Federated Learning Can Find Friends That Are Advantageous and Help with Low-Resource Machine Translation

Eduard Gorbunov

MBZUAI

NEO Seminar, Inria Center @ Université Côte d'Azur

November 26, 2024

Short Bio

Me in Dubai (October 2023)

- Research Scientist at MBZUAI (Abu Dhabi, UAE) hosted by Samuel Horváth and Martin Takáč (from April 2024)
- PhD in Computer Science, MIPT (2020-2021), Supervisors: Alexander Gasnikov and Peter Richtárik
- Research interests: Stochastic **Optimization**, Distributed **Optimization**, Variational Inequalities, Derivative-Free **Optimization**
- Selected awards: Ilya Segalovich Award 2019 (highly selective), best reviewer award (ICLR 2021, ICML 2021-2022, NeurIPS 2020-2022)

My website → (eduardgorbunov.github.io)

Outline

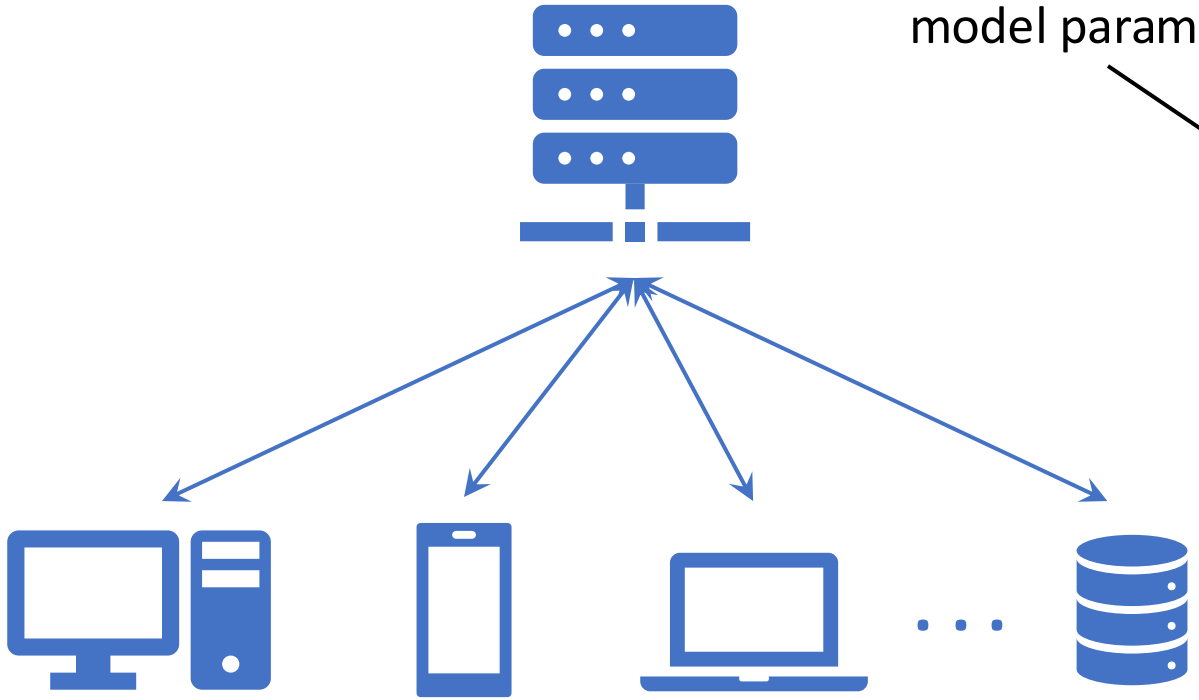
1. Federated Learning

2. Federated Learning with a Target Client

3. Low-Resource Machine Translation

Federated Learning

The Problem

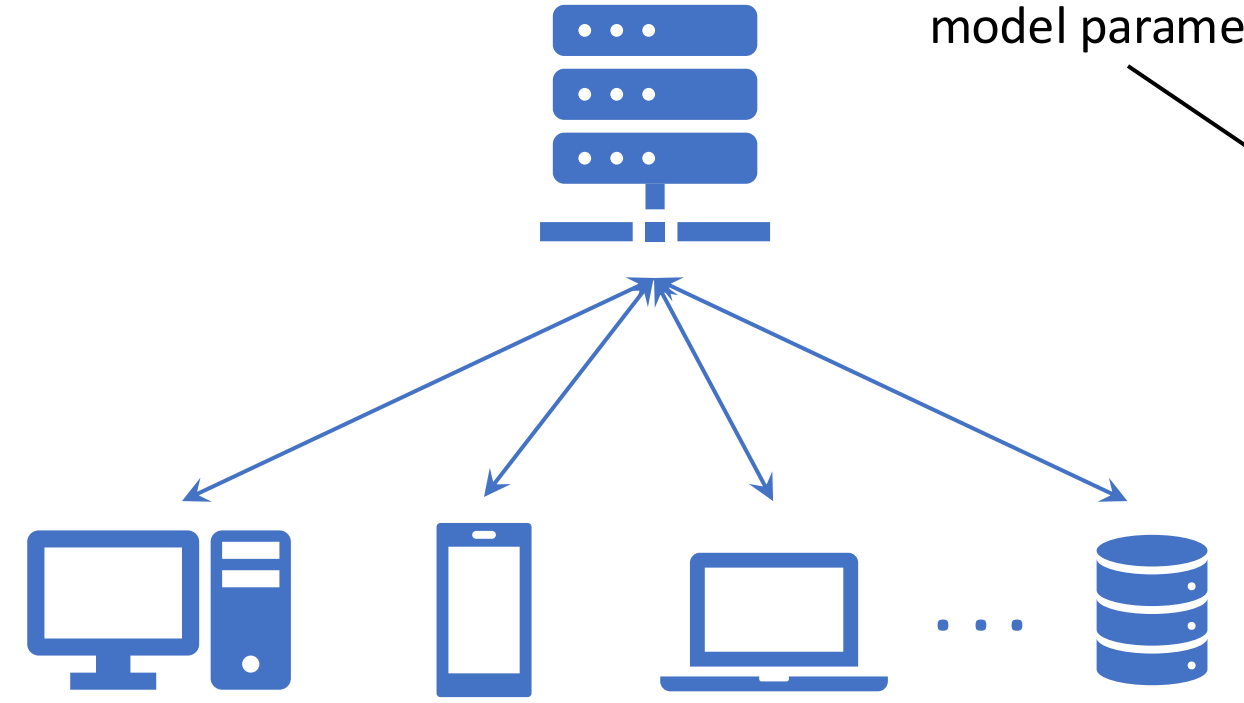


model parameters

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

# of parameters

The Problem



n workers/clients

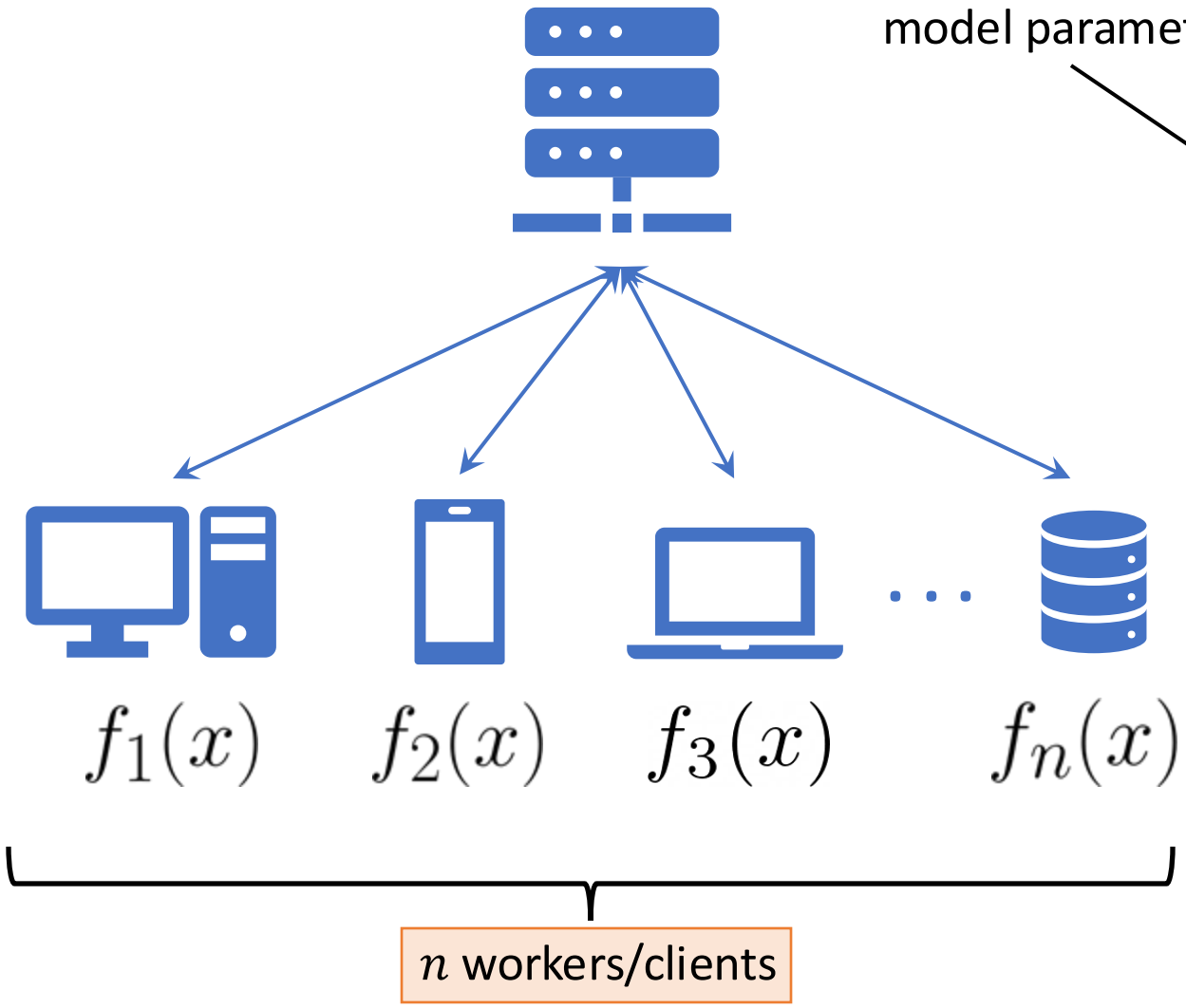
model parameters

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

# of parameters

# of workers/clients

The Problem



model parameters

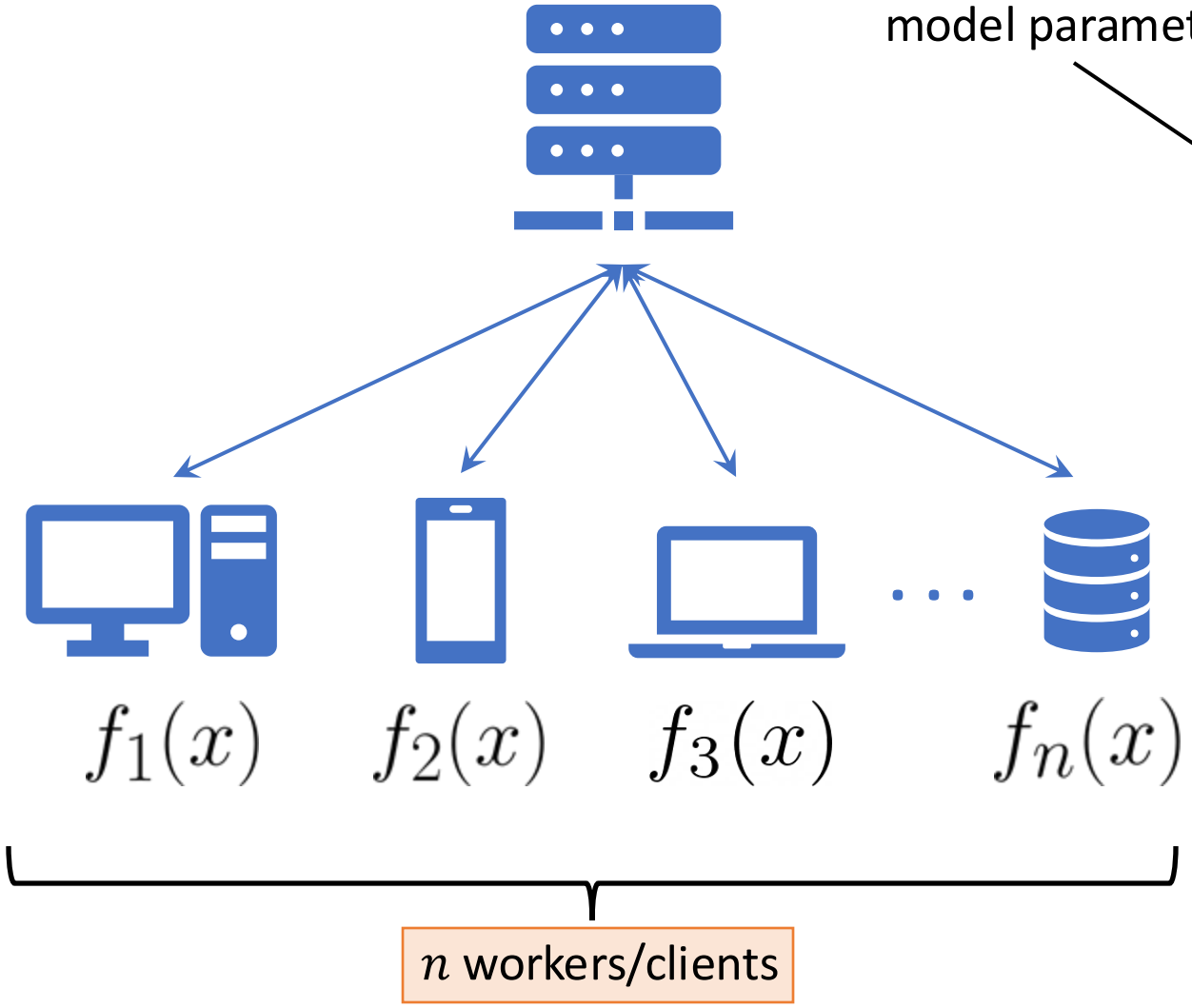
$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

# of parameters

# of workers/clients

loss on the data distribution on worker i

The Problem



model parameters

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

# of parameters

# of workers/clients

loss on the data distribution on worker i

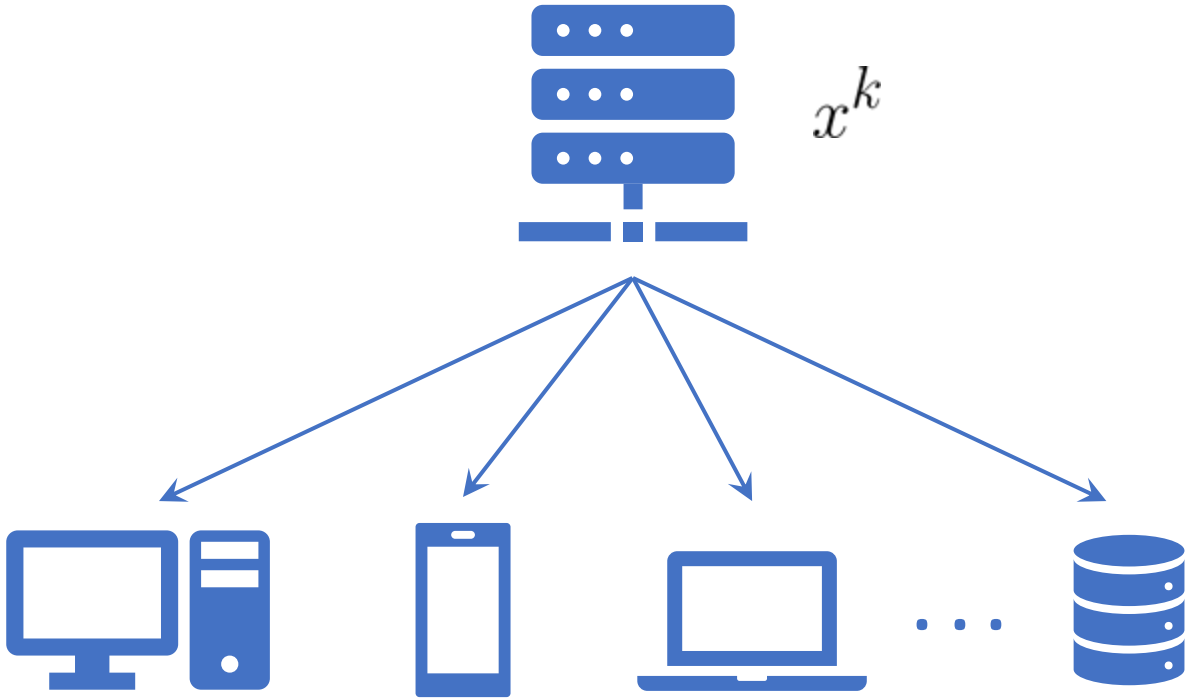
Key features:

- Data is heterogeneous among clients
- Data is private

Parallel SGD

Iteration k:

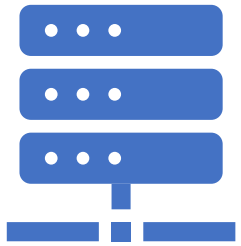
