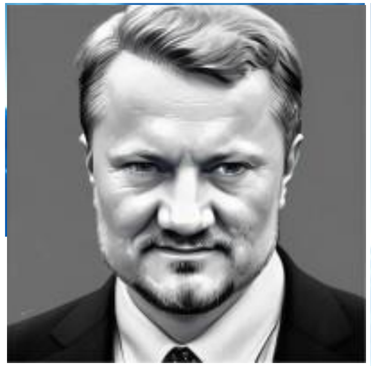
جامعة محمد بن زايد للذكاء الاصطناعي
MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE



Peter Richtarik

H. Brendan McMahan

Jakub Konecny

Empowering Distributed AI

via Federated Learning

Instructor: Martin Takac

Thanks to Samuel Horvath, Eduard Gorbunov and Praneeth Vepakomma for sharing their slides ☺

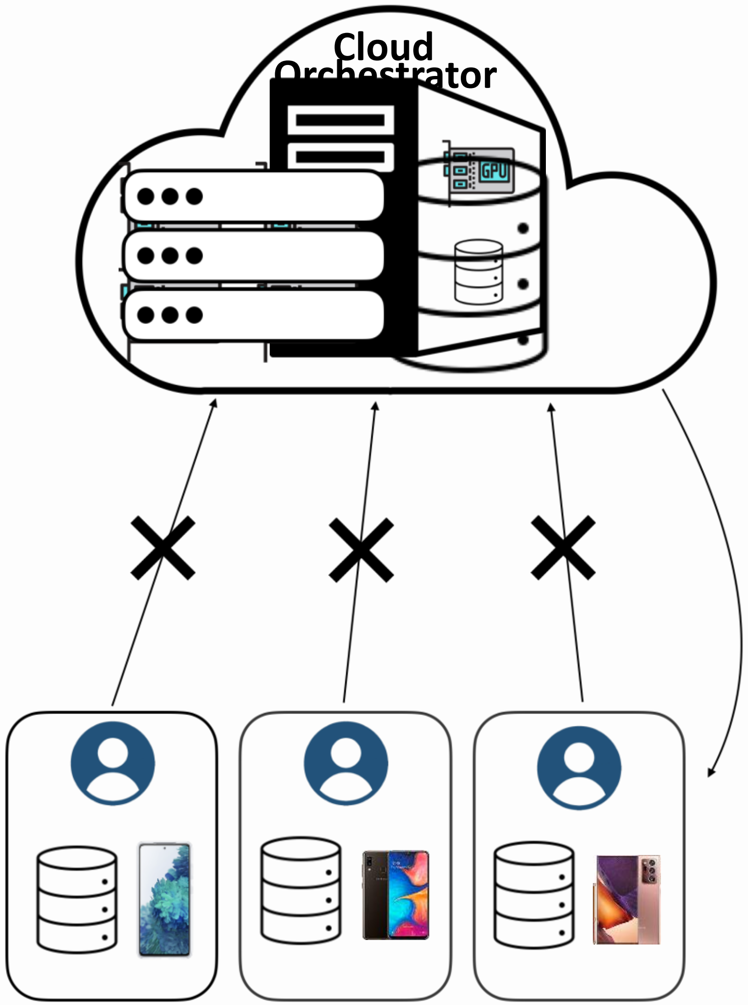
Agenda

- Motivation
- How we train machine learning privately
- Privacy concerns and applications
- Federated Learning (FL) and algorithms
- Personalized FL and LoRA
- Training as a service

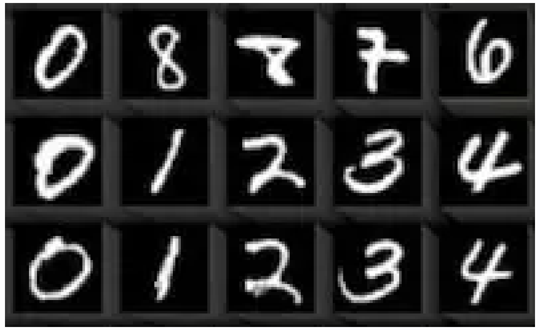
Motivation

Federated Learning – what it is and why?

ML – past and present

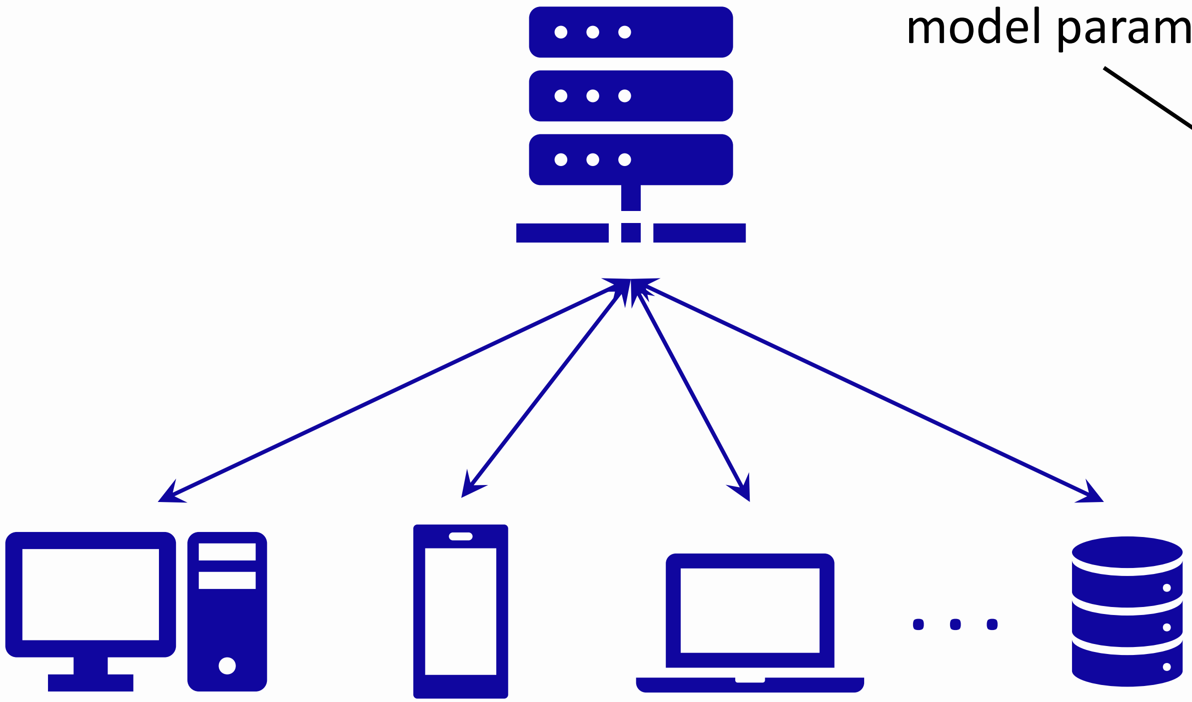


Yandex Ads



- **ML success:**
  - large-scale training infrastructures
  - the vast amounts of training data
- Negative **privacy** implications of data collection
- **Privacy Initiatives:**
  - GDPR (European Commission)
  - Learning with Privacy at Scale (Apple)
- We need to **bring training to the edge (decentralized)**
- Data locality paradigm (**lower carbon footprint** of distributed learning)

