Poster: Optimal Variance-Reduced Client Sampling for Multiple Models Federated Learning

Haoran Zhang, Zekai Li, Zejun Gong, Marie Siew, Carlee Joe-Wong, Rachid El-Azouzi
Contact: haoranz5@andrew.cmu.edu

Carnegie Mellon University
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN
AVIGNON UNIVERSITÉ

Background: Federated Learning

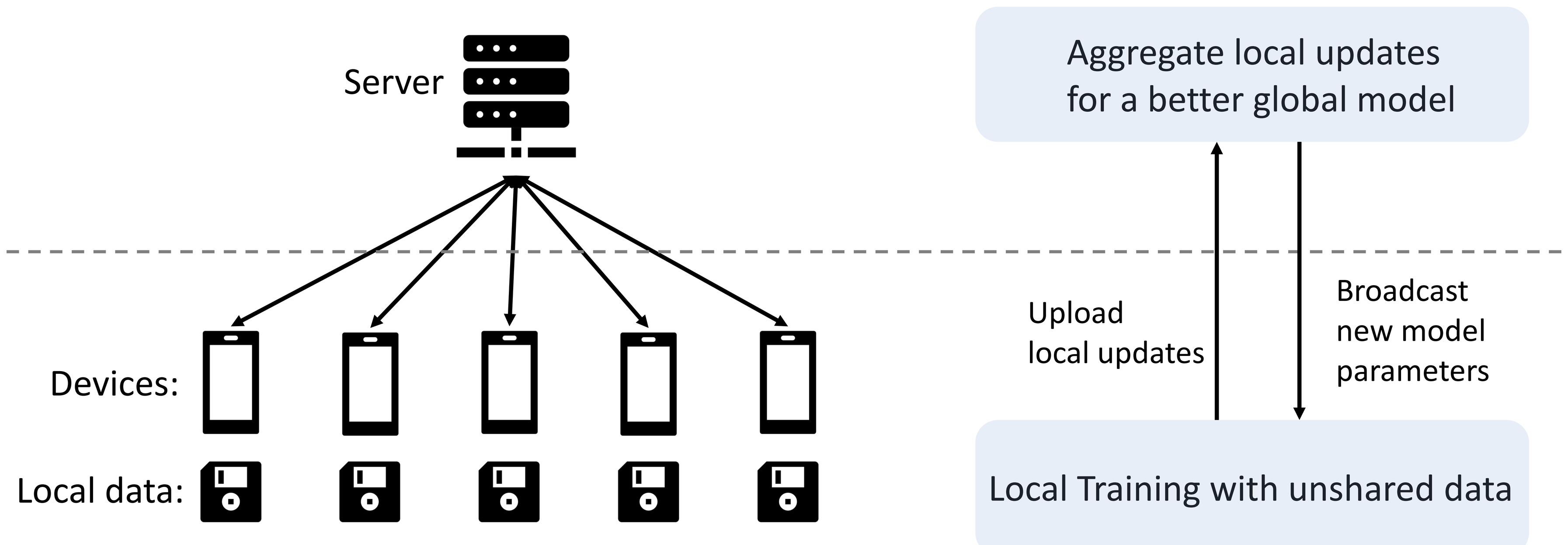

Figure 1. The process of federated learning.

- Federated learning (FL) is a technique that trains a **single** deep learning model across multiple edge devices without sharing their own datasets **(benefits in data privacy)**.
- Example Applications: **Google keyboard prediction, voice assistant on your phone, smart home IoT networks, healthcare applications.**

Multiple Models Federated Learning (MMFL)

Motivation for Multiple Models Federated Learning (MMFL)

- There could be multiple FL models running on an edge device [1].
- Example: smartphone training multiple FL models (Figure 2).