1. Server broadcasts x^k



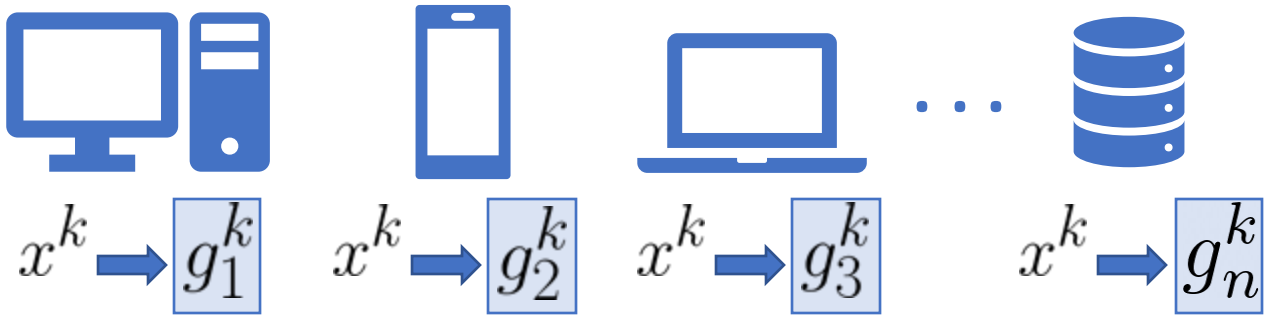
Parallel SGD

Iteration k:

1. Server broadcasts x^k
2. Workers compute stochastic gradients



x^k

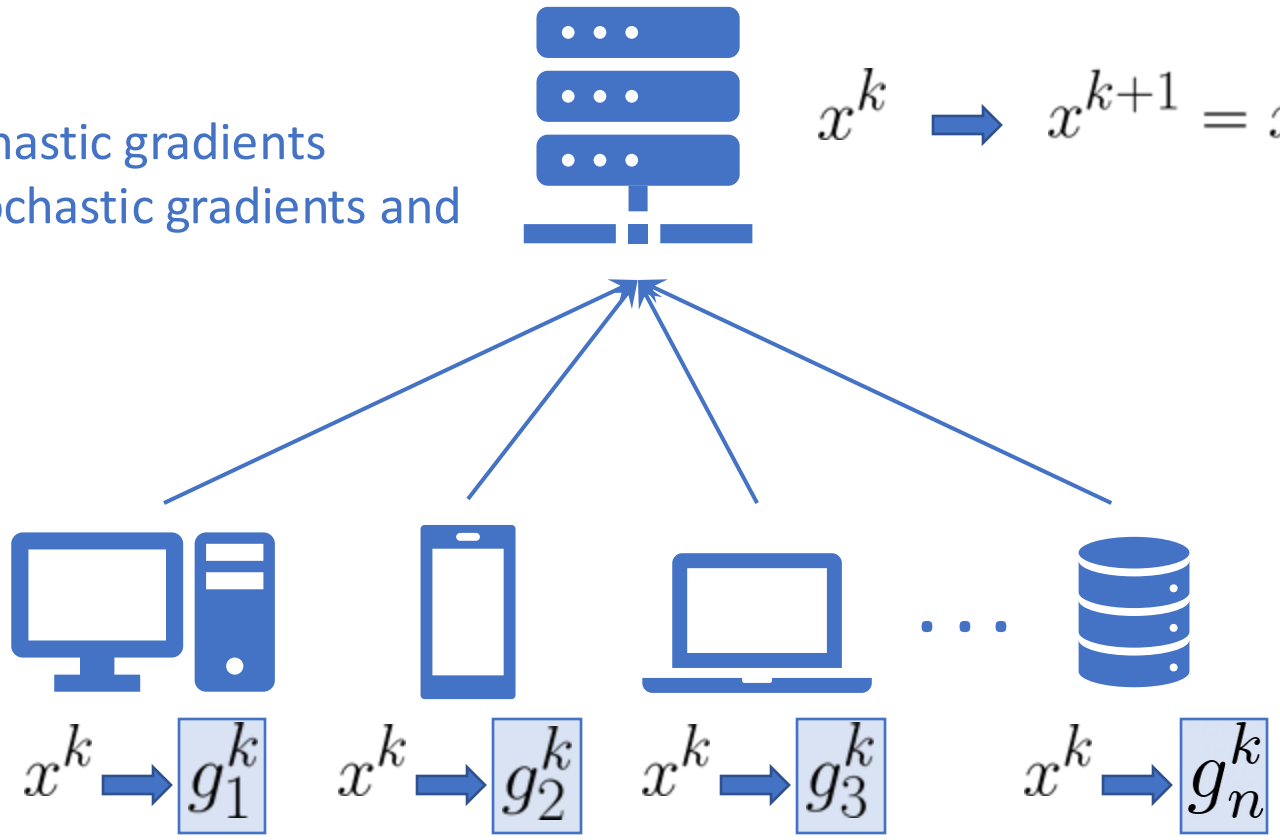


$$\mathbb{E}_k[g_i^k] = \nabla f_i(x^k)$$

Parallel SGD

Iteration k:

1. Server broadcasts x^k
2. Workers compute stochastic gradients
3. Server averages the stochastic gradients and makes an SGD step



$$x^k \rightarrow x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k$$

$$\mathbb{E}_k[g_i^k] = \nabla f_i(x^k)$$

Parallel SGD: (Some) Good Properties

- ✅ Simple method
- ✅ Provable convergence
- ✅ Cheap iterations
- ✅ Can be improved in terms of communication efficiency

Federated Learning with a Target Client

N. Tupitsa, S. Horváth, M. Takáč, E. Gorbunov. *Federated Learning Can Find Friends That Are Advantageous* (arXiv:2402.05050)

Nazarii Tupitsa
Research Assistant
MBZUAI

Samuel Horváth
Assistant Professor
MBZUAI

Martin Takáč
Associate Professor
MBZUAI

Federated Learning with a Target Client

Standard Federated Learning

Goal: find $\hat{x}$ such that

$$\hat{x} \approx \arg\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

Federated Learning with a Target Client

Standard Federated Learning

Goal: find $\hat{x}$ such that

$$\hat{x} \approx \arg\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

Federated Learning with a Target Client

Goal: find $\check{x}$ such that

$$\check{x} \approx \arg\min_{x \in \mathbb{R}^d} f_1(x)$$

Federated Learning with a Target Client

Standard Federated Learning

Goal: find $\hat{x}$ such that

$$\hat{x} \approx \arg\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

Federated Learning with a Target Client

Goal: find $\check{x}$ such that

$$\check{x} \approx \arg\min_{x \in \mathbb{R}^d} f_1(x)$$

🧐 Eduard, wait a second… Isn't it trivial?