The problem setting

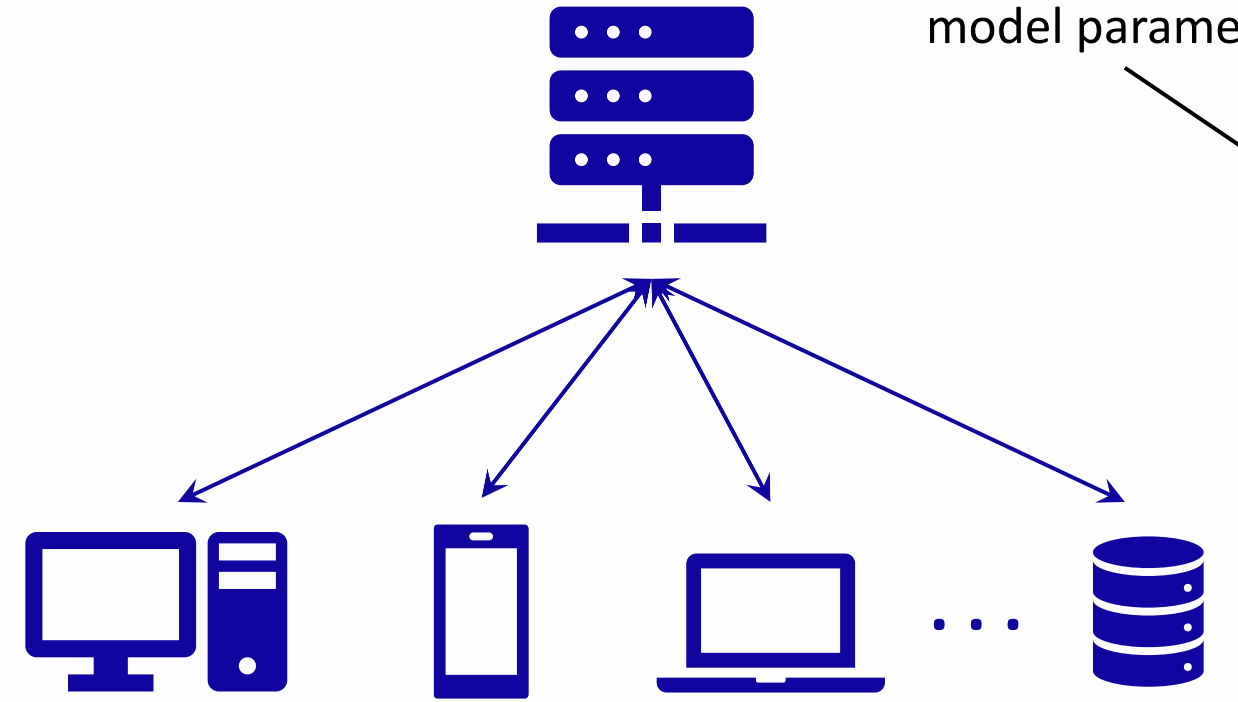


model parameters

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

# of parameters

The problem



n workers/clients

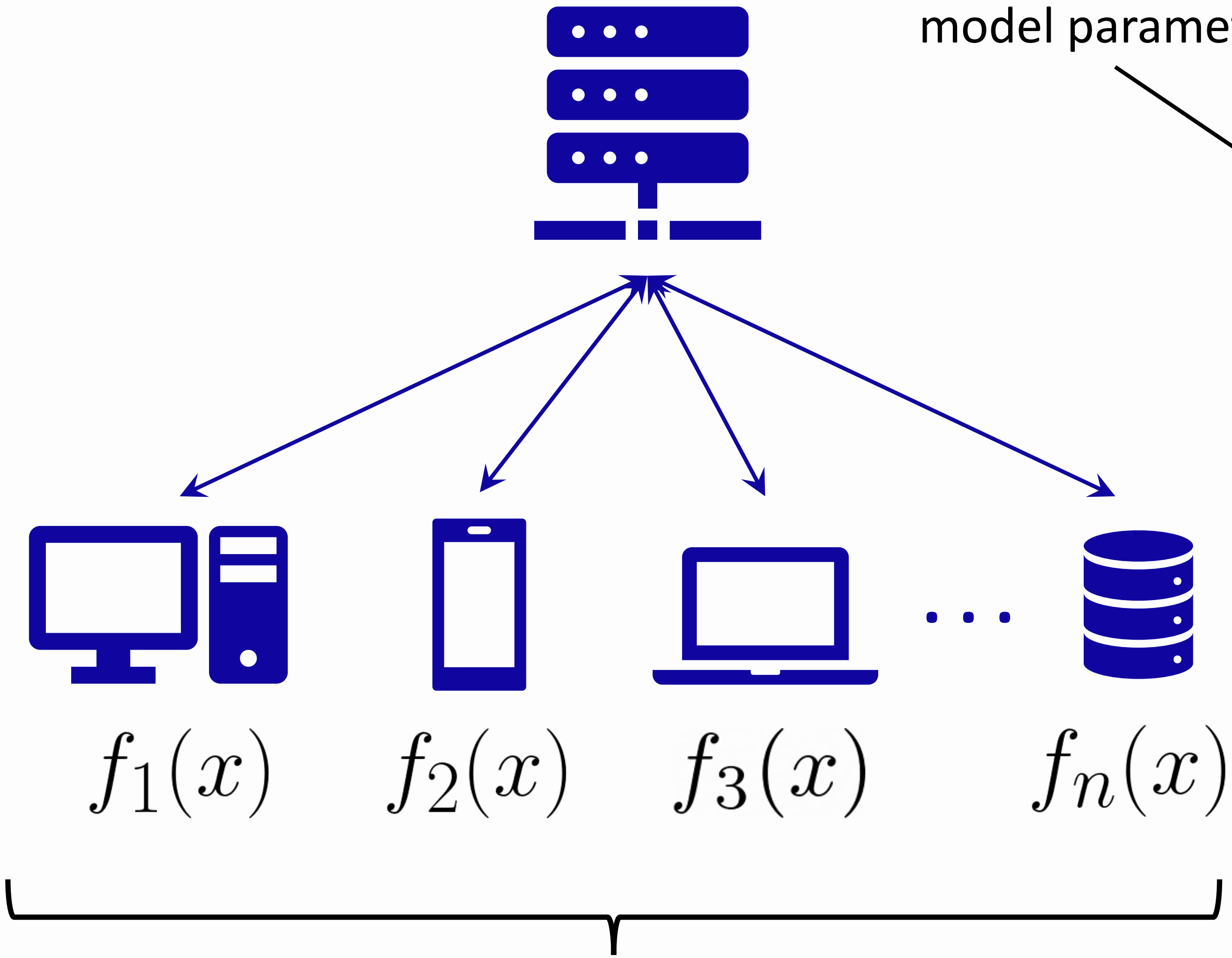
model parameters

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

# of parameters

# of workers/clients

The problem



n workers/clients

model parameters

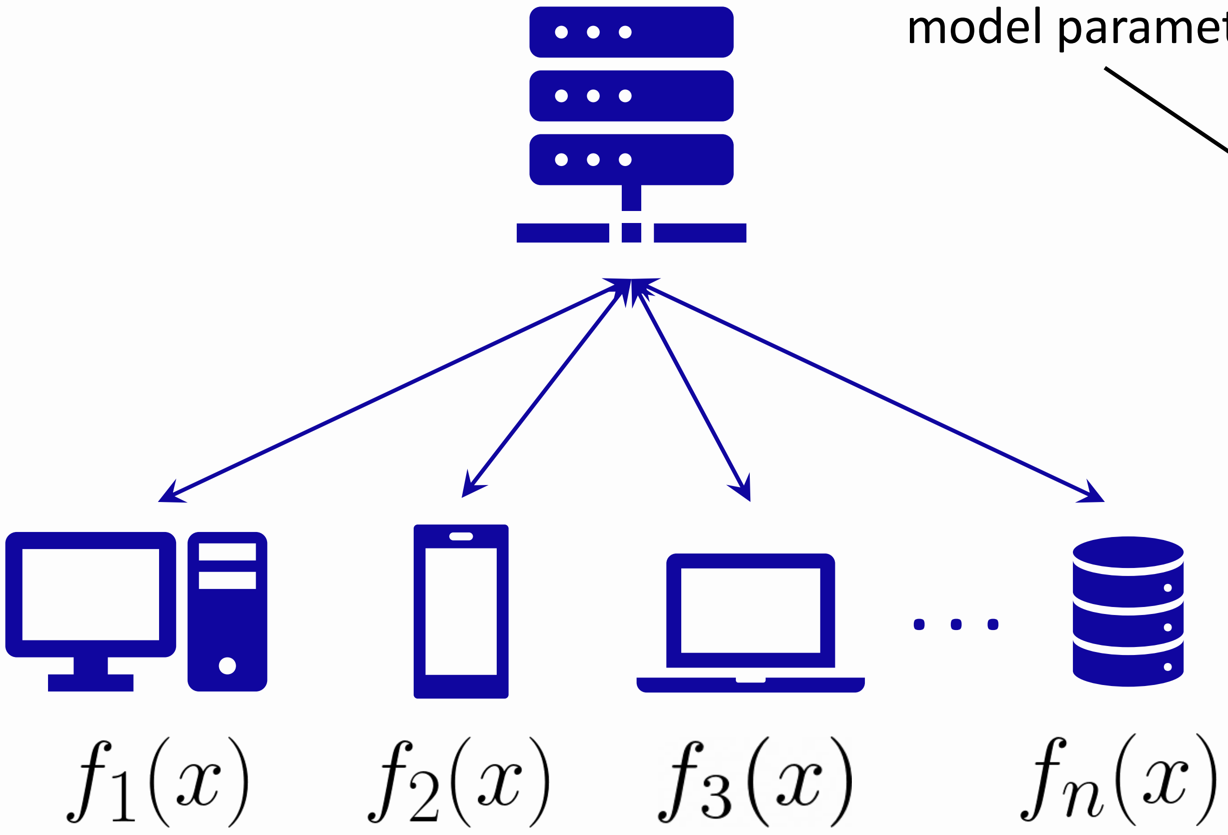
$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{\boxed{n}} \sum_{i=1}^{\boxed{n}} f_i(x) \right\}$$

# of parameters

# of workers/clients

loss on the data accessible on worker i

The problem



n workers/clients

model parameters

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

How can we train the ML model in a distributed way?

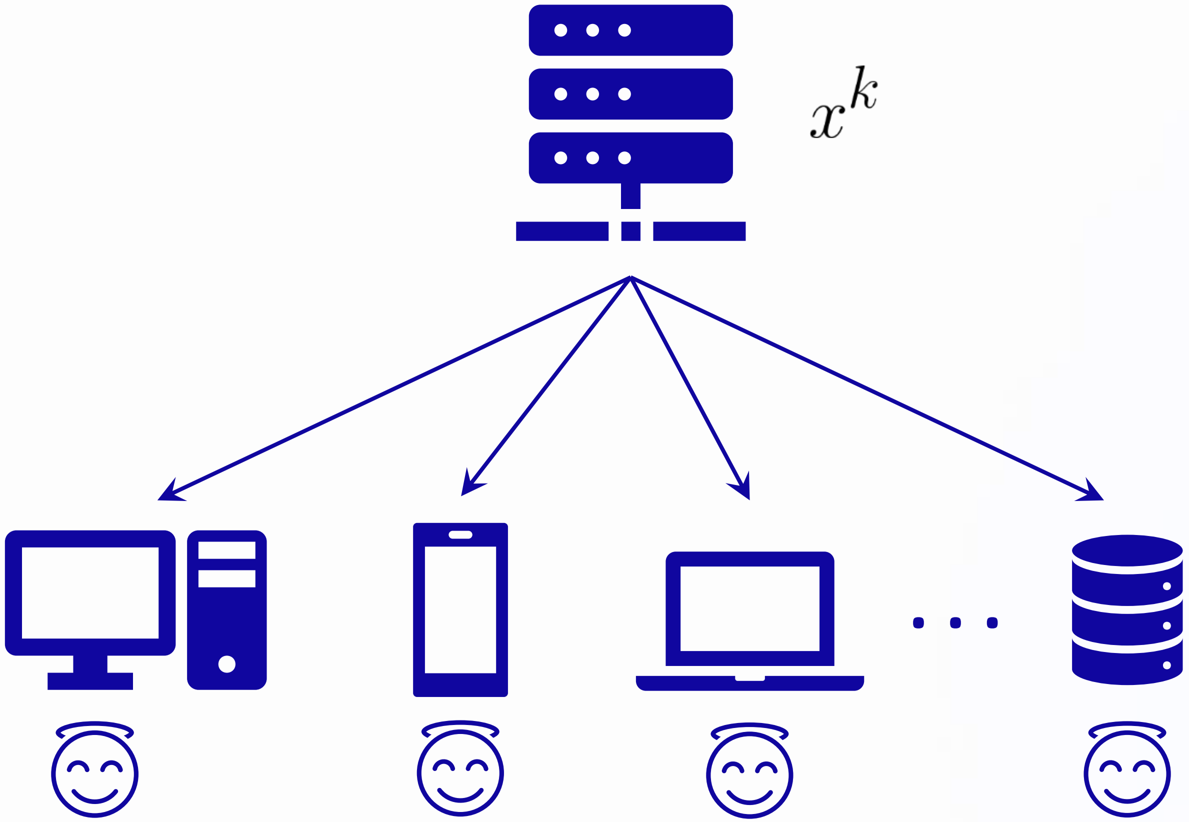
Key features:

- The problem is hard to solve for one client
- Clients do not know each other

Parallel SGD

Iteration k:

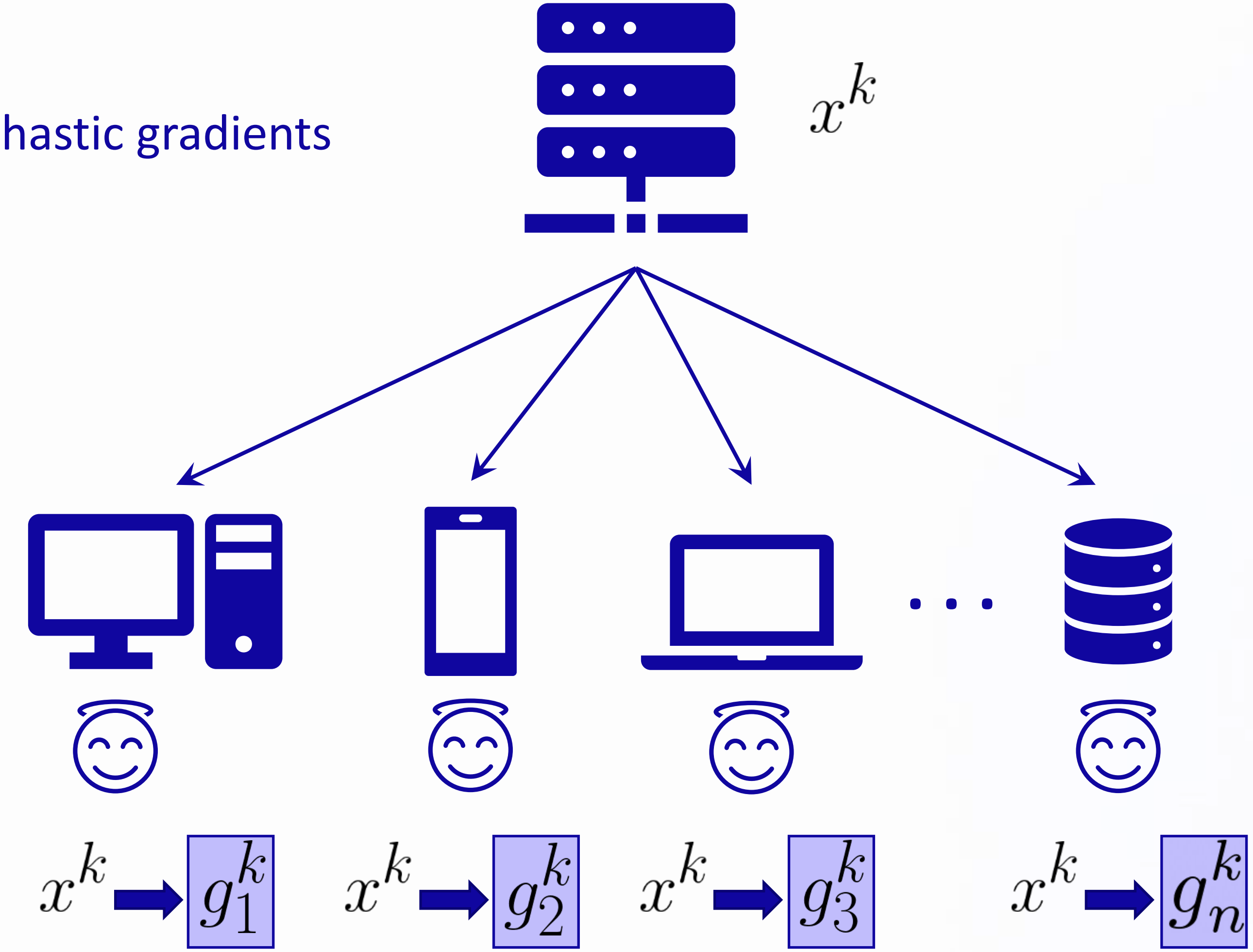
1. Server broadcasts x^k



Parallel SGD

Iteration k:

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

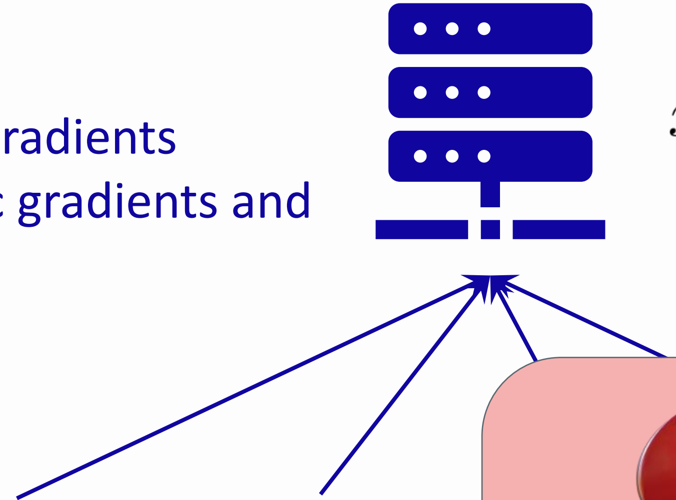


$$\mathbb{E}_k\left[\boxed{g_i^k}\right] = \nabla f_i(x^k)$$

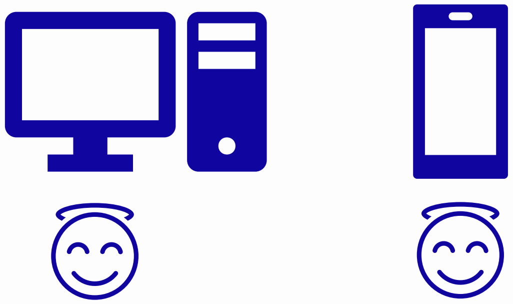
Parallel SGD

Iteration k:

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



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

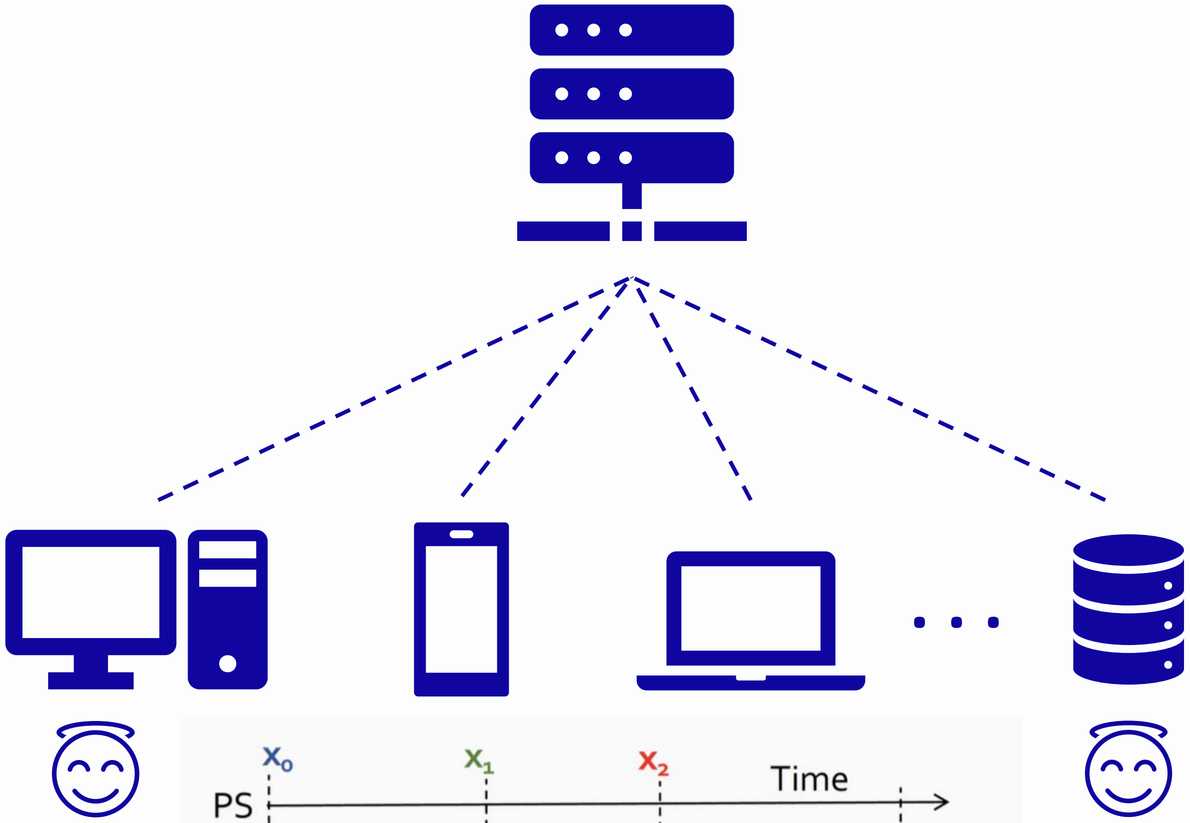


Is this the correct approach? Should we re-weight updates in practice?

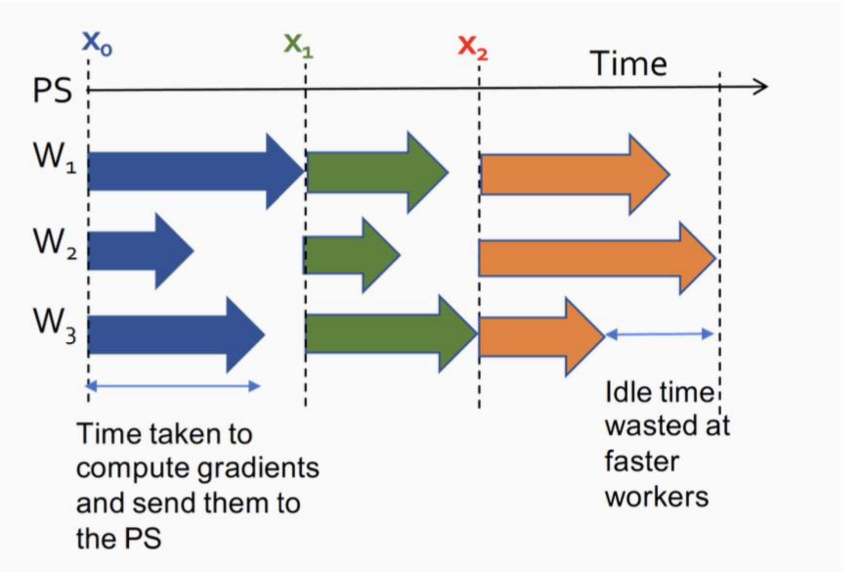
$$x^k \rightarrow \boxed{g_1^k} \qquad x^k \rightarrow \boxed{g_2^k} \qquad x^k \rightarrow \boxed{g_3^k} \qquad x^k \rightarrow \boxed{g_n^k}$$

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

Parallel SGD – bottlenecks



What are the issues with this approach?



1. Each optimization iteration needs two communications
2. We need to communicate $d \cdot 4$ bytes each way
3. Some workers can “die” during the training
4. Some workers can be much slower then others, leading to delays

Use-case 1 - Training Image-net

- SGD is awesome method
- Fast computation
  (each iteration = just one gradient)

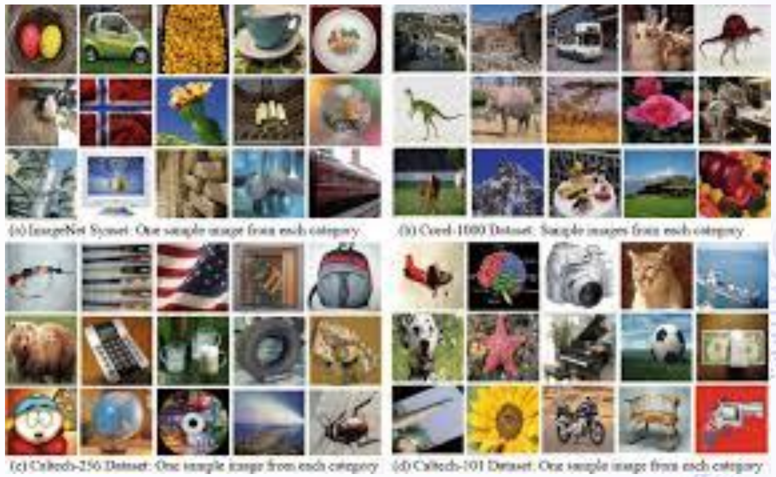
Challenge: How to utilize a huge computer cluster?

Idea: Choose subset of functions (batch) and use the average of their gradients

But: No free lunch

- More samples doesn't mean reduction of learning time
- Often, the optimal batch is around 128 (too much commu.)