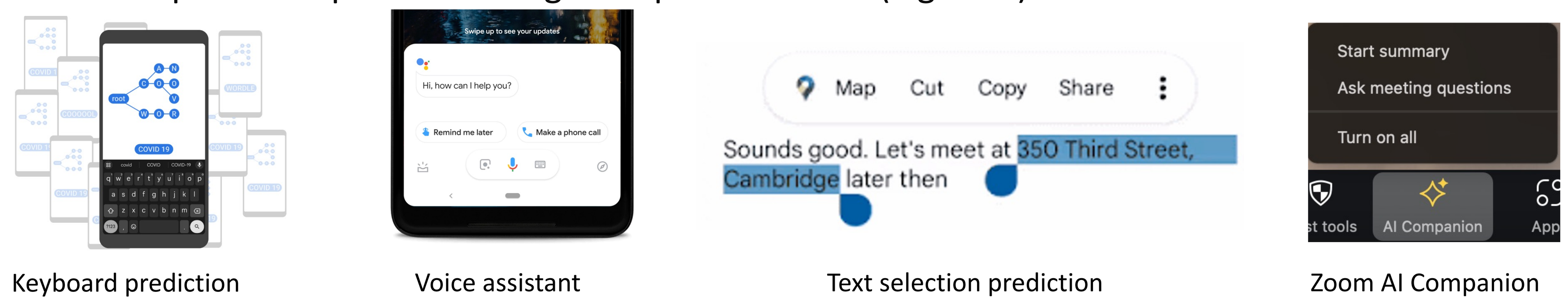

Figure 2: Real applications of federated learning

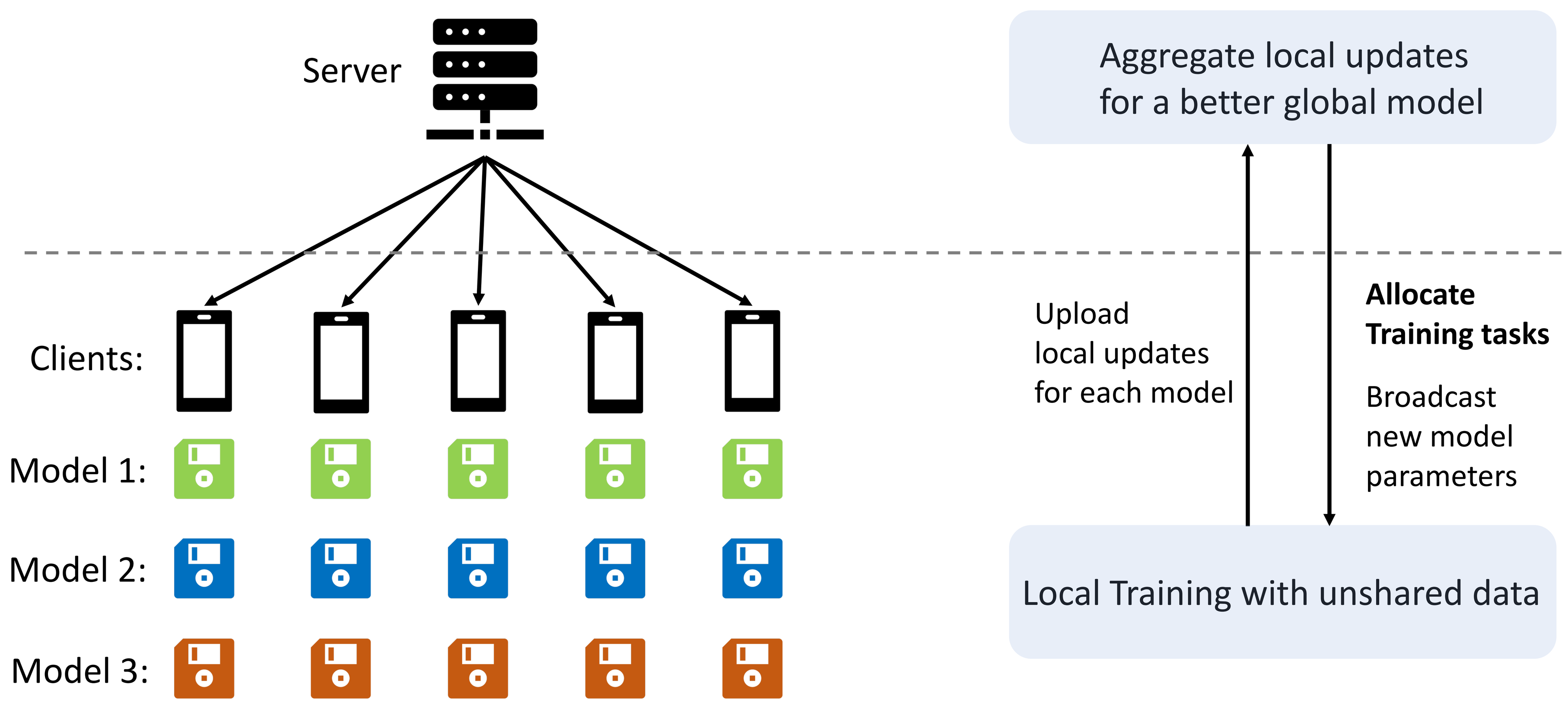

Figure 3: The process of multiple models federated learning (MMFL).

- In MMFL, N clients jointly train S models. MMFL objective: $\min_{w_1,\cdots,w_S} \sum_{s=1}^{S}\sum_{i=1}^{N} d_{i,s} f_{i,s}(w_s)$
  $d_{i,s}$: dataset size ratio of client i for model s, $\sum_{i=1}^{N} d_{i,s} = 1$
  $f_{i,s}(w_s)$: loss function for model weights w_s given client i's local data.

A Potential Challenge in MMFL System: **Communication Cost**

Example: A company (server) holds 1 million users (clients) training 5 LLM models for different downstream tasks in FL.

Full Participation: server receives 5 million model updates per round (too expensive).

Assumptions:

- **Server-side communication:** Server has limited parallel processing ability, therefore, conducts partial communication / participation (for example, active rate=0.1).
- **Client-side communication:** Each client can only afford sending 1 model's update, considering they may handle large models like LLM.

With assumptions 1 and 2: server only receives 100k model updates per round.

How to sample clients? How to allocate models (training tasks)?