Federated Learning with a Target Client

Standard Federated Learning

Goal: find $\hat{x}$ such that

$$\hat{x} \approx \arg\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

$$f_i(x) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} [f_{\xi_i}(x)]$$

Federated Learning with a Target Client

Goal: find $\check{x}$ such that

$$\check{x} \approx \arg\min_{x \in \mathbb{R}^d} f_1(x)$$

🧐 Eduard, wait a second… Isn't it trivial?
⚙️ Workers have only finite number of samples
🤝 Maybe, some workers can be helpful
😶 🌫️ But we don't know which ones

Federated Learning with a Target Client

Standard Federated Learning

Goal: find $\hat{x}$ such that

$$\hat{x} \approx \arg\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

$$f_i(x) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[f_{\xi_i}(x)]$$

Federated Learning with a Target Client

Goal: find $\check{x}$ such that

$$\check{x} \approx \arg\min_{x \in \mathbb{R}^d} f_1(x)$$

🧐 Eduard, wait a second… Isn't it trivial?
⚙️ Workers have only finite number of samples
🤝 Maybe, some workers can be helpful
😶 🌫️ But we don't know which ones

Personalized Federated Learning

Goal: find something "in between" $\hat{x}$ and $\tilde{x}$
("in between" can be formalized in a number of ways)

$$\tilde{x} \approx \arg\min_{x \in \mathbb{R}^d} f_i(x) \quad \forall i \in \{1, \ldots, n\}$$

Is Parallel SGD Good for FL with a Target Client?

Parallel SGD: $$x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k,$$ where $$\mathbb{E}_k[g_i^k] = \nabla f_i(x^k)$$

Is Parallel SGD Good for FL with a Target Client?

Parallel SGD: $x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k,$ where $\mathbb{E}_k[g_i^k] = \nabla f_i(x^k)$

Toy 1-dimensional problem: $n = 100$ and $f_1(x) := x^2$

$$f_2(x) = f_3(x) = \ldots = f_{10}(x) := (x - 0.001)^2$$

$$f_{11}(x) = f_{12}(x) = \ldots = f_{100}(x) := (x - 10)^2$$

Is Parallel SGD Good for FL with a Target Client?

Parallel SGD: $x^{k+1} = x^k - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^k,$ where $\mathbb{E}_k[g_i^k] = \nabla f_i(x^k)$

Toy 1-dimensional problem: $n = 100$ and $f_1(x) := x^2$

$$f_2(x) = f_3(x) = \ldots = f_{10}(x) := (x - 0.001)^2$$

$$f_{11}(x) = f_{12}(x) = \ldots = f_{100}(x) := (x - 10)^2$$

- Parallel SGD (with small enough stepsize) converges to $\hat{x} = 9.00009$ but $\check{x} = 0$

Is Parallel SGD Good for FL with a Target Client?

Parallel SGD: $x^{k+1} = x^k - \gamma \cdot \boxed{\frac{1}{n}} \sum_{i=1}^{n} g_i^k,$ where $\mathbb{E}_k[g_i^k] = \nabla f_i(x^k)$

Toy 1-dimensional problem: $n = 100$ and $f_1(x) := x^2$

$$f_2(x) = f_3(x) = \ldots = f_{10}(x) := (x - 0.001)^2$$

$$f_{11}(x) = f_{12}(x) = \ldots = f_{100}(x) := (x - 10)^2$$

- Parallel SGD (with small enough stepsize) converges to $\hat{x} = 9.00009$ but $\check{x} = 0$
- Natural behavior since Parallel SGD is designed to solve the standard FL problem, i.e., Parallel SGD uses uniform averaging

Naive Approach

Use target's client data only:

$$x^{k+1} = x^k - \gamma g_1^k$$

☑ The method converges (under some assumptions) to $\check{x}$ – exactly as desired

Naive Approach

Use target's client data only:

$$x^{k+1} = x^k - \gamma g_1^k$$

✅ The method converges (under some assumptions) to $\check{x}$ – exactly as desired

❗ No collaboration!

- ❌ Poor generalization if the target client's dataset is small
- ❌ Small mini-batch → slow convergence for some tasks

Non-Implementable Approach

Use data from clients with the same data distribution:

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} g_i^k$$

Non-Implementable Approach

Use data from clients with the same data distribution:

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} g_i^k$$

subset of clients with the same data distribution as for the target client

Non-Implementable Approach

Use data from clients with the same data distribution:

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} g_i^k$$

subset of clients with the same data distribution as for the target client

- ✅ The method converges (under some assumptions) to $\check{x}$ – exactly as desired
- ✅ Collaboration is performed with useful clients only

Non-Implementable Approach

Use data from clients with the same data distribution:

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} g_i^k$$

subset of clients with the same data distribution as for the target client

✅ The method converges (under some assumptions) to $\check{x}$ – exactly as desired

✅ Collaboration is performed with useful clients only

❗ We don't know in advance who has the same distribution → non-implementable method

❌ Some clients with close distribution are ignored, though they could be useful

New Method: MeritFed

⚙ Update rule:

$$x^{k+1} = x^k - \gamma \sum_{i=1}^{n} w_i^{k+1} g_i^k$$

New Method: MeritFed

Update rule:

$$x^{k+1} = x^k - \gamma \sum_{i=1}^{n} \boxed{w_i^{k+1}} g_i^k$$

aggregation weights

New Method: MeritFed

Update rule:

$$x^{k+1} = x^k - \gamma \sum_{i=1}^{n} \boxed{w_i^{k+1}} g_i^k$$

aggregation weights

Key novelty:

$$\boxed{w^{k+1}} \approx \arg\min_{\boxed{w} \in \Delta_1^n} f_1 \left(x^k - \gamma \sum_{i=1}^{n} \boxed{w_i} g_i^k \right)$$

New Method: MeritFed

Update rule:

$$x^{k+1} = x^k - \gamma \sum_{i=1}^{n} w_i^{k+1} g_i^k$$

aggregation weights

Key novelty:

$$w^{k+1} \approx \arg\min_{w \in \Delta_1^n} f_1\left(x^k - \gamma \sum_{i=1}^{n} w_i g_i^k\right)$$

probability simplex

$$\Delta_1^n := \left\{ w \in \mathbb{R}^n \mid w \geq 0 \text{ and } \sum_{i=1}^{n} w_i = 1 \right\}$$

New Method: MeritFed

Update rule:

$$x^{k+1} = x^k - \gamma \sum_{i=1}^{n} w_i^{k+1} g_i^k$$

aggregation weights

Key novelty:

$$w^{k+1} \approx \arg\min_{w \in \Delta_1^n} f_1\left(x^k - \gamma \sum_{i=1}^{n} w_i g_i^k\right)$$

probability simplex

$$\Delta_1^n := \left\{ w \in \mathbb{R}^n \mid w \geq 0 \text{ and } \sum_{i=1}^{n} w_i = 1 \right\}$$

How to solve the above auxiliary problem?

MeritFed: How to Solve Auxiliary Problem

Approach 1: use fresh data (if one can get fresh samples)

MeritFed: How to Solve Auxiliary Problem

Approach 1: use fresh data (if one can get fresh samples)

✅ Standard stochastic optimization on a simplex – can be solved via Stochastic Mirror Descent

❌ In practice, data is limited → we don't have fresh data

MeritFed: How to Solve Auxiliary Problem

Approach 1: use fresh data (if one can get fresh samples)

- ✅ Standard stochastic optimization on a simplex – can be solved via Stochastic Mirror Descent
- ❌ In practice, data is limited → we don't have fresh data

Approach 2: approximate f_1 with its sampled version based on extra validation dataset/train dataset

MeritFed: How to Solve Auxiliary Problem

Approach 1: use fresh data (if one can get fresh samples)

✅ Standard stochastic optimization on a simplex – can be solved via Stochastic Mirror Descent
❌ In practice, data is limited → we don't have fresh data

Approach 2: approximate f_1 with its sampled version based on extra validation dataset/train dataset

$$w^{k+1} \approx \arg\min_{w \in \Delta_1^n} \hat{f}_1 \left(x^k - \gamma \sum_{i=1}^{n} w_i g_i^k \right)$$

$$\hat{f}_1(x) := \frac{1}{|\hat{D}|} \sum_{\xi \in \hat{D}} f_\xi(x)$$

MeritFed: How to Solve Auxiliary Problem

Approach 1: use fresh data (if one can get fresh samples)

✅ Standard stochastic optimization on a simplex – can be solved via Stochastic Mirror Descent
❌ In practice, data is limited → we don't have fresh data

Approach 2: approximate f_1 with its sampled version based on extra validation dataset/train dataset

$$w^{k+1} \approx \arg\min_{w \in \Delta_1^n} \hat{f}_1 \left(x^k - \gamma \sum_{i=1}^{n} w_i g_i^k \right)$$

✅ Again, one can solve it with (mini-batched) Mirror Descent
✍️ In theory, $\hat{D}$ should be large enough to have $f_1 \approx \hat{f}_1$

$$\hat{f}_1(x) := \frac{1}{|\hat{D}|} \sum_{\xi \in \hat{D}} f_\xi(x)$$

extra validation/training dataset

Mirror Descent

on a probability simplex with Kulback-Leibler distance as a Bregman divergence

Problem:

$$\min_{w \in \Delta_1^n} \varphi(w)$$

Method:

$$w^{t+1} = \frac{w^t \exp(-\eta \nabla \varphi(w^t))}{\sum_{i=1}^{n} w_i^t \exp(-\eta [\nabla \varphi(w^t))]_i}$$

- ✅ Exponent and product are computed component-wise
- ✅ One can use a zeroth-order version of it for the auxiliary problem in MeritFed (for privacy)

Amir Beck. 2017. First-order methods in optimization. SIAM.

