \right) \qquad \text{(Single PGD step)}$$

$\mathcal{S}_M$ is the probability simplex in $\mathbb{R}^M$

Why do we need $M\times$ communication?

At the server - need to solve

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \nabla L_k(\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$= \text{argmin}_{\mathbf{w}} \left\| \underbrace{[\nabla L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]}_{\triangleq \nabla \mathbf{L}(\mathbf{x}) \in \mathbb{R}^{d \times M}} \mathbf{w} \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$\mathbf{w}^* \approx \text{Proj}_{\mathcal{S}_M} \left(\mathbf{w} - \eta_w \nabla \mathbf{L}(\mathbf{x})^\top \nabla \mathbf{L}(\mathbf{x}) \mathbf{w} \right) \qquad \text{(Single PGD step)}$$

$\mathcal{S}_M$ is the probability simplex in $\mathbb{R}^M$

- In Federated MGDA:

$$\underbrace{[\Delta_1, \ldots, \Delta_M]}_{\text{Matrix of task updates}} \approx \underbrace{\nabla \mathbf{L}(\mathbf{x})}_{\text{Jacobian}}$$

Why do we need $M\times$ communication?

At the server - need to solve

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \sum_k w_k \nabla L_k(\mathbf{x}) \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$= \text{argmin}_{\mathbf{w}} \left\| \underbrace{[\nabla L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]}_{\triangleq \nabla \mathbf{L}(\mathbf{x}) \in \mathbb{R}^{d \times M}} \mathbf{w} \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$\mathbf{w}^* \approx \text{Proj}_{\mathcal{S}_M} \left(\mathbf{w} - \eta_w \nabla \mathbf{L}(\mathbf{x})^\top \nabla \mathbf{L}(\mathbf{x}) \mathbf{w} \right) \qquad \text{(Single PGD step)}$$

$\mathcal{S}_M$ is the probability simplex in $\mathbb{R}^M$

- In Federated MGDA:

$$\underbrace{[\Delta_1, \ldots, \Delta_M]}_{\text{Matrix of task updates}} \approx \underbrace{\nabla \mathbf{L}(\mathbf{x})}_{\text{Jacobian}}$$

- How do we approximate $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ communication?

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

At clients

- Client i reshapes $\widetilde{H}_i$ to $\bar{H}_i \in \mathbb{R}^{\sqrt{dM} \times \sqrt{dM}}$

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

At clients

- Client i reshapes $\widetilde{H}_i$ to $\bar{H}_i \in \mathbb{R}^{\sqrt{dM} \times \sqrt{dM}}$
- Send $\widehat{H}_i \leftarrow$ Rand-SVD$(\bar{H}_i, r)$ (rank-r approximation of $\bar{H}_i$) to server

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

At clients

- Client i reshapes $\widetilde{H}_i$ to $\bar{H}_i \in \mathbb{R}^{\sqrt{dM} \times \sqrt{dM}}$
- Send $\widehat{H}_i \leftarrow$ Rand-SVD$(\bar{H}_i, r)$ (rank-r approximation of $\bar{H}_i$) to server
- Rank $r \leq \sqrt{d/M} \Rightarrow$ communication cost $\leq d$

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

At clients

- Client i reshapes $\widetilde{H}_i$ to $\bar{H}_i \in \mathbb{R}^{\sqrt{dM} \times \sqrt{dM}}$
- Send $\widehat{H}_i \leftarrow$ Rand-SVD$(\bar{H}_i, r)$ (rank-r approximation of $\bar{H}_i$) to server
- Rank $r \leq \sqrt{d/M} \Rightarrow$ communication cost $\leq d$

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

At clients

- Client i reshapes $\widetilde{H}_i$ to $\bar{H}_i \in \mathbb{R}^{\sqrt{dM} \times \sqrt{dM}}$
- Send $\widehat{H}_i \leftarrow$ Rand-SVD$(\bar{H}_i, r)$ (rank-r approximation of $\bar{H}_i$) to server
- Rank $r \leq \sqrt{d/M} \Rightarrow$ communication cost $\leq d$

At server

- Reshape $\widehat{H}_i$ to $H_i \in \mathbb{R}^{d \times M}$

Constructing $\nabla \mathbf{L}(\mathbf{x})$ with $\mathcal{O}(d)$ Communication

Server wants $\nabla \mathbf{L} = [\nabla L_1(\mathbf{x}), \ldots, \nabla L_M(\mathbf{x})]$