Overview of the Proposed Algorithm

In each round

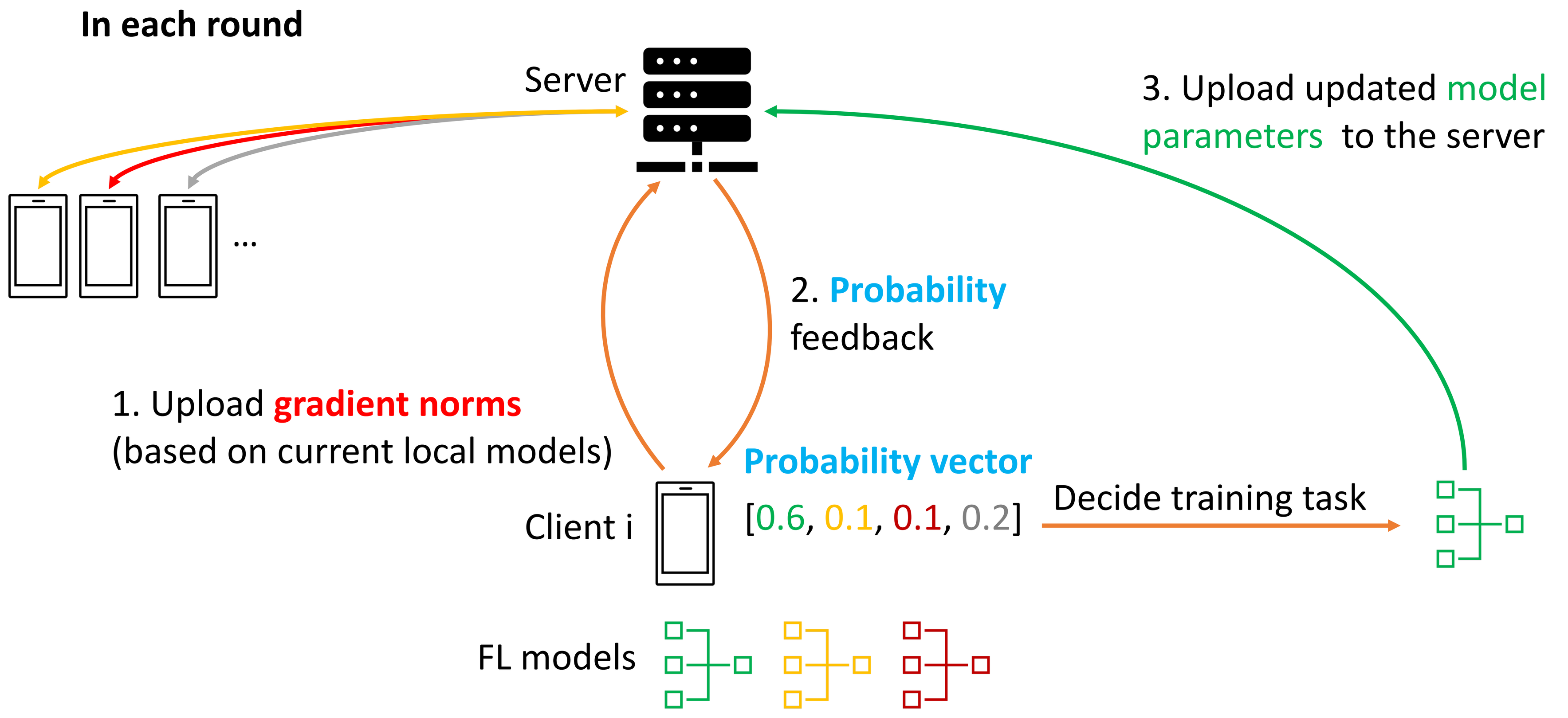

Figure 4: The process of the proposed algorithm.

How to optimize the sampling probability distribution?

Variance-Reduced Optimal Client Sampling

Improve the MMFL system with communication constraints. The method can achieve faster and more stable convergence.

Global Update Rule (Aggregation) for unbiased training with modified sampling distribution

$$w_s^{\tau+1} = w_s^{\tau} - \eta_\tau \sum_{i\in\mathcal{A}_{\tau,s}} \frac{d_{i,s}}{p_{s|i}} U_{i,s} \qquad U_{i,s}^{\tau} = \sum_{t=1}^{E} \nabla f_{i,s}(w_{i,s,\tau}^{t})$$

Client i update (gradient)

τ: global round number
i: client index
s: model index
m: expected number of active clients
$d_{i,s}$: dataset size ratio
t: local epoch number
$\mathcal{A}_{\tau,s}$: set of active clients

Why unbiased:

$$\mathbb{E}_{\mathcal{A}_{\tau,s}}\left[\sum_{i=1}^{N} \mathbb{1}_{i\in\mathcal{A}_{\tau,s}} \frac{d_{i,s}}{p_{s|i}^{\tau}} U_{i,s}^{\tau}\right] = \sum_{i=1}^{N} \mathbb{E}_{\mathcal{A}_{\tau,s}}\left[\mathbb{1}_{i\in\mathcal{A}_{\tau,s}}\right] \frac{d_{i,s}}{p_{s|i}^{\tau}} U_{i,s}^{\tau} = \sum_{i=1}^{N} d_{i,s} U_{i,s}^{\tau}$$

Optimize the probability distribution:

Sampling Probability distribution — Sampled Update — Full participation update

$$\min_{\{p_{s|i}^{\tau}\}} \sum_{s=1}^{S} \mathbb{E}_{\mathcal{A}_{\tau,s}}\left[\left\| \sum_{i\in\mathcal{A}_{\tau,s}} \frac{d_{i,s}}{p_{s|i}^{\tau}} U_{i,s}^{\tau} - \sum_{i=1}^{N} d_{i,s} U_{i,s}^{\tau} \right\|^2\right]$$

$$\text{s.t. } p_{s|i}^{\tau} \geq 0,\ \sum_{s=1}^{S} p_{s|i}^{\tau} \leq 1,\ \sum_{s=1}^{S}\sum_{i=1}^{N} p_{s|i}^{\tau} = m \quad \forall i, s$$

Closed-form solution (higher gradient-norm->higher sampling probability)

$$p_{s|i}^{\tau} = \begin{cases} (m-N+k)\frac{\|\tilde{U}_{i,s}^{\tau}\|}{\sum_{j=1}^{k} M_j^{\tau}} & \text{if } i = 1,2,\cdots,k, \\ \frac{\|\tilde{U}_{i,s}^{\tau}\|}{M_i^{\tau}} & \text{if } i = k+1,\cdots,N. \end{cases} \quad (5)$$

where $\|\tilde{U}_{i,s}^{\tau}\| = \|d_{i,s}U_{i,s}^{\tau}\|$ and $M_i^{\tau} = \sum_{s=1}^{S}\|\tilde{U}_{i,s}^{\tau}\|$. We reorder clients such that $M_i^{\tau} \leq M_{i+1}^{\tau}$ for all i, and k is the largest integer for which $0 < (m-N+k) \leq \frac{\sum_{j=1}^{k} M_j^{\tau}}{M_k^{\tau}}$.

Server only requests gradient norms from all clients to generate sampling probability distribution. Clients decide if they can send updates to the server based on this distribution.

The benefits of minimizing the variance of sampled update:

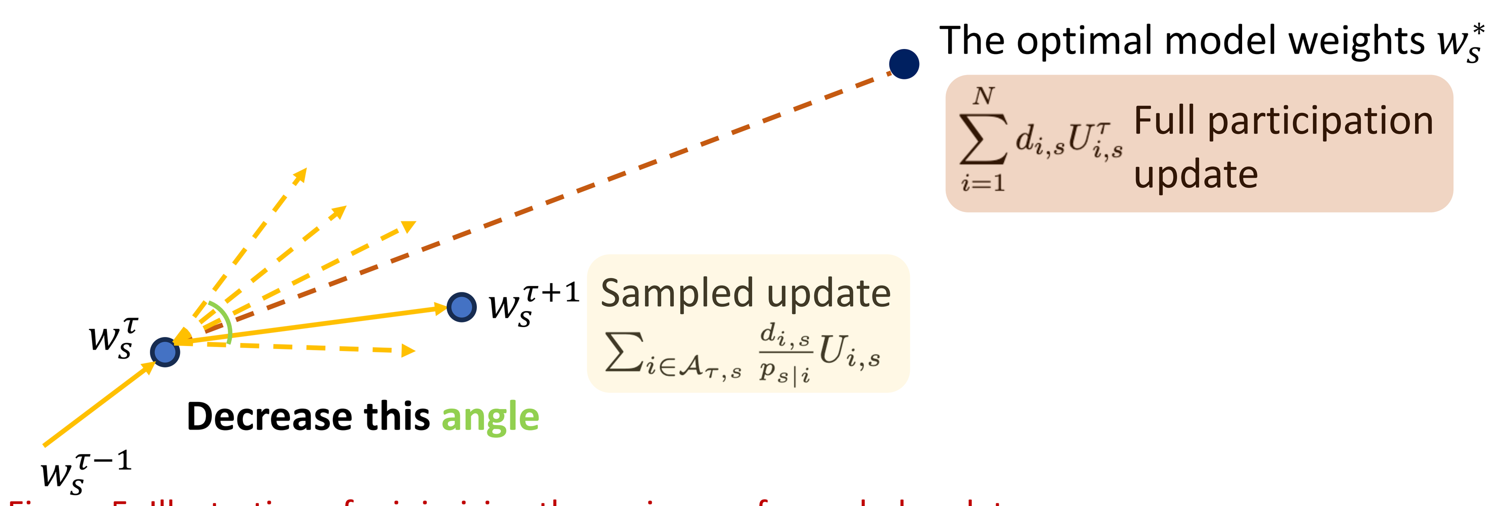

Figure 5: Illustration of minimizing the variance of sampled update.