Convergence Theory

Assumptions:

- Bounded variance
  ($\forall\ i \in \mathcal{G}$ - clients with the same distribution)

$$\mathbb{E}[\|g_i^k - \nabla f_1(x^k)\|^2 \mid x^k] \leq \sigma^2$$

Convergence Theory

Assumptions:

- Bounded variance
  ($\forall\ i \in \mathcal{G}$ - clients with the same distribution)

$$\mathbb{E}[\|g_i^k - \nabla f_1(x^k)\|^2 \mid x^k] \leq \sigma^2$$

- Smoothness

$$\|\nabla f_1(x) - \nabla f_1(y)\| \leq L\|x - y\|$$

Convergence Theory

Assumptions:

- Bounded variance
  ($\forall\, i \in \mathcal{G}$ - clients with the same distribution)

$$\mathbb{E}[\|g_i^k - \nabla f_1(x^k)\|^2 \mid x^k] \leq \sigma^2$$

- Smoothness

$$\|\nabla f_1(x) - \nabla f_1(y)\| \leq L\|x - y\|$$

- The auxiliary problem
  can be solved approximately

$$\mathbb{E}[f(x^{k+1}) \mid x^k, \{g_i^k\}_{i=1}^n] - \min_{w \in \Delta_1^n} f_1\left(x^k - \gamma \sum_{i=1}^n w_i g_i^k\right) \leq \delta$$

Convergence Theory

Assumptions:

- Bounded variance
  ($\forall\, i \in \mathcal{G}$ - clients with the same distribution)

$$\mathbb{E}[\|g_i^k - \nabla f_1(x^k)\|^2 \mid x^k] \leq \sigma^2$$

- Smoothness

$$\|\nabla f_1(x) - \nabla f_1(y)\| \leq L\|x - y\|$$

- The auxiliary problem
  can be solved approximately

$$\mathbb{E}[f(x^{k+1}) \mid x^k, \{g_i^k\}_{i=1}^n] - \min_{w \in \Delta_1^n} f_1\left(x^k - \gamma \sum_{i=1}^n w_i g_i^k\right) \leq \delta$$

Theorem: under the above assumptions MeritFed with $\gamma \leq {}^1/_{2L}$ converges as

$$\frac{1}{N}\sum_{k=0}^{N-1} \mathbb{E}[\|\nabla f_1(x^k)\|^2] \leq \frac{2\left(f_1(x^0) - f_1^{\inf}\right)}{\gamma N} + \frac{2\gamma L \sigma^2}{|\mathcal{G}|} + \frac{2\delta}{\gamma}$$

Convergence Theory

Assumptions:

- Bounded variance
  ($\forall\, i \in \mathcal{G}$ - clients with the same distribution)

$$\mathbb{E}[\|g_i^k - \nabla f_1(x^k)\|^2 \mid x^k] \le \sigma^2$$

- Smoothness

$$\|\nabla f_1(x) - \nabla f_1(y)\| \le L\|x - y\|$$

- The auxiliary problem
  can be solved approximately

$$\mathbb{E}[f(x^{k+1}) \mid x^k, \{g_i^k\}_{i=1}^n] - \min_{w \in \Delta_1^n} f_1\left(x^k - \gamma \sum_{i=1}^n w_i g_i^k\right) \le \delta$$

Theorem: under the above assumptions MeritFed with $\gamma \le {}^1/_{2L}$ converges as

$$\frac{1}{N}\sum_{k=0}^{N-1} \mathbb{E}[\|\nabla f_1(x^k)\|^2] \le \boxed{\frac{2\left(f_1(x^0) - f_1^{\inf}\right)}{\gamma N} + \frac{2\gamma L \sigma^2}{|\mathcal{G}|}} + \frac{2\delta}{\gamma}$$

the same bound as for “Ideal” Parallel SGD that uses only clients with the same data distribution

Convergence Theory

Assumptions:

- Bounded variance
  ($\forall\, i \in \mathcal{G}$ - clients with the same distribution)

$$\mathbb{E}[\|g_i^k - \nabla f_1(x^k)\|^2 \mid x^k] \le \sigma^2$$

- Smoothness

$$\|\nabla f_1(x) - \nabla f_1(y)\| \le L\|x - y\|$$

- The auxiliary problem
  can be solved approximately

$$\mathbb{E}[f(x^{k+1}) \mid x^k, \{g_i^k\}_{i=1}^n] - \min_{w \in \Delta_1^n} f_1\left(x^k - \gamma \sum_{i=1}^n w_i g_i^k\right) \le \boxed{\delta}$$

Theorem: under the above assumptions MeritFed with $\gamma \le 1/2L$ converges as

$$\frac{1}{N}\sum_{k=0}^{N-1} \mathbb{E}[\|\nabla f_1(x^k)\|^2] \le \underbrace{\frac{2\left(f_1(x^0) - f_1^{\inf}\right)}{\gamma N} + \frac{2\gamma L \sigma^2}{|\mathcal{G}|}}_{\text{the same bound as for "Ideal" Parallel SGD that uses only clients with the same data distribution}} + \boxed{\frac{2\delta}{\gamma}}$$

Term caused by inaccurate solution of aux. problem

Numerical Results: Problems and Baselines

We tested MeritFed on several tasks:

- Mean estimation
- Image classification: ResNet18 @ CIFAR10
- Emotions classification: BERT @ GoEmotions

Numerical Results: Problems and Baselines

We tested MeritFed on several tasks:

- Mean estimation
- Image classification: ResNet18 @ CIFAR10
- Emotions classification: BERT @ GoEmotions
- ResNet18 @ MedMNIST

Baselines:

- SGD Full: standard Parallel SGD
- SGD Ideal: only clients from $\mathcal{G}$ participate
- FedAdp (Wu & Wang, 2021): Parallel SGD with non-uniform aggregation weights based on cosine similarity between gradients
- TAWT (Chen et al., 2021): in theory, it uses hypergradient calculation; in practice, it is based on cosine similarity between gradients
- FedProx (Li et al., 2020)

- *Wu, H., & Wang, P. (2021). Fast-convergent federated learning with adaptive weighting. IEEE Transactions on Cognitive Communications and Networking*
- *Chen, S., Crammer, K., He, H., Roth, D., & Su, W. J. (2021). Weighted training for cross-task learning. arXiv preprint arXiv:2105.14095.*
- *Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems.*

Mean Estimation

Setup:

- Three distributions:
  $\mathcal{D}_1 = \mathcal{N}(0, \mathrm{I}), \mathcal{D}_2 = \mathcal{N}(\mu 1, \mathrm{I}), \mathcal{D}_3 = \mathcal{N}(e, \mathrm{I}),$
  where $1 = (1, \dots, 1)^\top$ and $\|e\| = 1$

Problem:

$$\min_{x \in \mathbb{R}^d} \mathbb{E}_{\xi \sim \mathcal{D}_1} \|x - \xi\|^2$$

Mean Estimation

Setup:

- Three distributions:
  $\mathcal{D}_1 = \mathcal{N}(0, \mathrm{I}), \mathcal{D}_2 = \mathcal{N}(\mu 1, \mathrm{I}), \mathcal{D}_3 = \mathcal{N}(e, \mathrm{I})$,
  where $1 = (1, \dots, 1)^\top$ and $\|e\| = 1$
- 150 clients: 5 workers have data from $\mathcal{D}_1$,
  95 – from $\mathcal{D}_2$, and 50 – from $\mathcal{D}_3$
- Each client has 1000 samples
  (target one has extra 1000 samples for validation)

Problem:

$$\min_{x \in \mathbb{R}^d} \mathbb{E}_{\xi \sim \mathcal{D}_1} \|x - \xi\|^2$$

Mean Estimation

Setup:

- Three distributions:
  $\mathcal{D}_1 = \mathcal{N}(0, \mathrm{I}), \mathcal{D}_2 = \mathcal{N}(\mu 1, \mathrm{I}), \mathcal{D}_3 = \mathcal{N}(e, \mathrm{I})$,
  where $1 = (1, \dots, 1)^\top$ and $\|e\| = 1$
- 150 clients: 5 workers have data from $\mathcal{D}_1$,
  95 – from $\mathcal{D}_2$, and 50 – from $\mathcal{D}_3$
- Each client has 1000 samples
  (target one has extra 1000 samples for validation)

Problem:

$$\min_{x \in \mathbb{R}^d} \mathbb{E}_{\xi \sim \mathcal{D}_1} \|x - \xi\|^2$$

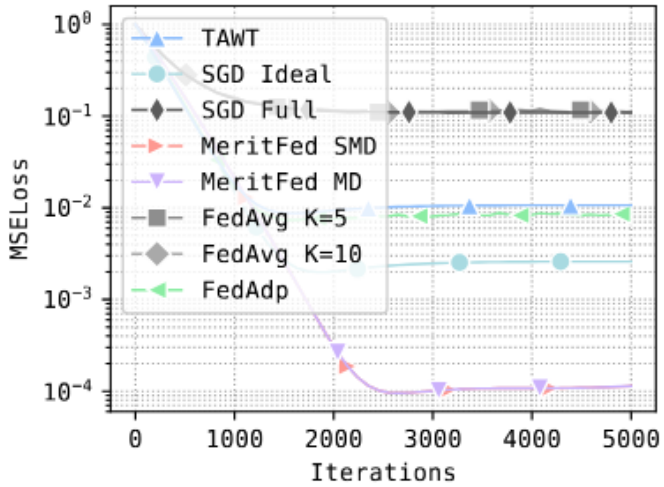


Figure 1: Mean Estimation: $\mu = 0.001$, MD learning rate = 3.5.

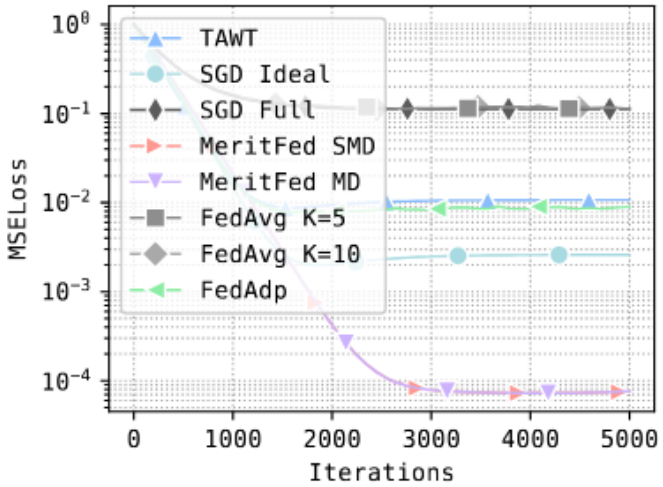


Figure 2: Mean Estimation: $\mu = 0.01$, MD learning rate = 4.5.

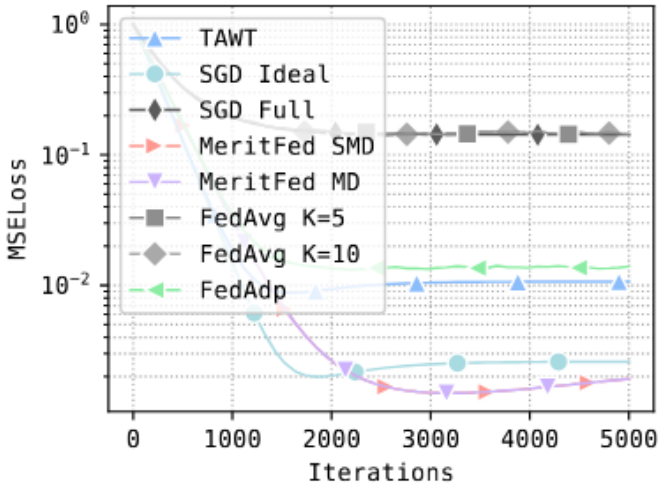


Figure 3: Mean Estimation: $\mu = 0.1$, MD learning rate = 12.5.