Client i has $\widetilde{H}_i = \left[\widetilde{\nabla} L_1^{(i)} \ldots \widetilde{\nabla} L_M^{(i)}\right] \in \mathbb{R}^{d \times M}$

At clients

- Client i reshapes $\widetilde{H}_i$ to $\bar{H}_i \in \mathbb{R}^{\sqrt{dM} \times \sqrt{dM}}$
- Send $\widehat{H}_i \leftarrow$ Rand-SVD$(\bar{H}_i, r)$ (rank-r approximation of $\bar{H}_i$) to server
- Rank $r \leq \sqrt{d/M} \Rightarrow$ communication cost $\leq d$

At server

- Reshape $\widehat{H}_i$ to $H_i \in \mathbb{R}^{d \times M}$
- Compute $G \leftarrow \left(\frac{1}{N}\sum_i H_i\right)^\top \left(\frac{1}{N}\sum_i H_i\right) \approx \nabla \mathbf{L}^\top \nabla \mathbf{L}$

Comparison with other Compression Schemes

Compression Method	CelebA	CIFAR10+ MNIST	MNIST+ F.MNIST	MultiMNIST	Mean
Sum of client Gram matrices	50.95	11.51	20.61	14.56	24.41
`ApproxGramJacobian` *two-way comm.* with randomized-SVD	10.15	1.38	2.04	1.76	3.83
`ApproxGramJacobian` *two-way comm.* with random masking	51.25	10.26	17.40	12.98	22.97
`ApproxGramJacobian` *two-way comm.* with top-k sparsification	17.15	0.15	0.04	0.15	4.38
`ApproxGramJacobian` *one-way comm.* with randomized-SVD	10.87	1.68	1.79	1.96	4.08
`ApproxGramJacobian` *one-way comm.* with random masking	76.54	4.23	6.29	5.70	23.19
`ApproxGramJacobian` *one-way comm.* with top-k sparsification	24.13	0.04	0.01	0.10	6.07

$$\text{Error (in \%)} = \frac{\|\widetilde{G} - G\|}{\|\widetilde{G}\|}, \text{ where } \widetilde{G} = \left(\frac{1}{N}\sum_i \widetilde{H}_i\right)^\top \left(\frac{1}{N}\sum_i \widetilde{H}_i\right)$$

Updating aggregation weights

At the server - need to solve

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \underbrace{[\nabla L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]}_{\triangleq \nabla \mathbf{L}(\mathbf{x}) \in \mathbb{R}^{d \times M}} \mathbf{w} \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$\approx \text{Proj}_{\mathcal{S}_M} \left(\mathbf{w} - \eta_w \nabla \mathbf{L}(\mathbf{x})^\top \nabla \mathbf{L}(\mathbf{x}) \mathbf{w} \right) \qquad \text{(Single PGD step)}$$

$\mathcal{S}_M$ is the probability simplex in $\mathbb{R}^M$

Updating aggregation weights

At the server - need to solve

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \left\| \underbrace{[\nabla L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]}_{\triangleq \nabla \mathbf{L}(\mathbf{x}) \in \mathbb{R}^{d \times M}} \mathbf{w} \right\|_2^2 \quad \text{s.t.} \quad \mathbf{w} \geq \mathbf{0},\ \mathbf{w}^\top \mathbf{1} = 1$$

$$\approx \text{Proj}_{\mathcal{S}_M} \left(\mathbf{w} - \eta_w \nabla \mathbf{L}(\mathbf{x})^\top \nabla \mathbf{L}(\mathbf{x}) \mathbf{w} \right) \qquad \text{(Single PGD step)}$$

$\mathcal{S}_M$ is the probability simplex in $\mathbb{R}^M$

Server uses $G \approx \nabla \mathbf{L}(\mathbf{x})^\top \nabla \mathbf{L}(\mathbf{x})$ (obtained with $\mathcal{O}(d)$ communication) to compute

$$\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M} \left(\mathbf{w} - \eta_w G \mathbf{w} \right)$$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$
- Update the global model
$$\mathbf{x} \leftarrow \mathbf{x} - \eta_s \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \nabla^{(i)}$$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$
- Update the global model
$$\mathbf{x} \leftarrow \mathbf{x} - \eta_s \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \nabla^{(i)}$$