IMAGENET

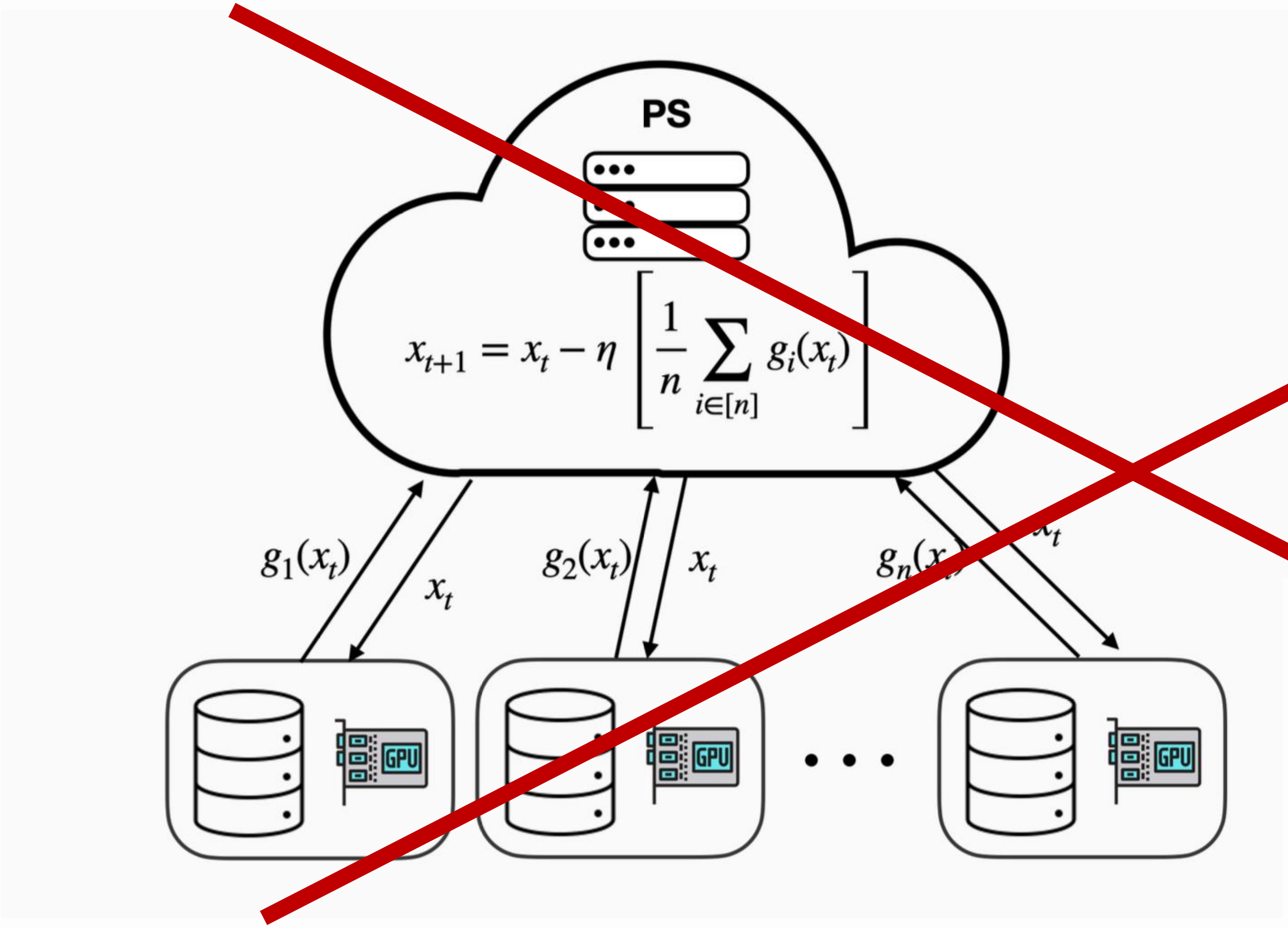


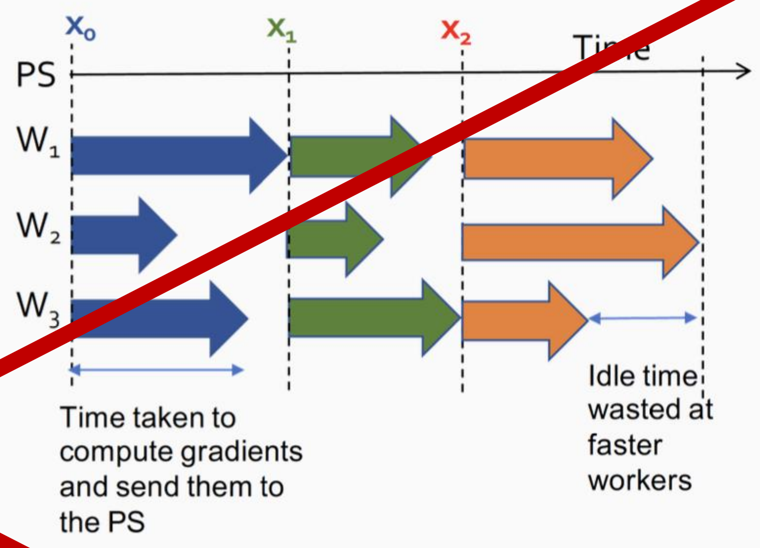
- ~14,000,000 images
- model size: 200 MB
- one pass over data = 250,000 it.
  (batch size of 128 and 10 nodes)
- network: 1Gb/s

Communication cost (for 1 epoch) = 9 days (with IDLE cpus)

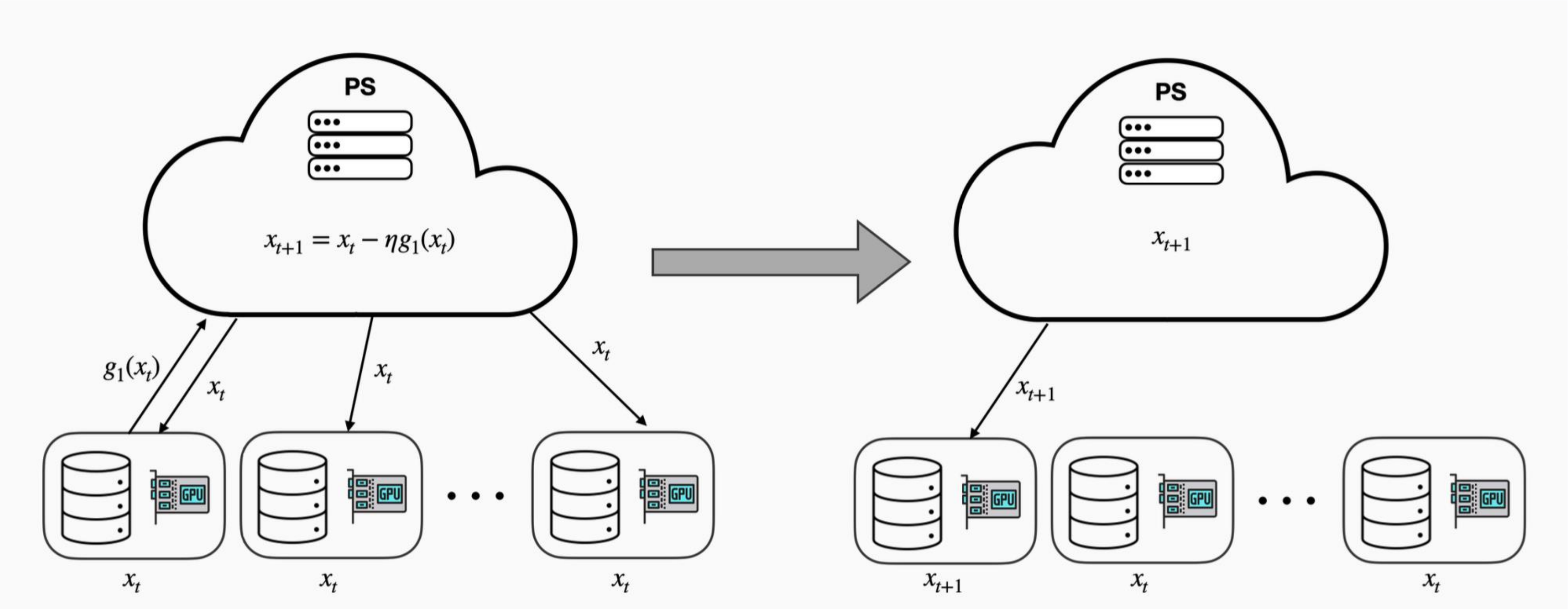
Parallel Assynchronou SGD

Idea: Do not **wait** for slow workers!





Parallel Assynchronou SGD

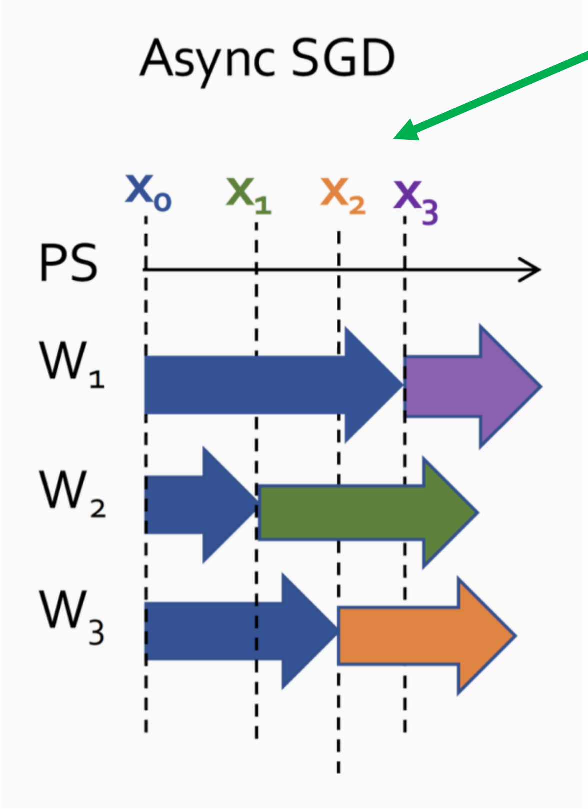


Each worker asynchronously does the following:

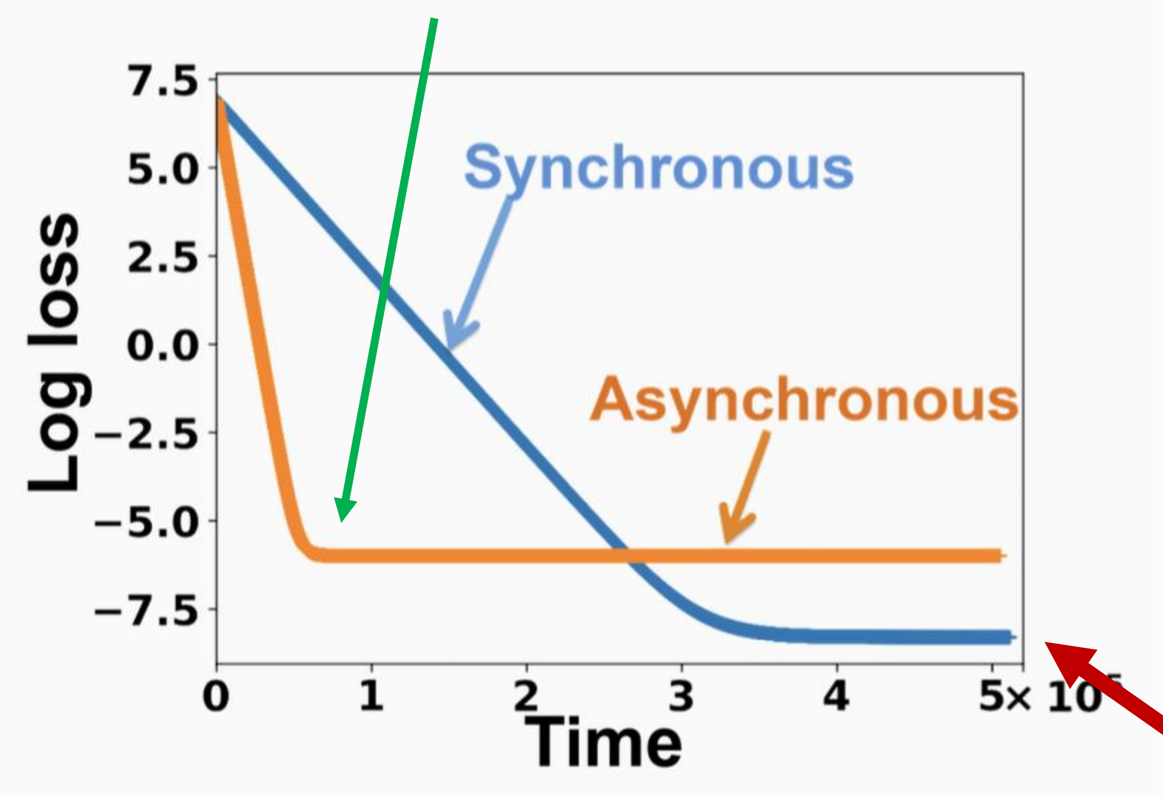
1. Pulls the current version x_t of the model
2. Computes a mini-batch gradient $g_i(x_t)$ and sends it to the PS

Each time the PS receives a gradient $g_i(x_{\tau_i(t)})$ where $\tau_i(t) \leq t$ from a worker it updates the model as $x_{t+1} = x_t - \eta\, g_i(x_{\tau_i(t)})$

Parallel Assynchronou SGD



Benefit: Faster updates!