ResNet18 @ CIFAR10: Setup

- 20 clients are divided into three groups
- Goal of the first client: classify the first three classes (has data for those classes only)
- Second group (10 workers): $\alpha \in (0,1]$ portion of data consists of samples from classes 0, 1, 2;
  $1 - \alpha$ portion of data contains samples from classes 3, 4, 5
- Third group (9 workers): data consists of samples from classes 6, 7, 8, 9

ResNet18 @ CIFAR10: Results with Extra Data

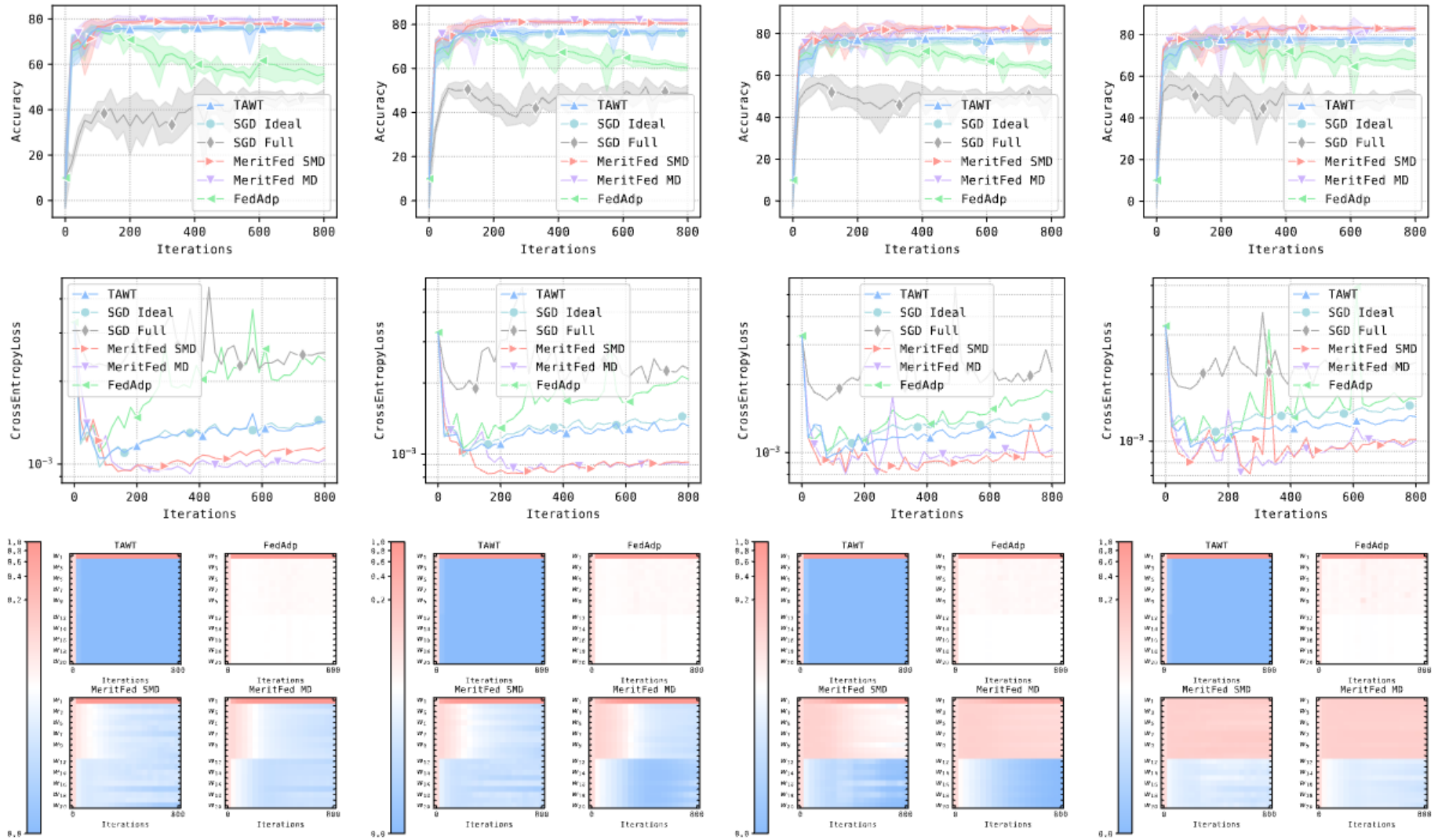


Figure 4: CIFAR10 (extra val.): $\alpha = 0.5$

Figure 5: CIFAR10 (extra val.): $\alpha = 0.7$

Figure 6: CIFAR10 (extra val.): $\alpha = 0.9$

Figure 7: CIFAR10 (extra val.): $\alpha = 0.99$.

ResNet18 @ CIFAR10: Results without Extra Data

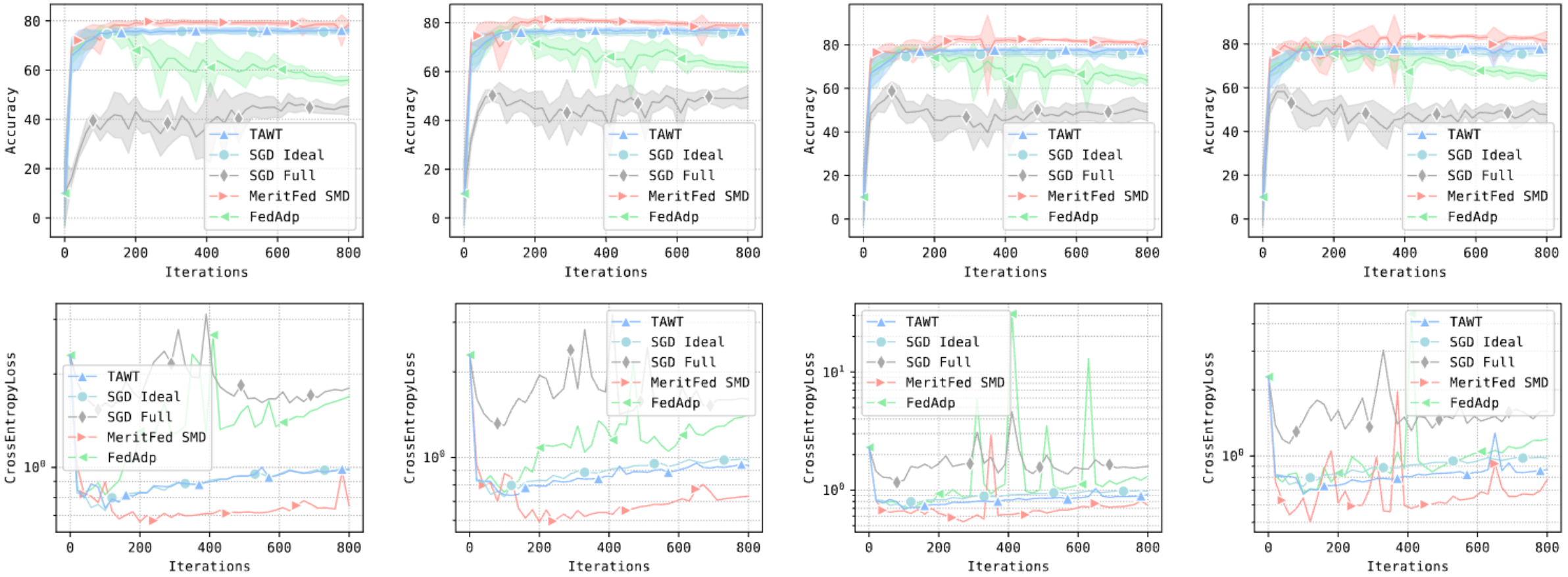


Figure 14: CIFAR10: $\alpha = 0.5$

Figure 15: CIFAR10: $\alpha = 0.7$

Figure 16: CIFAR10: $\alpha = 0.9$

Figure 17: CIFAR10: $\alpha = 0.99$

MeritFed works even without extra data!

BERT @ GoEmotions: Setup

- 20 clients are divided into three groups
- Goal of the first client: classify the “joy” emotion
- Second group (10 workers): $\alpha \in (0,1]$ portion of data consists of samples from “joy” class; $1-\alpha$ portion of data contains samples from “neutral” class
- Third group (9 workers): data consists of samples from “neutral” class and other basic emotions (like “anger”, “fear”, “sadness” and so on)