Server Communication Cost

- FMGDA - d

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$
- Update the global model
$$\mathbf{x} \leftarrow \mathbf{x} - \eta_s \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \nabla^{(i)}$$

Server Communication Cost

- FMGDA - d
- FedCMOO - $d + M \approx d$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$
- Update the global model
$$\mathbf{x} \leftarrow \mathbf{x} - \eta_s \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \nabla^{(i)}$$

Server Communication Cost

- FMGDA - d
- FedCMOO - $d + M \approx d$

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$
- Update the global model
$$\mathbf{x} \leftarrow \mathbf{x} - \eta_s \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \nabla^{(i)}$$

Server Communication Cost

- FMGDA - d
- FedCMOO - $d + M \approx d$

Per-client Communication Cost

- FMGDA - Md

FedCMOO: Federated Communication-efficient MOO

Server and client steps

- Select client subset $\mathcal{C}$ - send $\mathbf{x}$
- Compute $\widetilde{\nabla}\mathbf{L}_i(\mathbf{x})$ and send $\widehat{H}_i = \mathcal{Q}(\widetilde{\nabla}\mathbf{L}_i(\mathbf{x}))$ to server
- Reshape $\widehat{H}_i$ to H_i. Compute $G \approx \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)^\top \left(\frac{1}{|\mathcal{C}|}\sum_i H_i\right)$
- Compute $\mathbf{w} \leftarrow \text{Proj}_{\mathcal{S}_M}(\mathbf{w} - \eta_w G\mathbf{w})$ and send to clients
- Start with $\mathbf{x}$, multiple (τ) local updates
$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} - \eta_c \sum_k w_k \widetilde{\nabla} L_k^{(i)}(\mathbf{x}^{(i)})$$
- Return updates to server
$$\nabla^{(i)} = \mathbf{x} - \mathbf{x}^{(i)}$$
- Update the global model
$$\mathbf{x} \leftarrow \mathbf{x} - \eta_s \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \nabla^{(i)}$$

Server Communication Cost

- FMGDA - d
- FedCMOO - $d + M \approx d$

Per-client Communication Cost

- FMGDA - Md
- FedCMOO - $2d$

Theory: Assumptions

Problem

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

Theory: Assumptions

Problem

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- **Smoothness:** $\|\nabla L_k^{(i)}(\mathbf{x}) - \nabla L_k^{(i)}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$

Theory: Assumptions

Problem

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- **Smoothness:** $\|\nabla L_k^{(i)}(\mathbf{x}) - \nabla L_k^{(i)}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$
- **Unbiased** stochastic gradients with **bounded variance**

Theory: Assumptions

Problem

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- **Smoothness:** $\|\nabla L_k^{(i)}(\mathbf{x}) - \nabla L_k^{(i)}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$
- **Unbiased** stochastic gradients with **bounded variance**
- **Bounded Gradients:** $\|\nabla L_k^{(i)}(\mathbf{x})\| \leq B$

Theory: Assumptions

Problem

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- **Smoothness:** $\|\nabla L_k^{(i)}(\mathbf{x}) - \nabla L_k^{(i)}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$
- **Unbiased** stochastic gradients with **bounded variance**
- **Bounded Gradients:** $\|\nabla L_k^{(i)}(\mathbf{x})\| \leq B$
- **Unbiased Compression:** $\mathbb{E}[Q(\mathbf{x})] = \mathbf{x}$ and $\mathbb{E}\|Q(\mathbf{x}) - \mathbf{x}\|^2 \leq q\|\mathbf{x}\|^2$, for some $q > 0$

Theoretical Result: Convergence

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$$\min_{[T]} \| \sum_k w_k \nabla L_k(\mathbf{x}) \|^2 \leq \underbrace{\mathcal{O}\left(\frac{1}{T\eta_s\eta_c\tau} + \frac{L\eta_s\eta_c}{n} \right)}_{\substack{\text{Centralized Optimization Error} \\ \text{for Scalarized Loss}}} + \underbrace{\mathcal{O}\left(L\eta_s\eta_c\tau B \right)}_{\substack{\text{Partial Participation} \\ \text{Error}}}$$

$$+ \underbrace{\mathcal{O}\left(L\eta_c(\tau B + \sqrt{\tau} B) \right)}_{\text{Local Drift Error}} + \underbrace{\mathcal{O}\left(\eta_w M \left[\frac{q}{n} + \frac{qB}{n} + B \right]^2 \right)}_{\text{MOO Weight Error}}$$