Evaluation Results

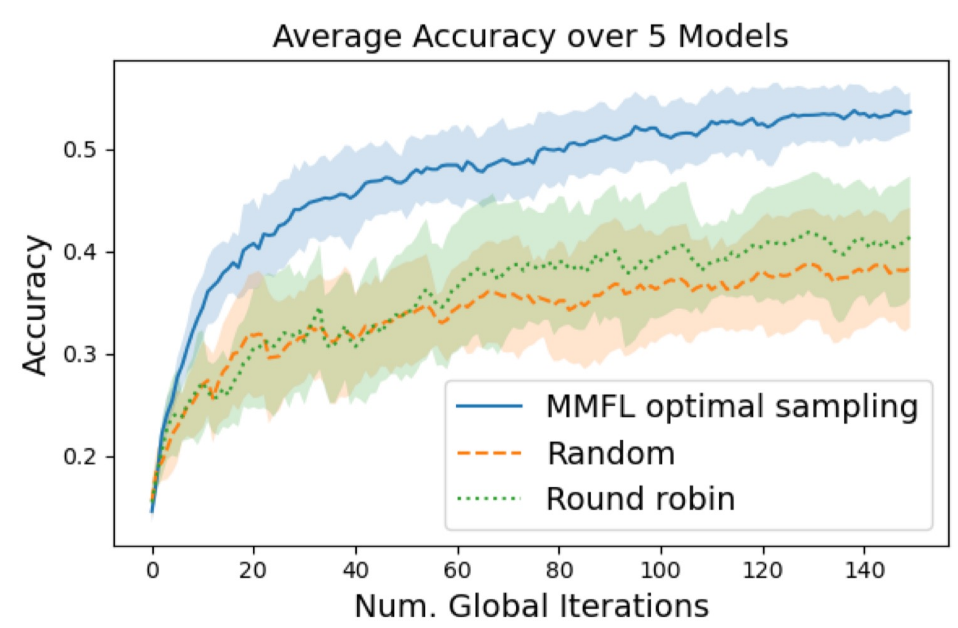

Figure 6: Experiment results: average accuracy across multiple models

- We evaluate our proposed algorithm using an MMFL setting, with 120 clients training 5 models with different non-iid levels on local datasets (Fashion-MNIST).
- **Partial participation ratio: 10%**
- Dataset details: Each client receives data from 30% labels of the total in model 1,2,3, and 40% labels of the total in model 4,5. 10% clients possess 52.6% data to simulate data heterogeneity in real world.

- Our algorithm achieves an average accuracy across multiple models that is **over 30% higher** compared to baseline methods.
- The sampled update is unbiased and can be viewed as an estimator of the full update.
- In partial participation MMFL, the variance of the sampled update can be large, leading to less accurate global updates. Our method minimizes this variance, resulting in faster and more stable convergence. Local updates with high norms dominate the direction of the full update.
- Limitations: The method requires the gradient norms between different models to be similar in scale. Some normalization methods may be helpful when incorporating models with very different gradient norm scales.

Selected References and Acknowledgments

[1] Bhuyan, N., Moharir, S. and Joshi, G., 2022. Multi-Model Federated Learning with Provable Guarantees. arXiv preprint arXiv:2207.04330.
[2] Chen, Wenlin, Samuel Horváth, and Peter Richtárik. "Optimal Client Sampling for Federated Learning." Transactions on Machine Learning Research.

This work is supported by NSF CNS-2106891 and ANR-22-CE23-0024. M. Siew is supported by the SUTD Faculty Early Career Award.