BERT @ GoEmotions: Results with Extra Data

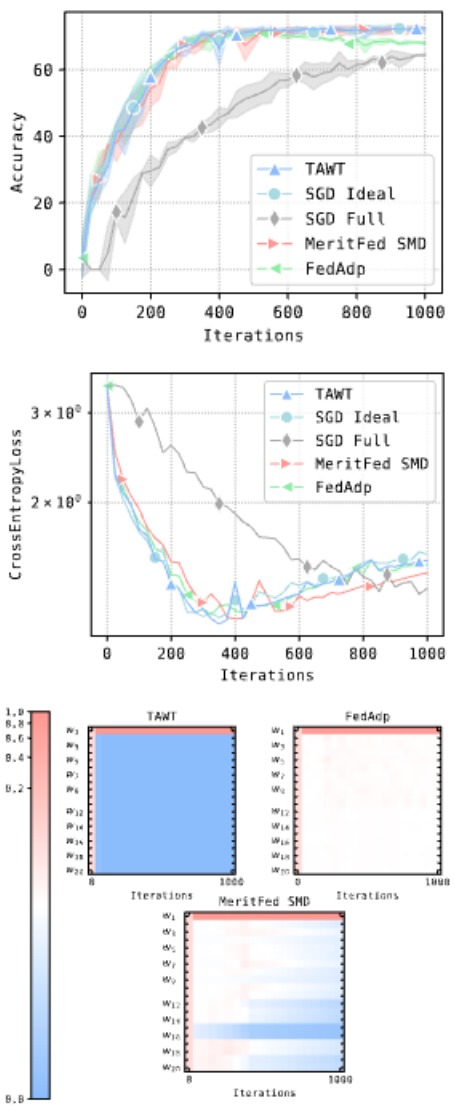


Figure 8: GoEmotions (extra val.): $\alpha = 0.5$

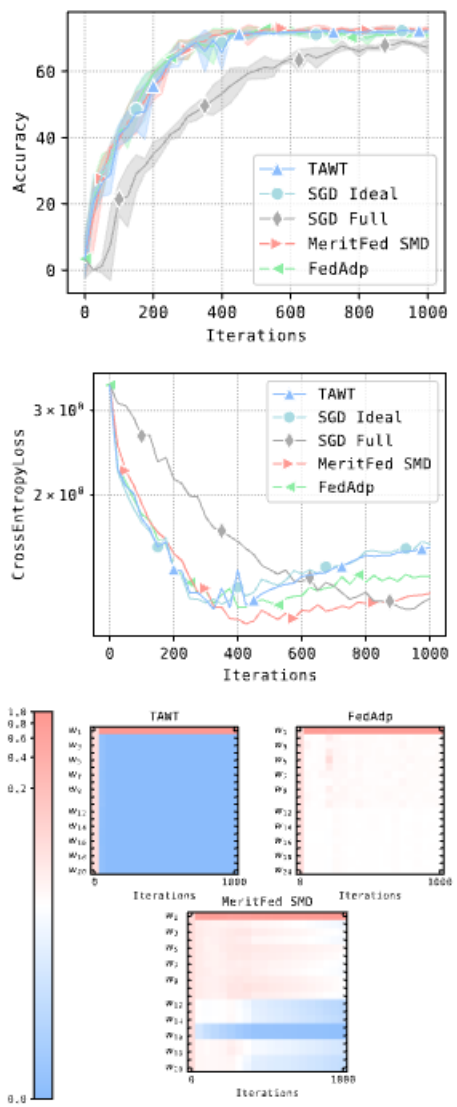


Figure 9: GoEmotions (extra val.): $\alpha = 0.7$

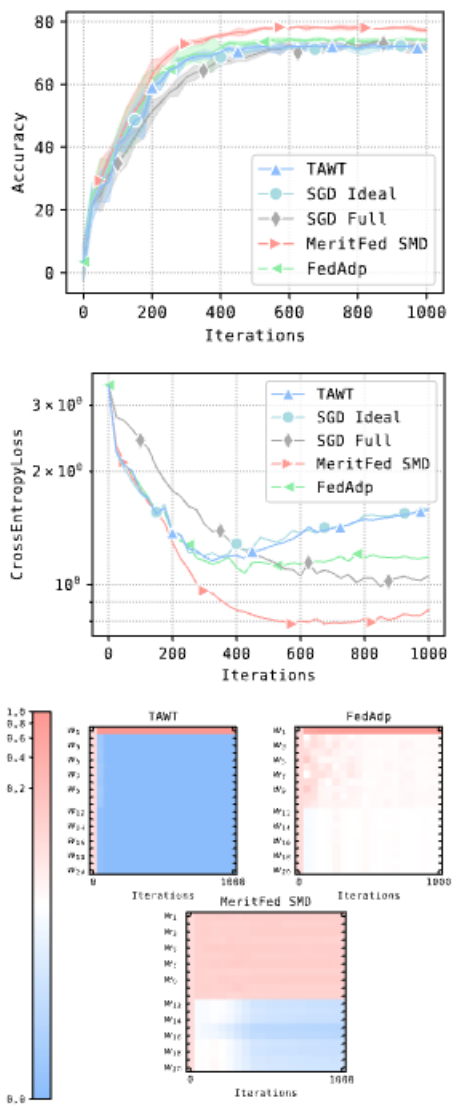


Figure 10: GoEmotions (extra val.): $\alpha = 0.9$

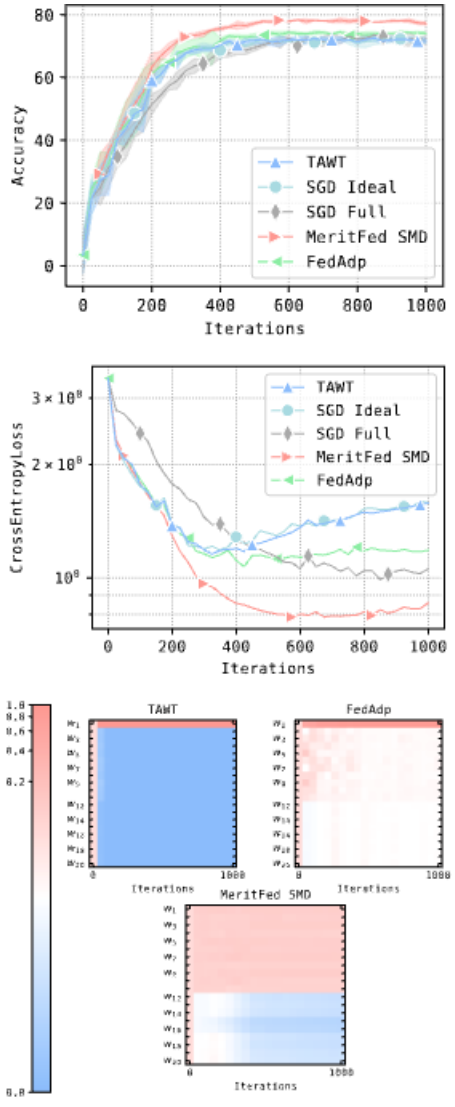


Figure 11: GoEmotions (extra val.): $\alpha = 0.99$

Bonus: Byzantine-Robustness for Free

💡 Observation: MeritFed will work even if other workers send (intentionally) “harmful” vectors
✅ In other words, MeritFed is *Byzantine-robust*

Setup:

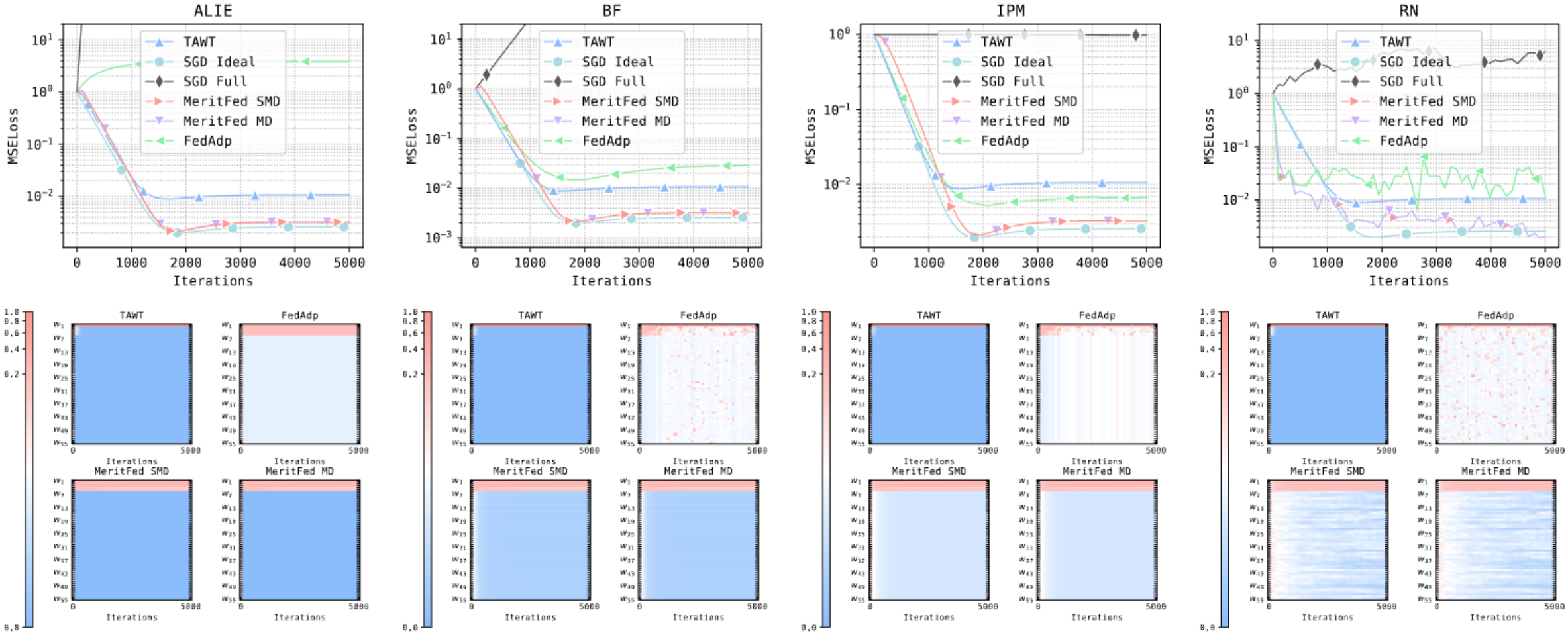
- Mean estimation with 55 clients
- 5 clients have the same distribution as a target one
- 50 clients know the target distribution and perform a Byzantine-attack

Attacks:

- ALIE (Baruch et al., 2019)
- IPM (Xie et al., 2019)
- Bit Flipping (BF): Byzantines send $(-g_i^k)$ instead of g_i^k
- Random Noise (RN): Byzantines send g_i^k + Gaussian noise

- *Baruch, G., Baruch, M., & Goldberg, Y. A little is enough: Circumventing defenses for distributed learning. NeurIPS 2019.*
- *Xie, C., Koyejo, O., & Gupta, I. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. UAI 2020.*

Bonus: Byzantine-Robustness for Free



Low-Resource Machine Translation

V. Moskvoretskii, N. Tupitsa, C. Biemann, S. Horváth, E. Gorbunov, I. Nikishina. *Low-Resource Machine Translation through the Lens of Personalized Federated Learning* (EMNLP Findings 2024)

Viktor Moskvoretskii
Researcher
Skoltech
(searches for PhD positions!)

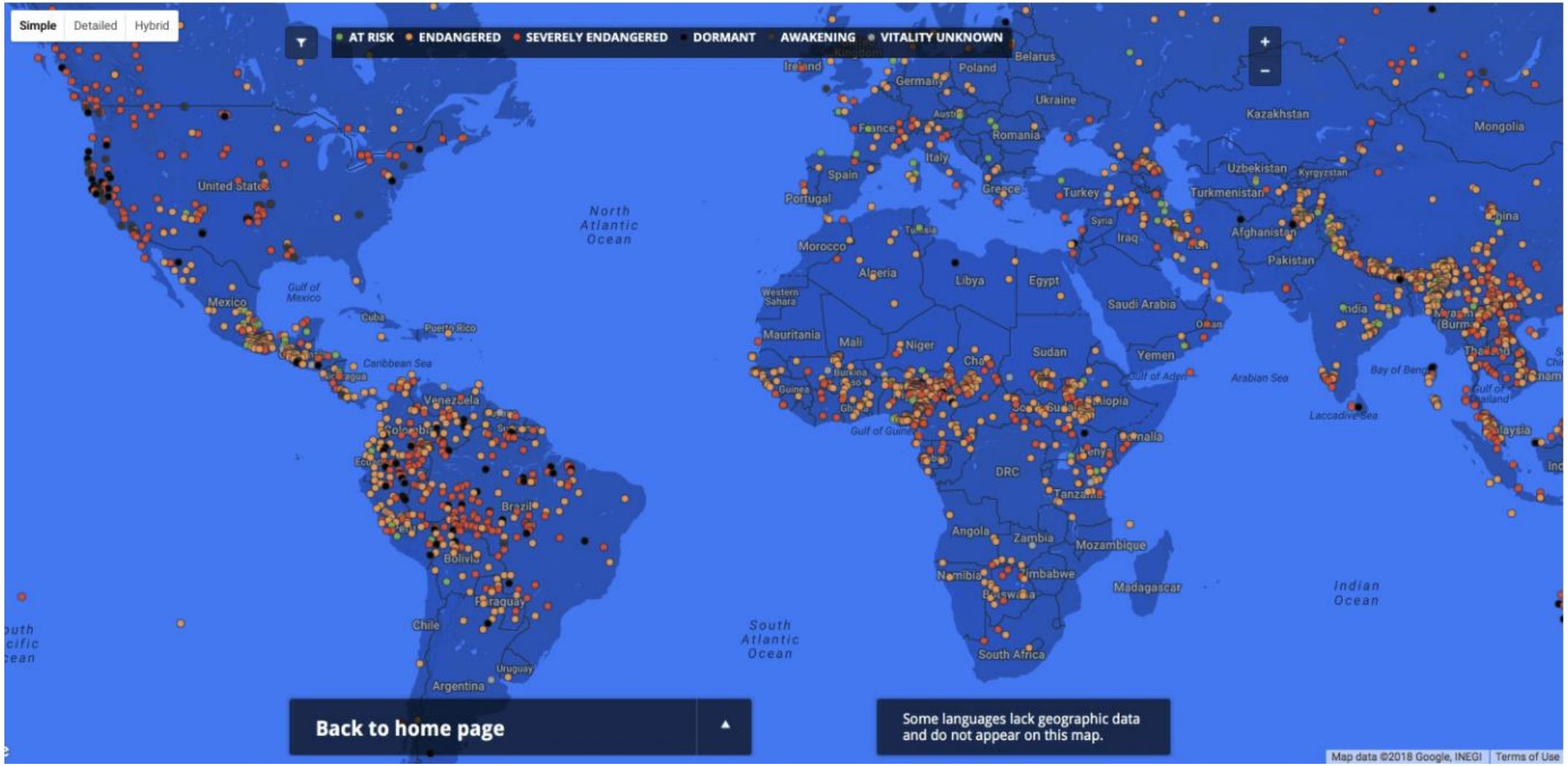
Nazarii Tupitsa
Research Assistant
MBZUAI