Theoretical Result: Convergence

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

$$\min_{[T]} \| \sum_k w_k \nabla L_k(\mathbf{x}) \|^2 \leq \underbrace{\mathcal{O}\left(\frac{1}{T\eta_s\eta_c\tau} + \frac{L\eta_s\eta_c}{n} \right)}_{\substack{\text{Centralized Optimization Error} \\ \text{for Scalarized Loss}}} + \underbrace{\mathcal{O}\left(L\eta_s\eta_c\tau B \right)}_{\substack{\text{Partial Participation} \\ \text{Error}}}$$

$$+ \underbrace{\mathcal{O}\left(L\eta_c(\tau B + \sqrt{\tau} B) \right)}_{\text{Local Drift Error}} + \underbrace{\mathcal{O}\left(\eta_w M \left[\frac{q}{n} + \frac{qB}{n} + B \right]^2 \right)}_{\text{MOO Weight Error}}$$

- With appropriate choice of parameters η_s, η_c, η_w

$$\min_{t \in [T]} \| \sum_k w_k(t) \nabla L_k(\mathbf{x}(t)) \|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{T}} \right)$$

Dependence on Number of Objectives M

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

To achieve $\min_{t \in [T]} \| \sum_k w_k(t) \nabla L_k(\mathbf{x}(t)) \|^2 \leq \epsilon$

---

*Xiao, et al. "Direction-oriented multi-objective learning: Simple and provable stochastic algorithms." *NeurIPS'23*.

†Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Dependence on Number of Objectives M

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

To achieve $\min_{t \in [T]} \| \sum_k w_k(t) \nabla L_k(\mathbf{x}(t)) \|^2 \leq \epsilon$

Work	Federated	Complexity
SDMGrad*	✗	$\mathcal{O}\left(\frac{M^2}{\epsilon^2}\right)$

*Xiao, et al. "Direction-oriented multi-objective learning: Simple and provable stochastic algorithms." *NeurIPS'23*.

†Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Dependence on Number of Objectives M

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

To achieve $\min_{t \in [T]} \| \sum_k w_k(t) \nabla L_k(\mathbf{x}(t)) \|^2 \leq \epsilon$

Work	Federated	Complexity
SDMGrad*	✗	$\mathcal{O}\left(\frac{M^2}{\epsilon^2}\right)$
FMGDA†	✓	$\mathcal{O}\left(\frac{M^4}{\epsilon^2}\right)$

*Xiao, et al. "Direction-oriented multi-objective learning: Simple and provable stochastic algorithms." *NeurIPS'23*.

†Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Dependence on Number of Objectives M

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

To achieve $\min_{t \in [T]} \| \sum_k w_k(t) \nabla L_k(\mathbf{x}(t)) \|^2 \leq \epsilon$

Work	Federated	Complexity
SDMGrad*	✗	$\mathcal{O}\left(\frac{M^2}{\epsilon^2}\right)$
FMGDA†	✓	$\mathcal{O}\left(\frac{M^4}{\epsilon^2}\right)$
FedCMOO (Ours)	✓	$\mathcal{O}\left(\frac{M}{\epsilon^2}\right)$

*Xiao, et al. "Direction-oriented multi-objective learning: Simple and provable stochastic algorithms." *NeurIPS'23*.

†Yang, et al. "Federated Multi-Objective Learning," *NeurIPS'23*.