Main Drawback: Stale updates

Reducing volume of communication

What should we expect from this compressor?

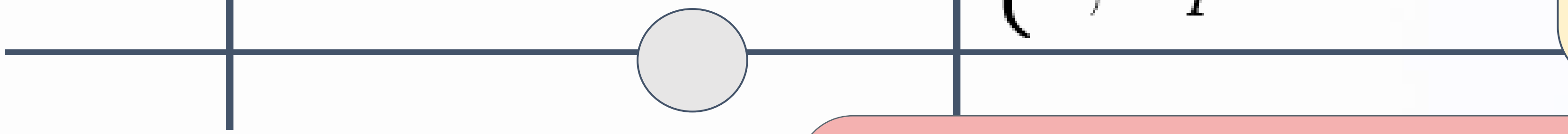
Unbiased random compressor $C(x)$

$$E[\mathcal{C}(x)] = x$$

$$E[\|\mathcal{C}(x) - x\|^2] \leq \omega \|x\|^2$$

Compressions are used to minimize the volume of communication

$$\mathcal{C}(0.75) = \begin{cases} 0, & p = 0.25 \\ 1, & p = 0.75 \end{cases}$$

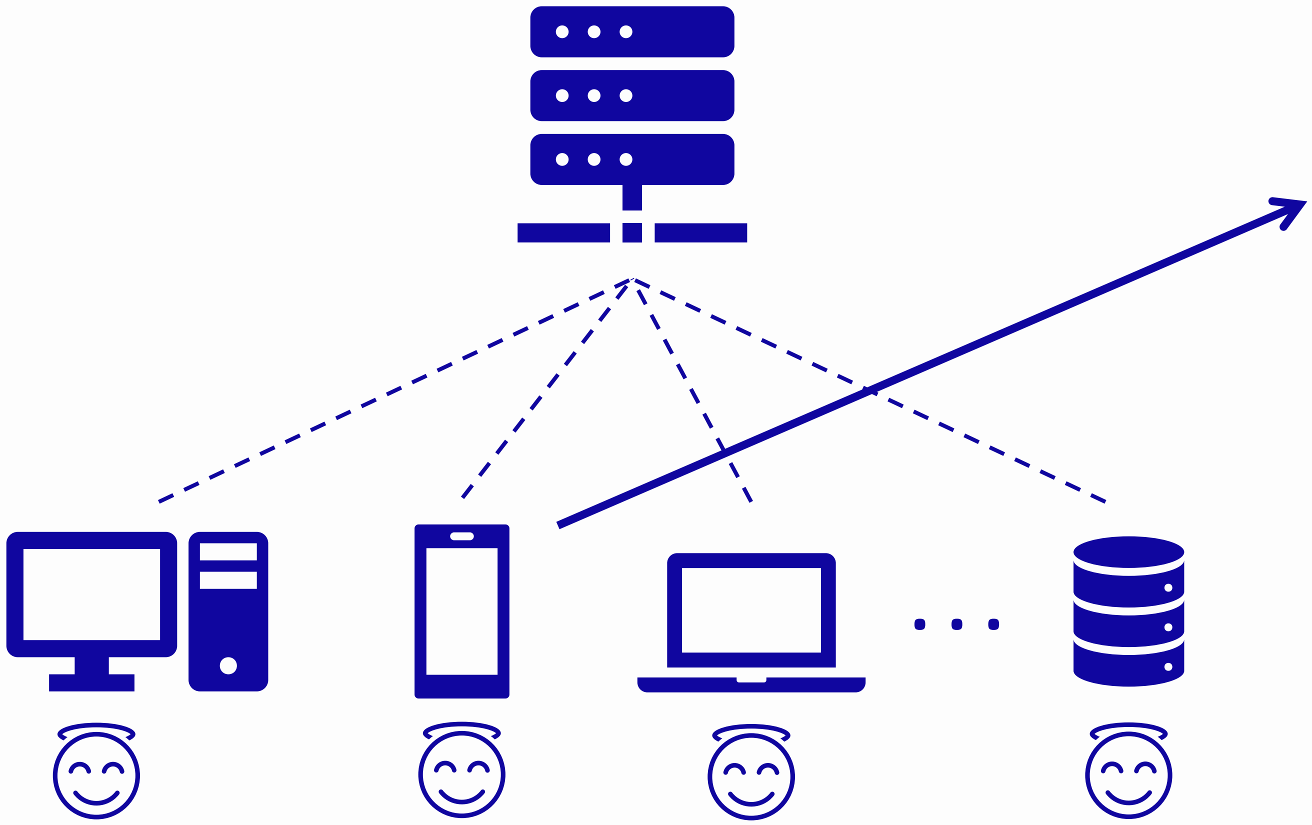


0

Assume $x_i \in [0,1]$ and $n = 10^6$ workers and let $M = \frac{1}{n}\sum_{i=1}^{n} C(x_i)$.
We have $\bar{x} = E[M] = \frac{1}{n}\sum_{i=1}^{n} x_i$ and
$Var[M] = \frac{1}{n^2}\sum_{i=1}^{n} V(x_i) \leq \frac{1}{n} max_{p\in[0,1]}\, p(1-p) \leq \frac{1}{4n}$.

Privacy Concerns, Federated Learning and Applications

Federated Learning

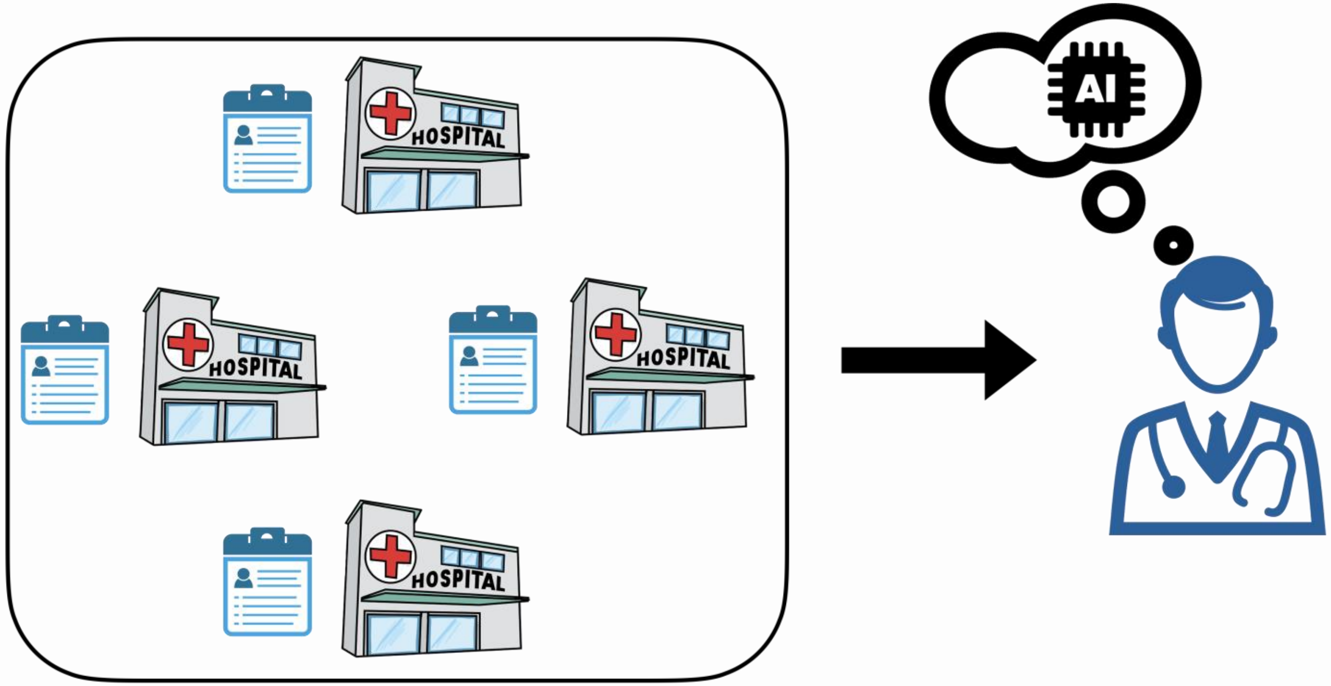


- Would you share your private data on your phone (emails, photos, …) ?
- How could we train faster with less ammount of communication?