Chris Biemann
Professor
Universität Hamburg

Samuel Horváth
Assistant Professor
MBZUAI

Irina Nikishina
Postdoc
Universität Hamburg

There are about 7000 Languages



…but many of them possess low amount of resources (texts)

https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Training LLMs for Low-Resource Languages

Existing approaches:

- Cross-lingual transfer learning (from high-resource to low-resource languages)

https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Training LLMs for Low-Resource Languages

Existing approaches:

- Cross-lingual transfer learning (from high-resource to low-resource languages)
- Zero-shot learning: rain a model in one domain and assume it generalizes more or less out-of-the-box in a low-resource domain

https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Training LLMs for Low-Resource Languages

Existing approaches:

- Cross-lingual transfer learning (from high-resource to low-resource languages)
- Zero-shot learning: rain a model in one domain and assume it generalizes more or less out-of-the-box in a low-resource domain
- Few shot learning: train a model in one domain and use only few examples from a low-resource domain to adapt it

https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Training LLMs for Low-Resource Languages

Existing approaches:

- Cross-lingual transfer learning (from high-resource to low-resource languages)
- Zero-shot learning: rain a model in one domain and assume it generalizes more or less out-of-the-box in a low-resource domain
- Few shot learning: train a model in one domain and use only few examples from a low-resource domain to adapt it
- Joint multilingual learning: train a single model on a mix of datasets in all languages

https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Training LLMs for Low-Resource Languages

Existing approaches:

- Cross-lingual transfer learning (from high-resource to low-resource languages)
- Zero-shot learning: rain a model in one domain and assume it generalizes more or less out-of-the-box in a low-resource domain
- Few shot learning: train a model in one domain and use only few examples from a low-resource domain to adapt it
- Joint multilingual learning: train a single model on a mix of datasets in all languages

🧐 Main difficulty: it is unclear in-advance what languages to use as “helpers” and to what extent

https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Training LLMs for Low-Resource Languages

Existing approaches:

- Cross-lingual transfer learning (from high-resource to low-resource languages)
- Zero-shot learning: rain a model in one domain and assume it generalizes more or less out-of-the-box in a low-resource domain
- Few shot learning: train a model in one domain and use only few examples from a low-resource domain to adapt it
- Joint multilingual learning: train a single model on a mix of datasets in all languages

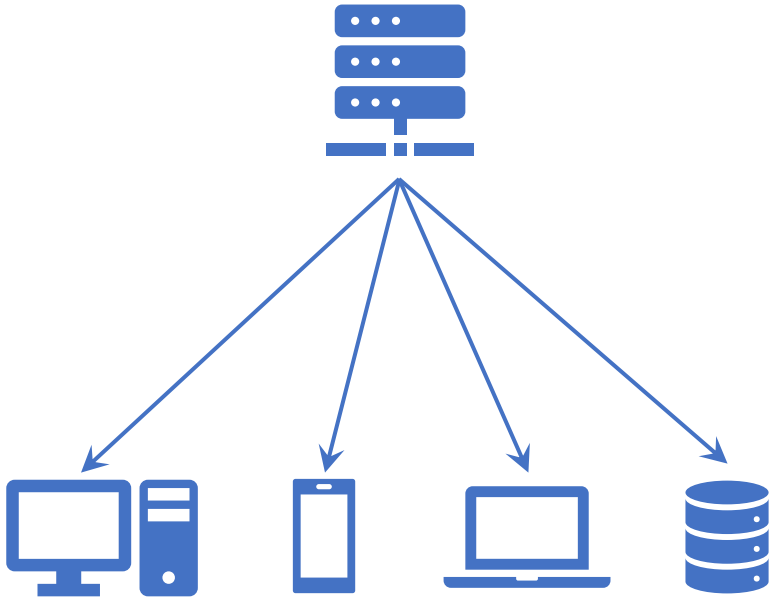
🧐 Main difficulty: it is unclear in-advance what languages to use as “helpers” and to what extent

💡 Wait a second… but this is exactly the same problem as in FL with a target client

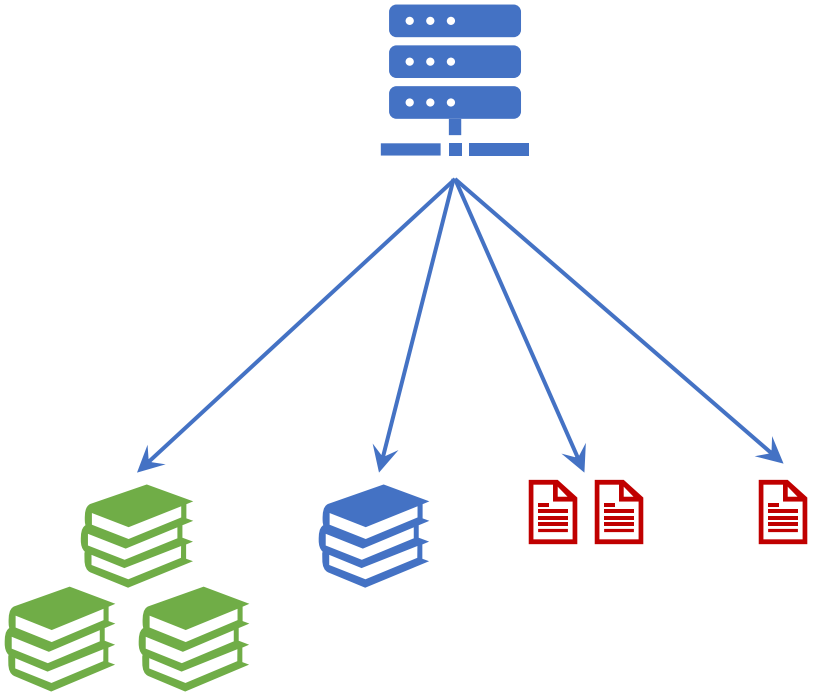
https://phontron.com/class/multiling2022/assets/slides/multiling-03-typology.pdf