Limitations of Our Analysis

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- With appropriate choice of parameters η_s, η_c, η_w

$$\min_{[T]} \| \sum_k w_k \nabla L_k(\mathbf{x}) \|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$$

Limitations of Our Analysis

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- With appropriate choice of parameters η_s, η_c, η_w

$$\min_{[T]} \| \sum_k w_k \nabla L_k(\mathbf{x}) \|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$$

- **Bounded Gradients:** $\|\nabla L_k^{(i)}(\mathbf{x})\| \leq B$

Limitations of Our Analysis

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- With appropriate choice of parameters η_s, η_c, η_w

$$\min_{[T]} \| \sum_k w_k \nabla L_k(\mathbf{x}) \|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$$

- **Bounded Gradients:** $\|\nabla L_k^{(i)}(\mathbf{x})\| \leq B$
- No provable benefit of parallelization (i.e., speedup in N)

Limitations of Our Analysis

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

- With appropriate choice of parameters η_s, η_c, η_w

$$\min_{[T]} \| \sum_k w_k \nabla L_k(\mathbf{x}) \|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$$

- **Bounded Gradients:** $\|\nabla L_k^{(i)}(\mathbf{x})\| \leq B$
- No provable benefit of parallelization (i.e., speedup in N)
  - **Culprit:** the inner product terms

Outline

1. Background: MOO and MGDA
2. Federated MGDA and its Drawbacks
3. FedCMOO: Improving Federated MGDA
4. Experiment Results
5. Conclusion

Experiment Results

Experiments: Datasets

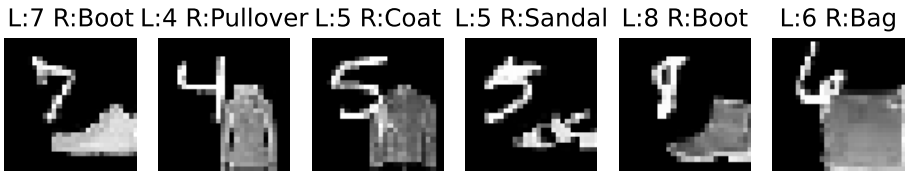
MNIST+FMNIST, MultiMNIST, CIFAR10+MNIST

- Constructed by combining two randomly sampled images: one from each dataset
- 2 objectives

Experiments: Datasets

MNIST+FMNIST, MultiMNIST, CIFAR10+MNIST

- Constructed by combining two randomly sampled images: one from each dataset
- 2 objectives