Types of Federated Learning

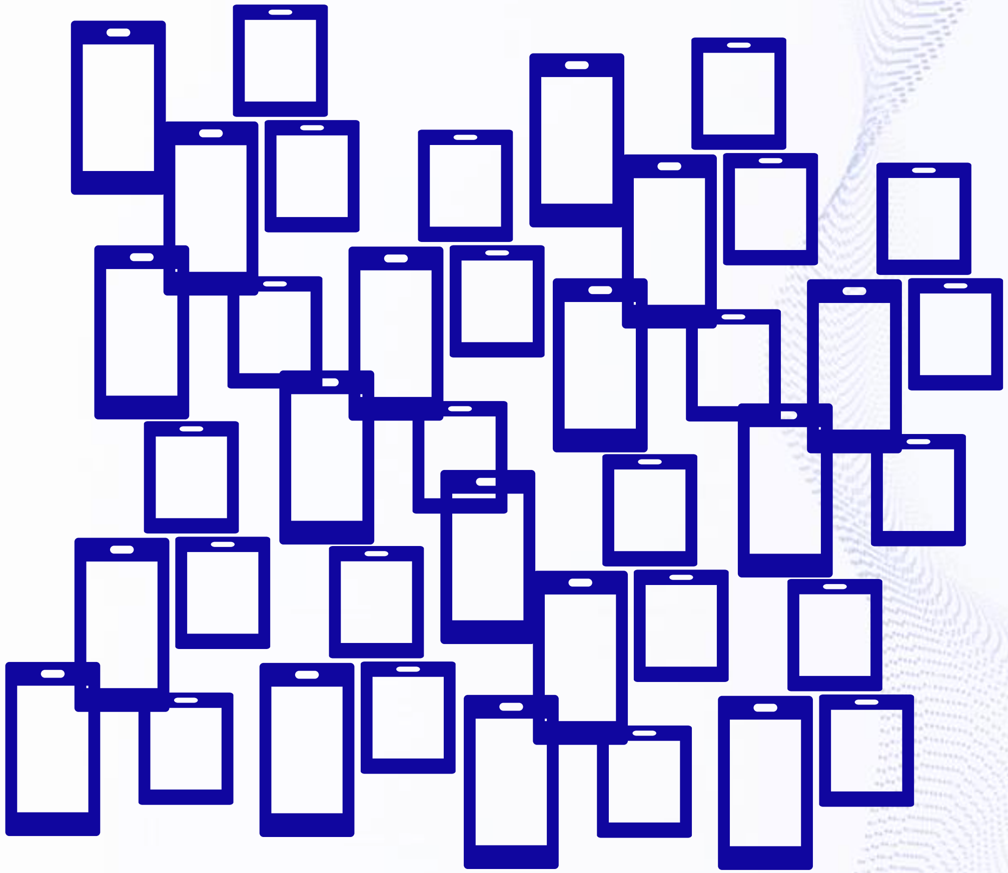
cross-silo FL

collaborative learning among several organizations



cross-device FL

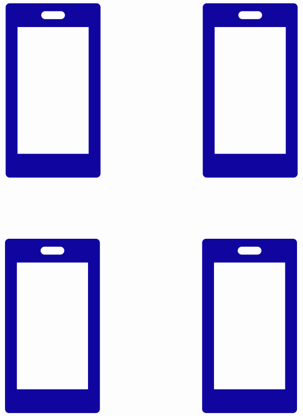
large populations of mobile devices



Types of Federated Learning

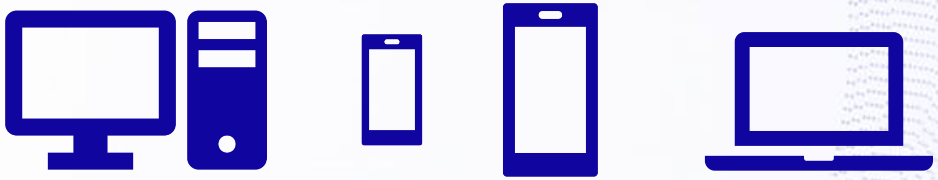
homogeneous FL

- data across devices come from the same distribution
- all computing devices are the same



heterogeneous FL

large populations of mobile devices

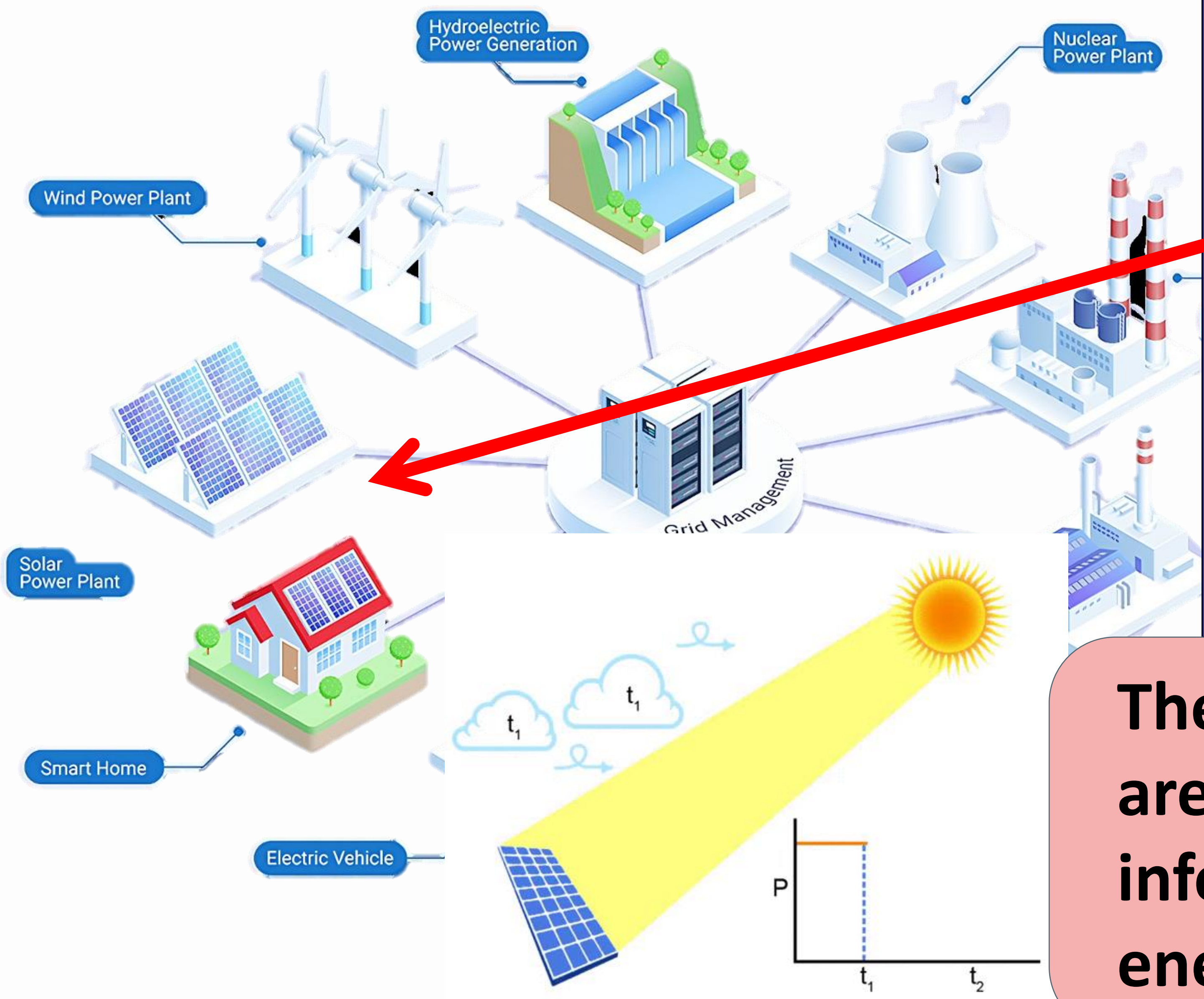


Applications of FL – Use Cases

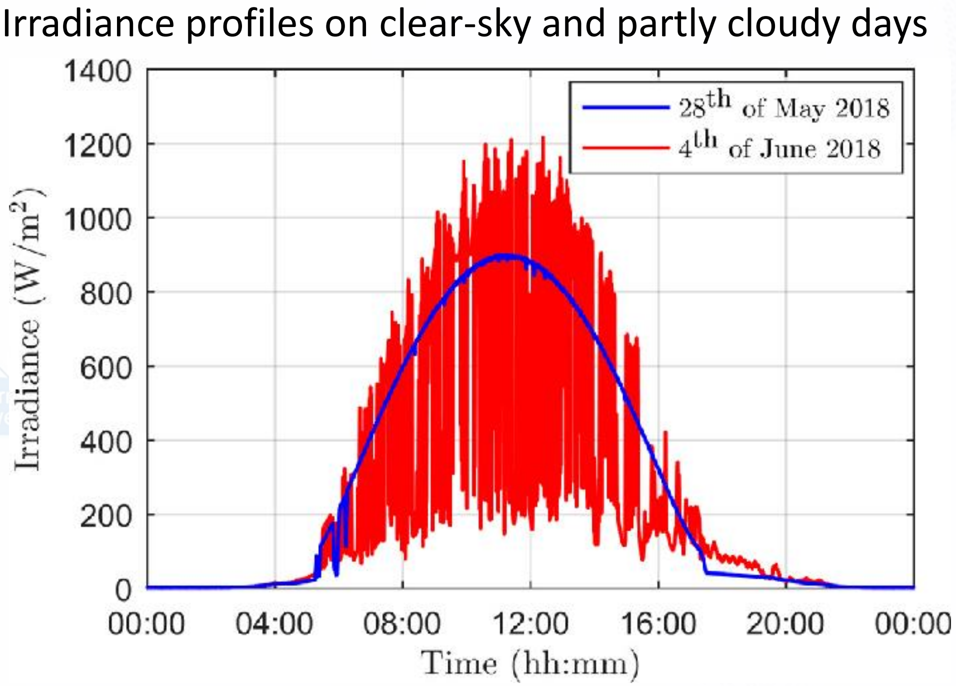
- **Commercial applications already in production:**
  - Apple: “Hey Siri”, QuickType
  - Google: “Hey Google”, Gboard
- **Next Game Changer for:**
  - Smart Health Applications: Medical Research and Diagnosis (doc.ai, Owkin)
  - FinTech Applications: Fraud Detection (WeBank)

Are there serioius consequences for Gboard’s wrong predictions?

Learning demand and generation profiles



Source: http://code.eng.buffalo.edu/cloud/index.html



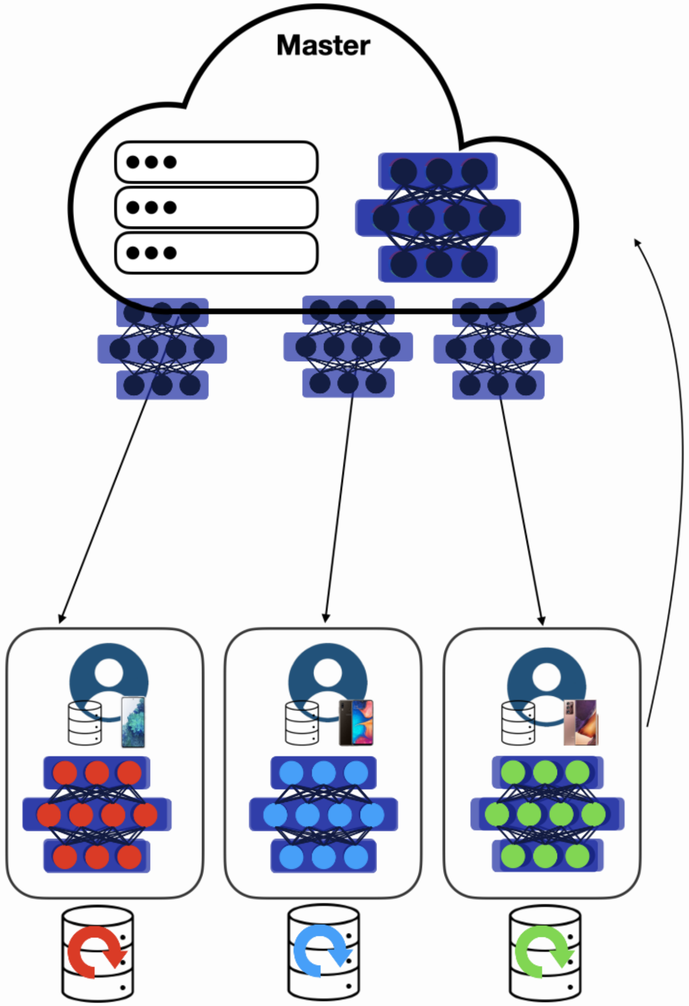
Markku Jarvela, Kari Lappalainen, and Seppo Valkealahti. Characteristics of the cloud enhancement phenomenon and pv power plants. Solar Energy, 196:137–145, 2020.

The generation and demands are private but crucial information for efficient energy grid!