Languages in NLP As Clients in FL

Federated Learning

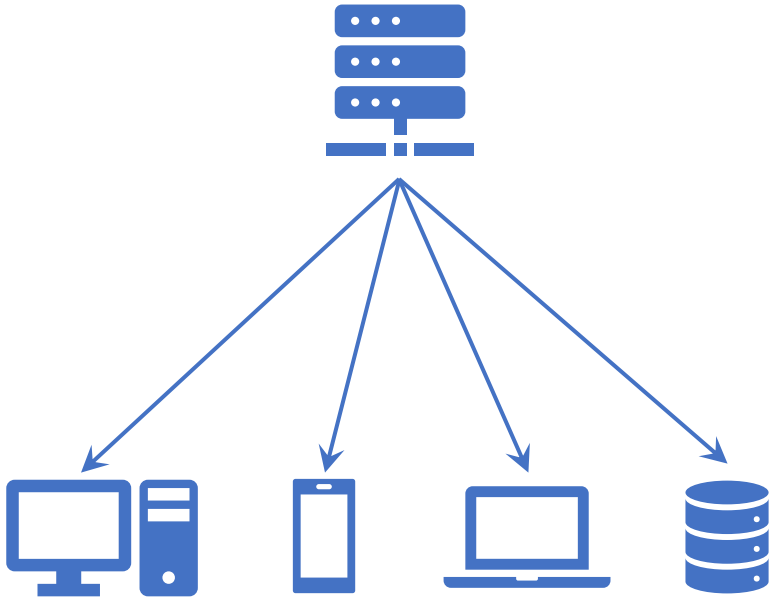


Natural Language Processing

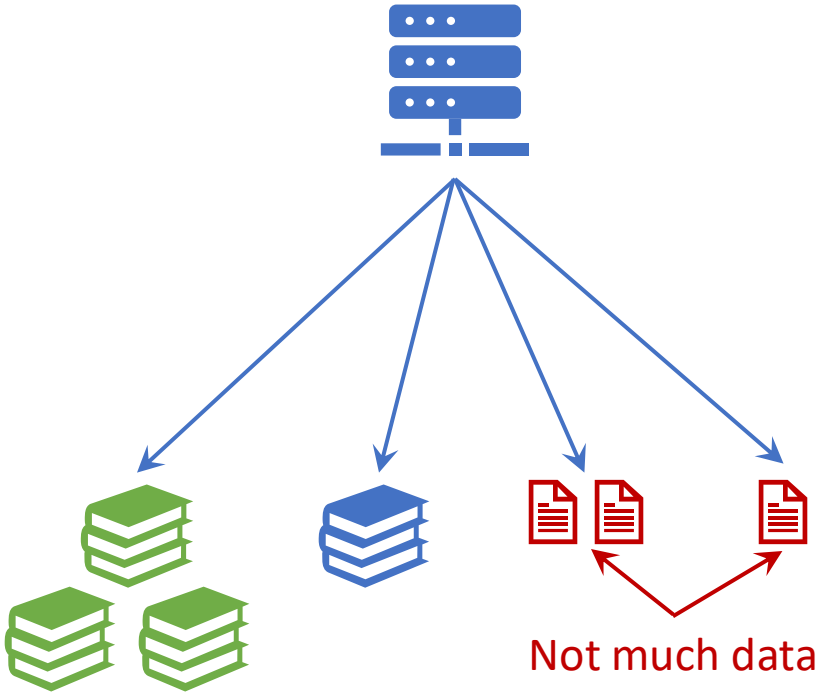


Languages in NLP As Clients in FL

Federated Learning



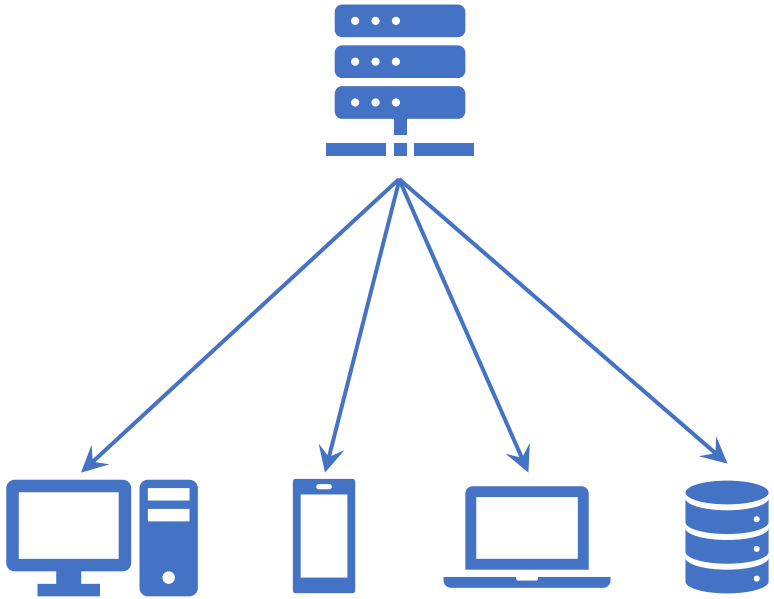
Natural Language Processing



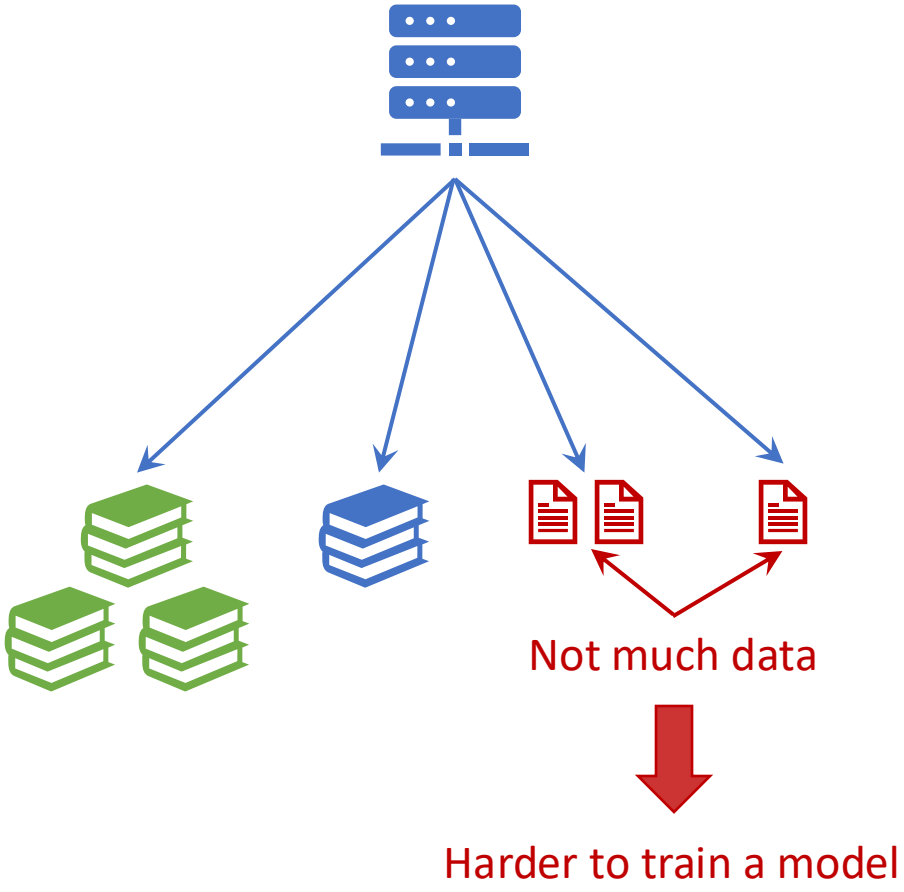
Harder to train a model

Languages in NLP As Clients in FL

Federated Learning



Natural Language Processing



Let us apply MeritFed!

MeritOpt

a.k.a. MeritFed-type wrapper for any stochastic first-order method

$$x^{k+1} = \texttt{OptStep}\left(x^k, \sum_{i=1}^{n} w_i^{k+1} g_i^k, \gamma_k\right)$$

$$w^{k+1} \approx \arg\min_{w \in \Delta_1^n} f_1\left(\texttt{OptStep}\left(x^k, \sum_{i=1}^{n} w_i g_i^k, \gamma_k\right)\right)$$

For example, OptStep can be a step of SGD, AdaGrad, RMSProp, Adam, etc.

Considered Tasks and Setup

Tasks:

- Multilingual Machine Translation on South East Asian languages
- Multilingual Machine Translation on he subset of Sami languages (from Finno-Ugric languages)

Model: M2M100

Baselines:

- Fine-tuning to all languages including the target language (FTall)
- Fine-tuning to all languages except the target language (FTnot)
- Fine-tune to the target language only (FTonlyt)
- Continuous Pretraining to all languages, followed by additional fine-tuning to the target language (CPall)
- Continuous Pretraining to all languages but the target, followed by additional fine-tuning to the target language (CPnot)

South East Asian Languages

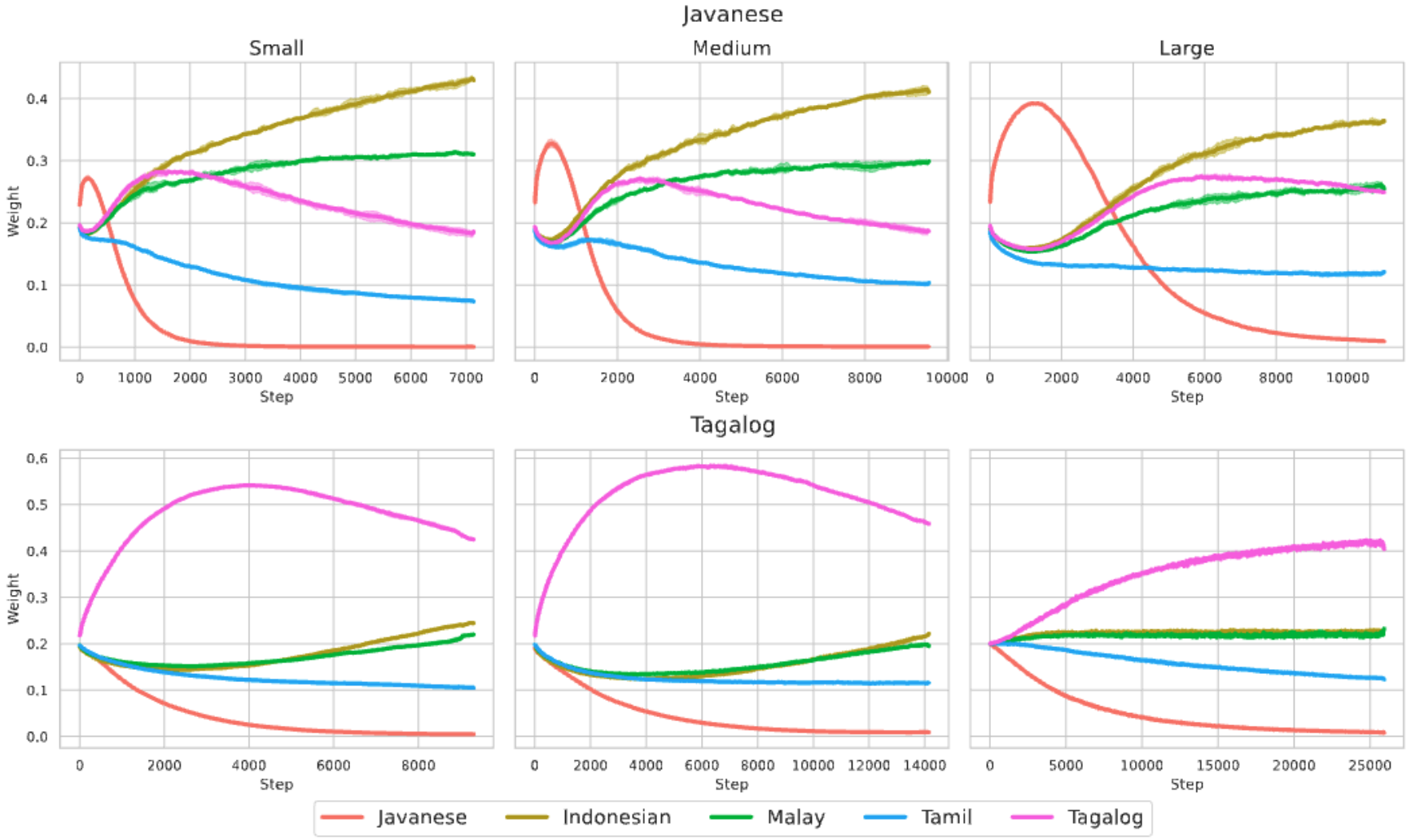


Figure 1: Weights distribution for South East Asian languages. Target languages and data sizes are in captions.

South East Asian Languages

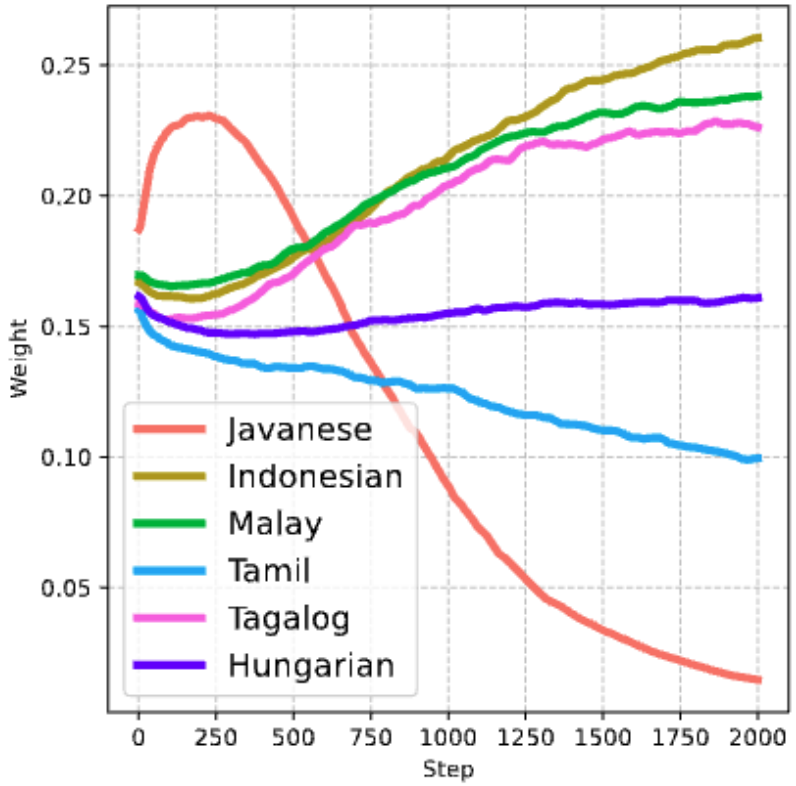


Figure 3: Weights distribution for target Indonesian language with unrelated Hungarian included.

South East Asian Languages

Method	Tagalog						Javanese					
	Small		Medium		Large		Small		Medium		Large	
	Score	Steps	Score	Steps	Score	Steps	Score	Steps	Score	Steps	Score	Steps
CP_{All}	29.24 ± 0.06	21K	30.99 ± 0.04	40K	33.89 ± 0.15	124K	19.43 ± 0.14	12K	20.05 ± 0.12	25K	20.97 ± 0.13	87K
CP_{NoT}	28.72 ± 0.16	15K	30.50 ± 0.12	42K	33.74 ± 0.19	129K	19.46 ± 0.12	12K	19.95 ± 0.12	25K	21.19 ± 0.09	89K
MeritFed	**29.73 ± 0.04**	14K	**31.42 ± 0.07**	14K	33.53 ± 0.27	47K	**19.74 ± 0.03**	2K	**20.23 ± 0.11**	3K	**21.44 ± 0.13**	8K

Table 2: Mean SpBLEU scores and the number of steps required to reach them for baselines and MeritFed within the different data sizes of Javanese and Tagalog languages.

Conclusion

Main Takeaways

- Merit(Fed)Opt – new method for (FL with a target client) for learning from the diverse datasets
- Personalized Federated Learning can be applied to the training of LLMs for low-resource languages

Limitations

- Scalability
- We tried the approach for low-resource languages only
- Results are provided for M2M100, while numerous LLMs are available
- Only one target client
- Theory does not show how helpful other clients are if they have different but close distribution

Thank you!