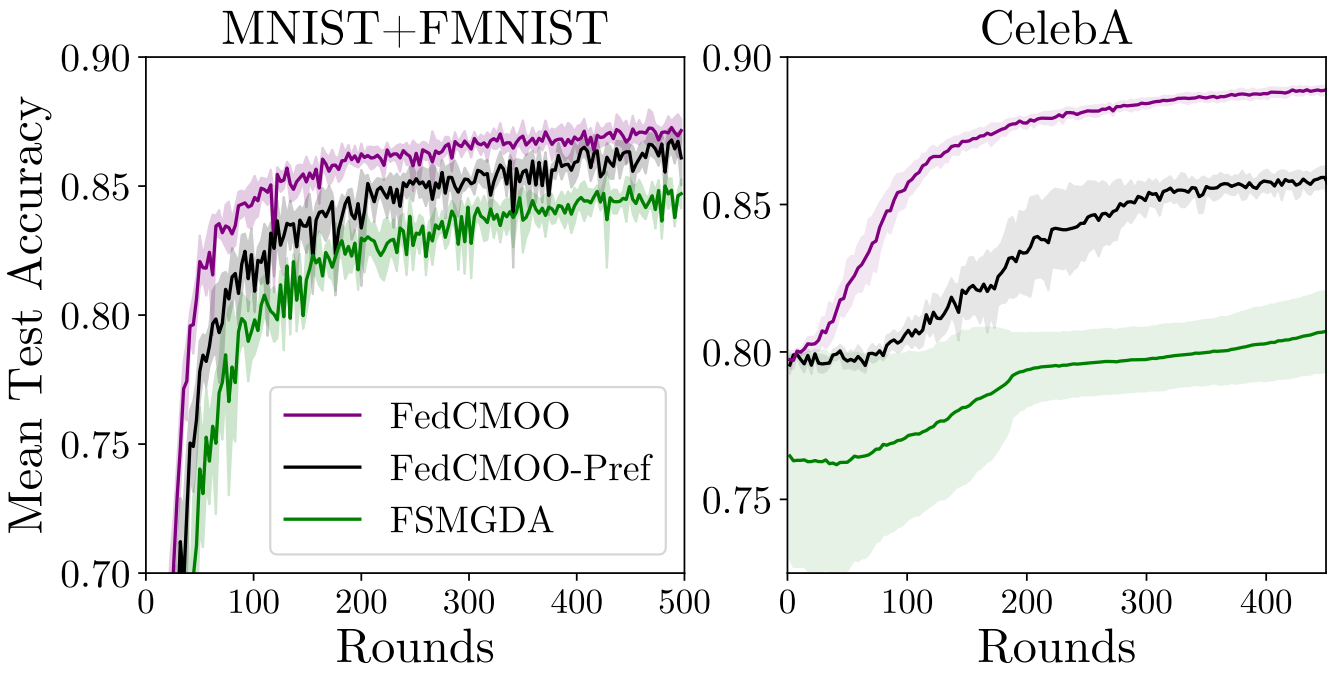


CelebA

- CelebA dataset - each image has 40 attributes
- Each attribute gives a binary classification $\Rightarrow$ 40 objectives

Experiments: Mean Test Accuracy

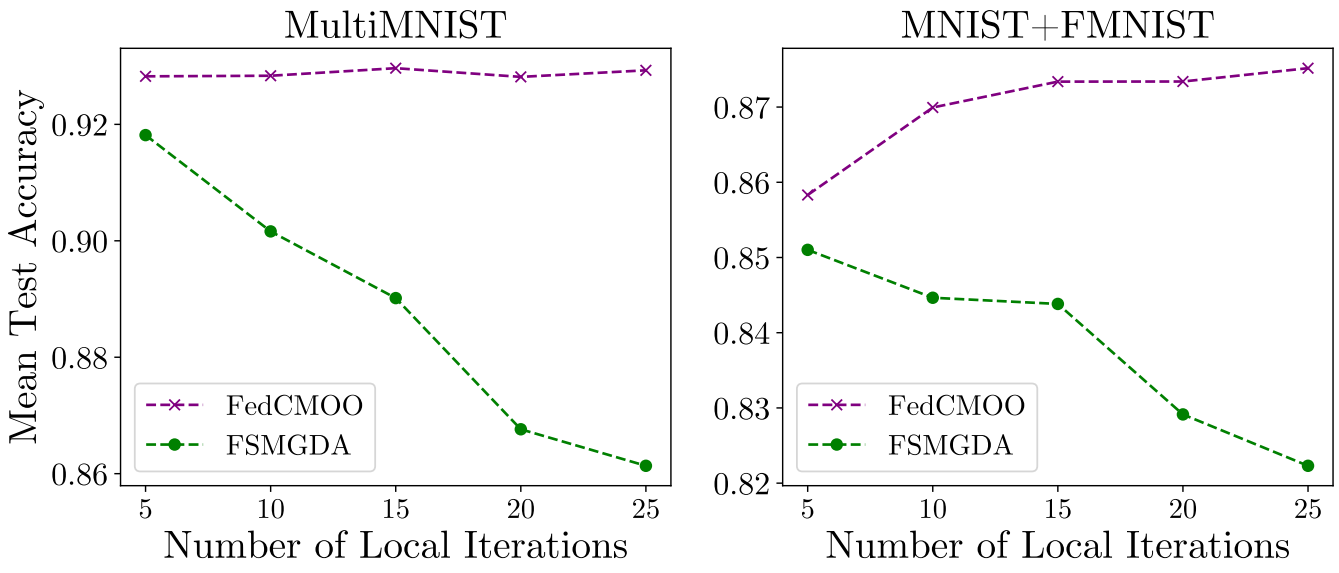
100 clients in total, 10 selected in each round