Federated Learning Learning Algorithms

training the FL problems efficiently

Federated Averaging - FedAvg



- Repeat Until Convergence:

1. Global model is sent to available devices
2. Devices train local models on local data (local epochs)
3. Devices send the updates back
4. Aggregation step and global model update

Federated Averaging - FedAvg

MBZUAI

Algorithm 1: Generalized FedAvg (also known as FedOpt [211])

Input: Initial model $\boldsymbol{x}^{(0)}$; ClientOpt, ServerOpt with learning rate η, η_s

for $t \in \{0, 1, \ldots, T-1\}$ **do**
    Sample a subset $\mathcal{S}^{(t)}$ of clients
    for client $i \in \mathcal{S}^{(t)}$ **in parallel do**
        Initialize local model $\boldsymbol{x}_i^{(t,0)} = \boldsymbol{x}^{(t)}$
        for $k = 0, \ldots, \tau_i - 1$ **do**
            Compute local stochastic gradient $g_i(\boldsymbol{x}_i^{(t,k)})$
            Perform local update $\boldsymbol{x}_i^{(t,k+1)} = \text{ClientOpt}(\boldsymbol{x}_i^{(t,k)}, g_i(\boldsymbol{x}_i^{(t,k)}), \eta, t)$
        end
        Compute local model changes $\Delta_i^{(t)} = \boldsymbol{x}_i^{(t,\tau_i)} - \boldsymbol{x}_i^{(t,0)}$
    end
    Aggregate local changes $\Delta^{(t)} = \sum_{i \in \mathcal{S}^{(t)}} p_i \Delta_i^{(t)} / \sum_{i \in \mathcal{S}^{(t)}} p_i$
    Update global model $\boldsymbol{x}^{(t+1)} = \text{ServerOpt}(\boldsymbol{x}^{(t)}, -\Delta^{(t)}, \eta_s, t)$
end

We need to tune the number of local steps! Why?

arXiv:2107.06917

SCAFFOLD

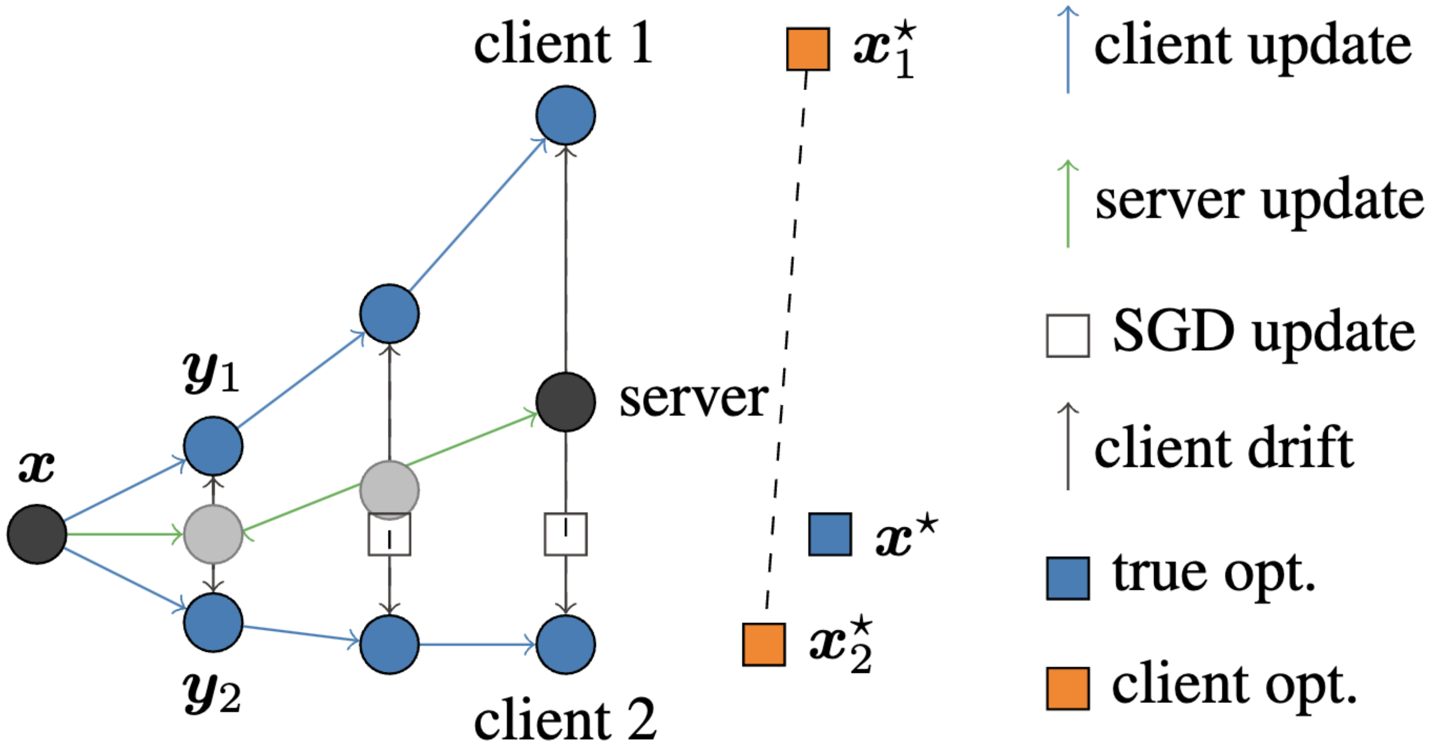


Figure 1. Client-drift in FEDAVG is illustrated for 2 clients with 3 local steps ($N = 2$, $K = 3$). The local updates $\boldsymbol{y}_i$ (in blue) move towards the individual client optima $\boldsymbol{x}_i^\star$ (orange square). The server updates (in red) move towards $\frac{1}{N}\sum_i \boldsymbol{x}_i^\star$ instead of to the true optimum $\boldsymbol{x}^\star$ (black square).

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh: *SCAFFOLD: Stochastic controlled averaging for on-device federated learning*, ICML 2020.

SCAFFOLD

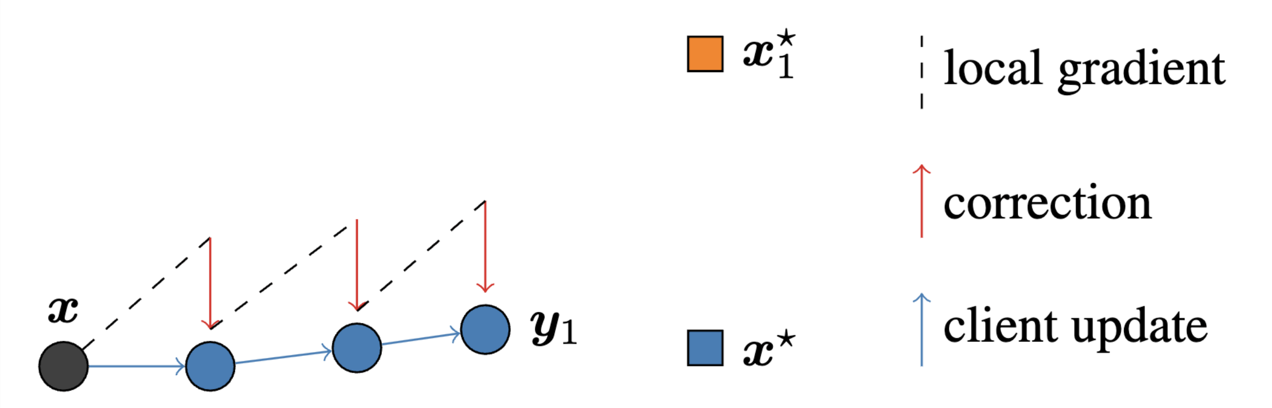


Figure 2. Update steps of SCAFFOLD on a single client. The local gradient (dashed black) points to $x_1^\star$ (orange square), but the correction term $(c - c_i)$ (in red) ensures the update moves towards the true optimum $x^\star$ (black square).

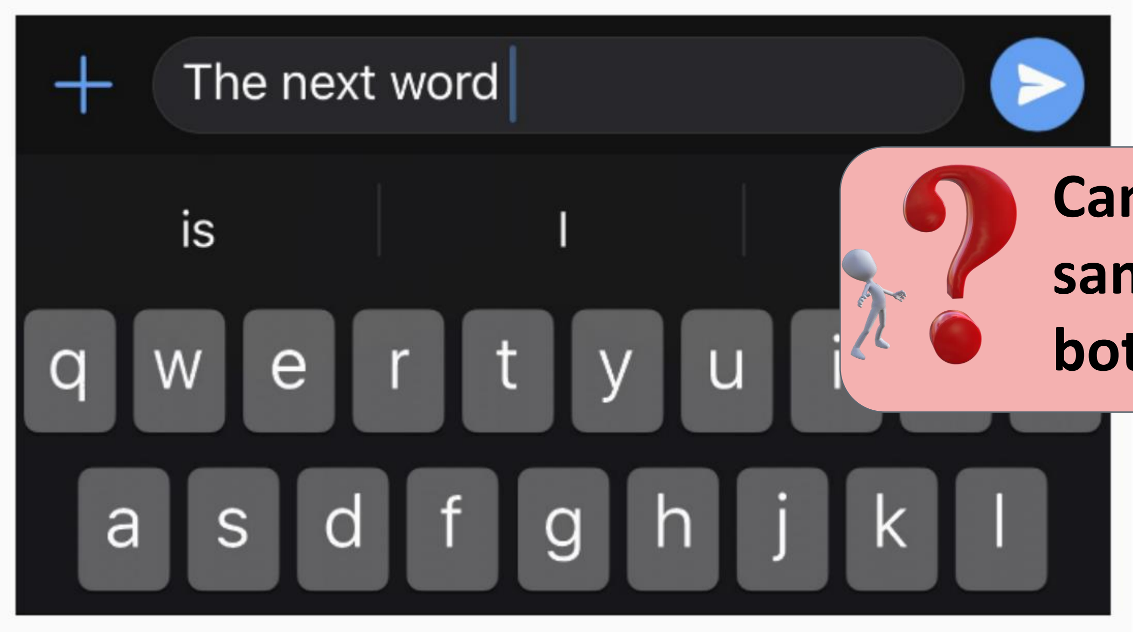
Algorithm 1 SCAFFOLD: Stochastic Controlled Averaging for federated learning

server input: initial x and c, and global step-size η_g
client i's input: c_i, and local step-size η_l
for each round $r = 1, \dots, R$ **do**
  sample clients $\mathcal{S} \subseteq \{1, \dots, N\}$
  communicate (x, c) to all clients $i \in \mathcal{S}$
  on client $i \in \mathcal{S}$ **in parallel do**
    initialize local model $y_i \leftarrow x$
    for $k = 1, \dots, K$ **do**
      compute mini-batch gradient $g_i(y_i)$
      $y_i \leftarrow y_i - \eta_l \left(g_i(y_i) - c_i + c\right)$
    end for
    $c_i^+ \leftarrow$ (i) $g_i(x)$, or (ii) $c_i - c + \frac{1}{K\eta_l}(x - y_i)$
    communicate $(\Delta y_i, \Delta c_i) \leftarrow (y_i - x, c_i^+ - c_i)$
    $c_i \leftarrow c_i^+$
  end on client
  $(\Delta x, \Delta c) \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (\Delta y_i, \Delta c_i)$
  $x \leftarrow x + \eta_g \Delta x$ and $c \leftarrow c + \frac{|\mathcal{S}|}{N} \Delta c$
end for

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh: *SCAFFOLD: Stochastic controlled averaging for on-device federated learning*, ICML 2020.