Test Accuracy averaged over all the objectives

Experiments: Controlling Local Drift

100 clients in total, 10 selected in each round

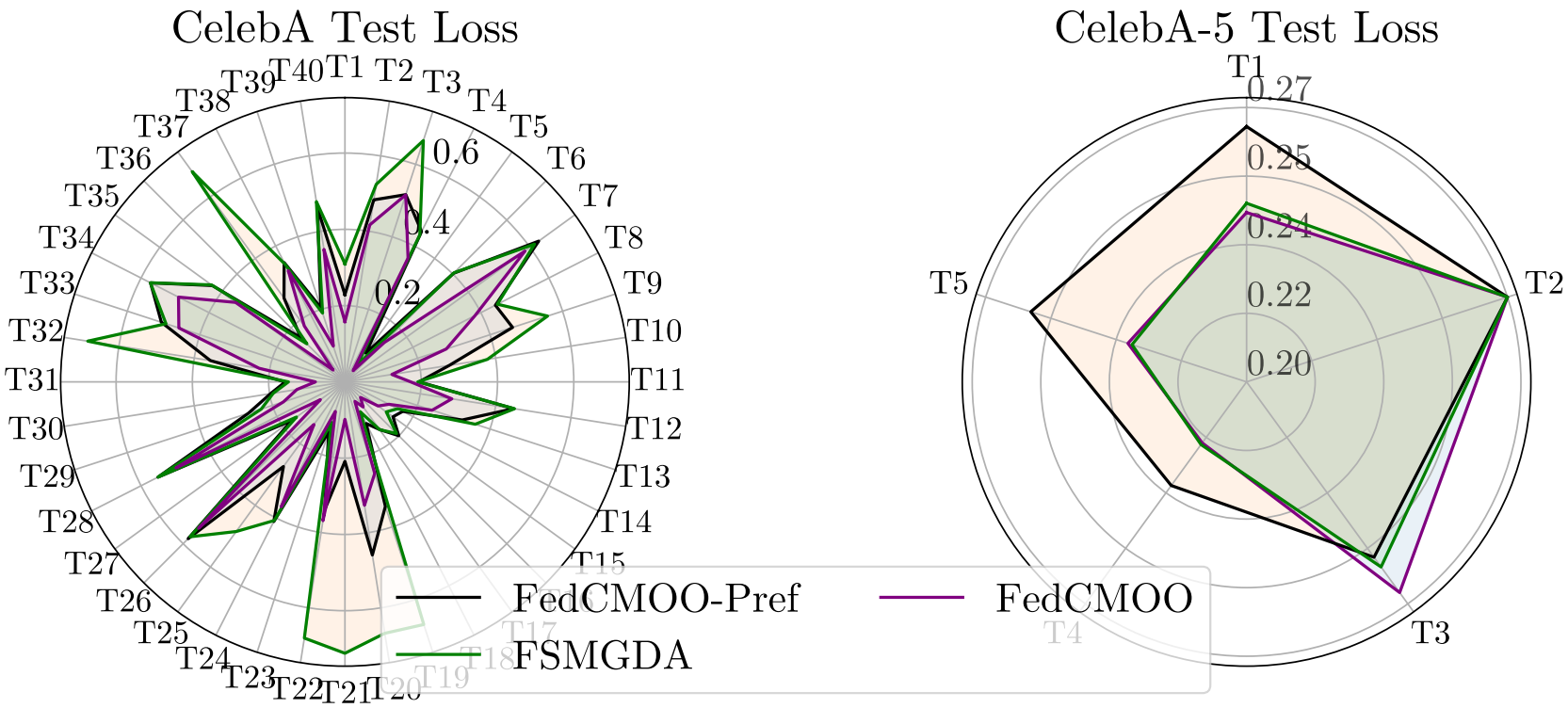


Test Accuracy averaged over all the objectives

- Increasing local iterations (τ) $\Rightarrow$ larger local drift in FSMGDA $\Rightarrow$ worse performance

Experiments: Final Loss Values

- CelebA dataset - 40 objectives



Preferences in MOO

- Users sometimes prefer solutions with specific trade-offs among the different objectives
  - Healthcare
  - Privacy $\leftrightarrow$ Utility
- One possible solution: specific ratios for loss values

$$\min_{\mathbf{x}\in\mathbb{R}^d} \mathbf{L}(\mathbf{x}) := [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top$$
$$\text{subject to} \quad r_1 L_1(\mathbf{x}) = \cdots = r_M L_M(\mathbf{x}).$$

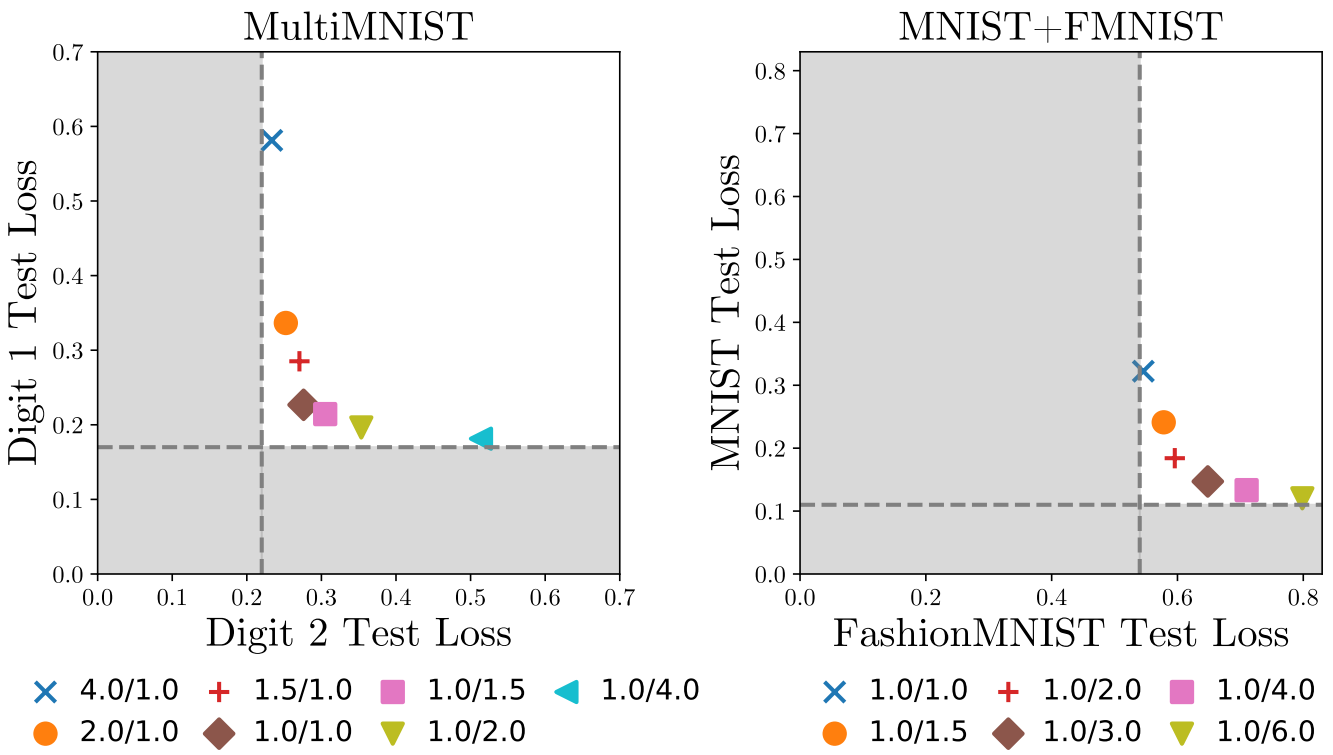
- MOO methods, by default, do not let us control the trade-off
- Ensure that normalized scaled losses are almost uniform*

$$\hat{u}_k(\mathbf{r}) = \frac{r_k L_k(\mathbf{x})}{\sum_j r_j L_j(\mathbf{x})} \approx \frac{1}{M}$$

---

*Mahapatra and Rajan, "Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization." *ICML'20*.

Experiments: Preference-based Pareto Solutions



- Vertical and horizontal dashed lines - minimum loss from single-objective training
- Gray regions are infeasible
- More difficult to satisfy preferences with significantly different weights

Conclusion

Some Open Questions

- How to explore the entire Pareto front?
- Generalization guarantees?
- Impact of MOO optimization on MTL generalization[§]
  - Generalization gap between single-task and multi-task trajectories early into training
  - Standard optimization metrics (sharpness, minimum loss, etc.) do not explain the generalization gaps between single-task and multi-task models

[§]Mueller, et al. "Can Optimization Trajectories Explain Multi-Task Transfer?" *rXiv:2408.14677*.

Summary: Federated Multi-Objective Optimization (MOO)

$$\min_{\mathbf{x}} \mathbf{L}(\mathbf{x}) := \min_{\mathbf{x}} [L_1(\mathbf{x}), \ldots, L_M(\mathbf{x})]^\top \quad \text{where} \quad L_k(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} L_k^{(i)}(\mathbf{x})$$

M objectives, N clients

- Communication-efficient algorithm FedCMOO
  - Communication cost does not scale with the number of objectives M
- Convergence to Pareto-stationary point
  - Improved sample complexity in terms of M
  - Under weaker assumptions
- Improved empirical performance

Thanks! (pranaysh@iitb.ac.in)