Personalized Federated Learning

Do we need personalization?



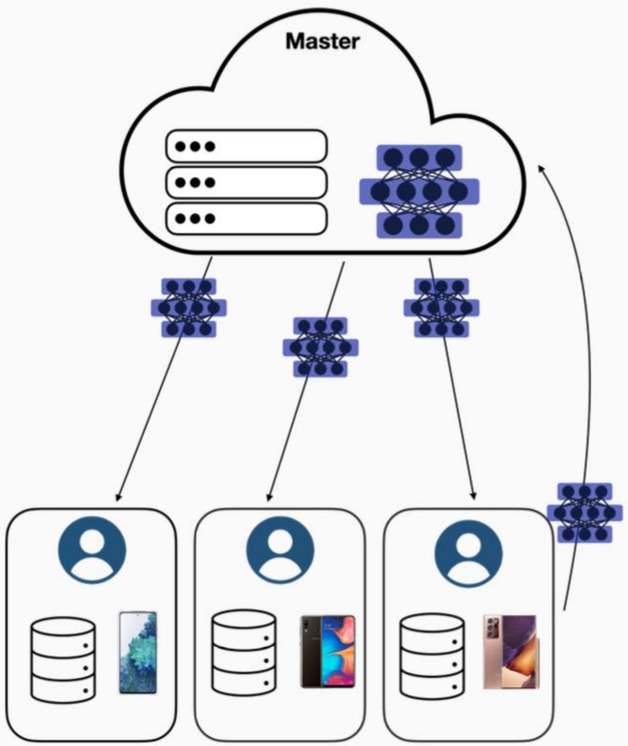
Can we have the same model for both of them?

Remedy: New personalized objectives

The Global-only Approach to Federated Learning

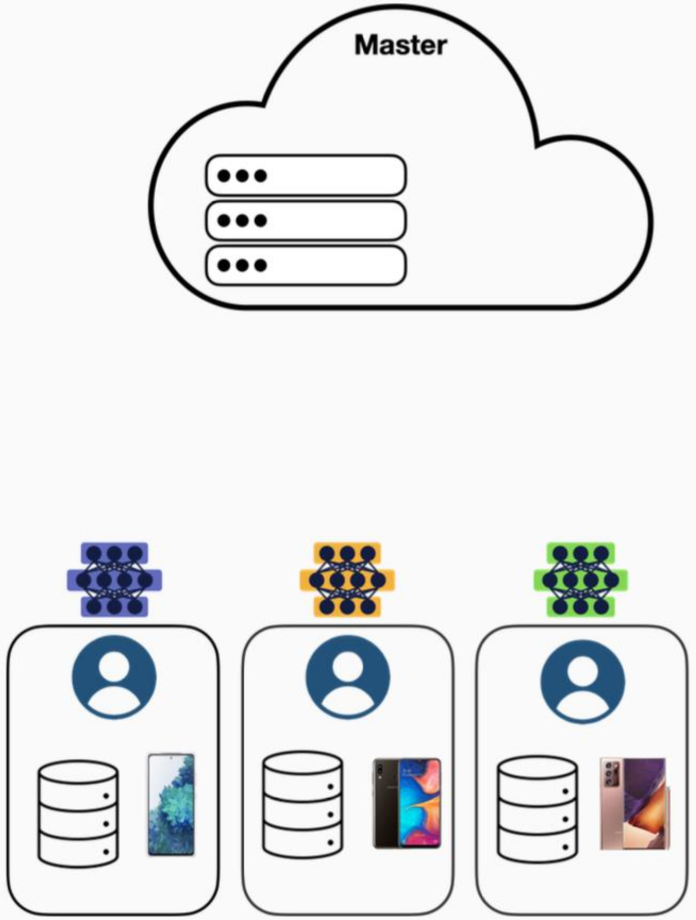
$$\min_{x\in\mathbb{R}^d} \left\{ f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\}$$

- In standard federated learning we learn a single global model x that captures all the union of the local training datasets at the client
- The global model might **not** work well for minority clients who have rare data
- Such clients may want to learn **personalized models** that are customized to their datasets



The Local-only Approach to Federated Learning

$$\min_{x_1,x_2,\ldots,x_n} \frac{1}{n}\sum_{i=1}^{n} f_i\left(x_i\right)$$



- One may take a local-only approach, where each client trains a model x_i in isolation, using its local dataset D_i
- The dataset at each client may be **too small** to learn an accurate model, and generalization suffer
- For instance, it may be beneficial to average the personalized models across similar clients (clustering and training models withing a cluster)

There is a whole spectrum of approaches between the global-only and local-only extremes

Personalized FL Objective

$$\min_{x} \frac{1}{n} \sum_{i=1}^{n} f_i\,(x)$$

Global-only Objective

$$\min_{x_1, x_2, \ldots, x_n} \frac{1}{n} \sum_{i=1}^{n} f_i\,(x_i)$$

Local-only Objective

- What should be the objective function of learning personalized models that **generalize better** than the local-only approach?
- Some combination of the local and global objective functions?

Personalized FL Objective - Clustering

- Building on the insight that it is beneficial to coordinate with similar clients, suppose we decide to cluster the clients that K clusters, and learn a model x_k $(k \in \{1,2,\ldots,K\})$ for each cluster
- How to do the clustering?
- **Does the problem reduce to K separate federated learning systems?**

- **Idea 1:** Cluster the clients based on their local data

HOW?

- **Problem:** Data cannot be shared across clients due to privacy concerns (maybe we can cluster based on the public metadata, e.g., geographic location)

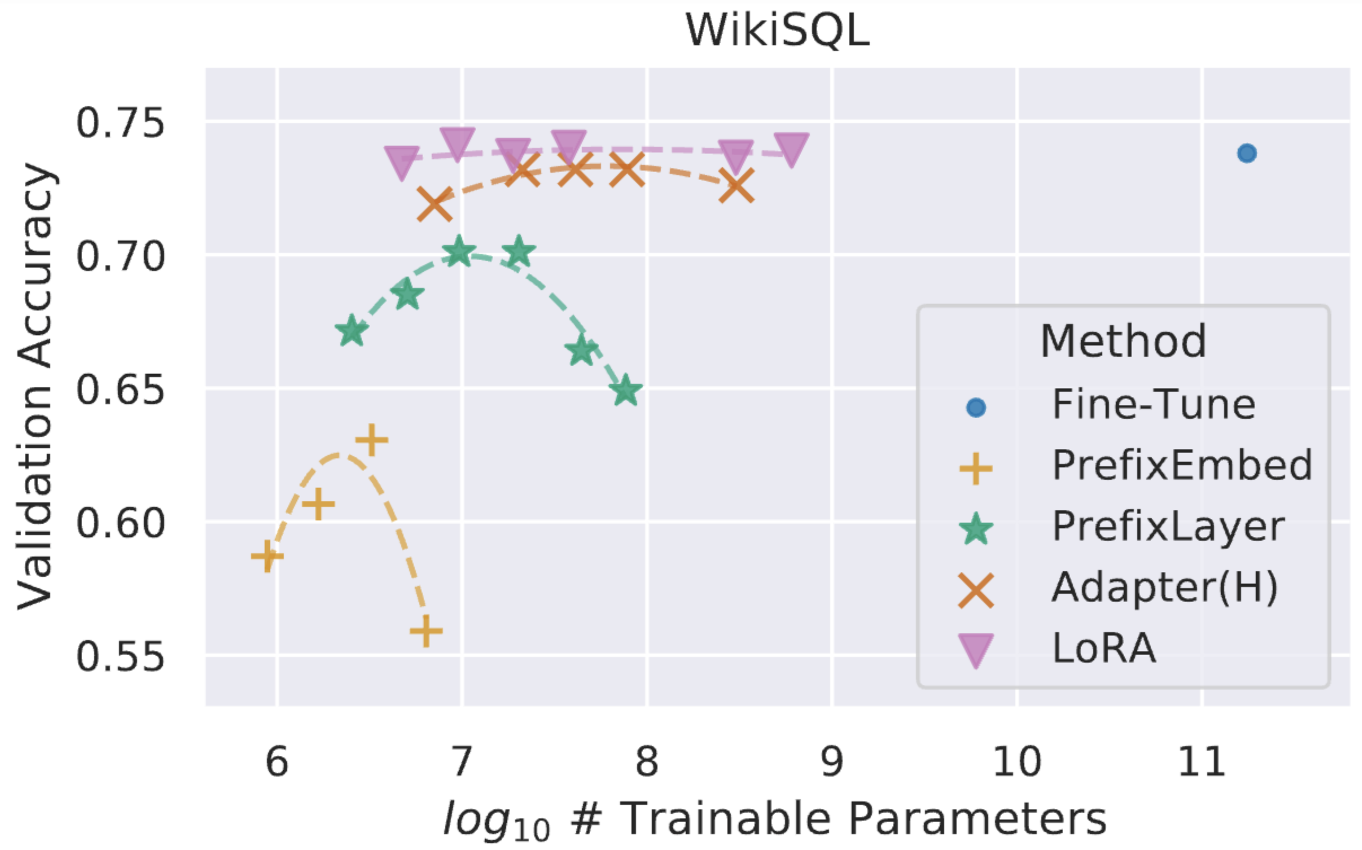
LoRA: Low-Rank Adaptation of Large Language Models

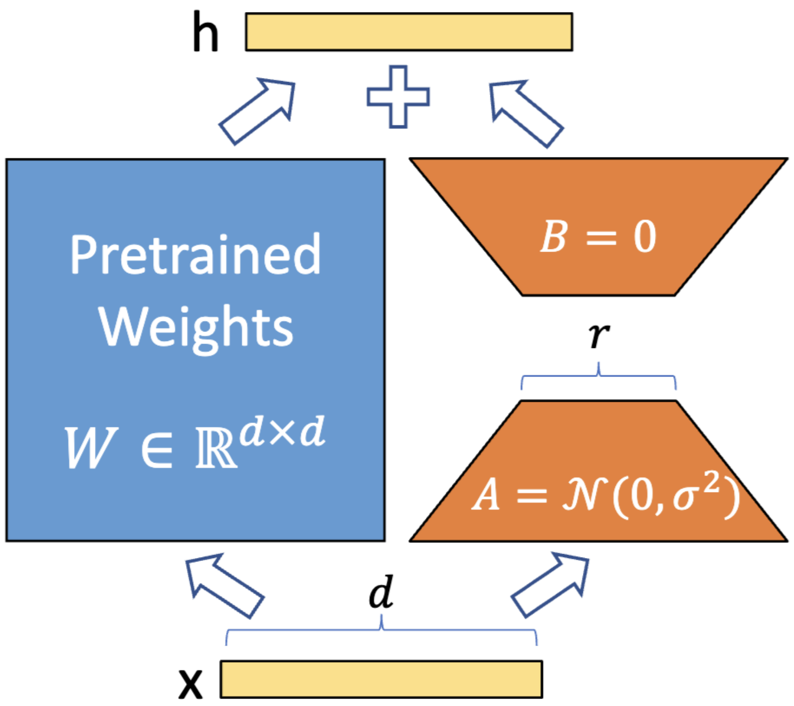
Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, ICLR 2022

LoRA

Example usage/features:

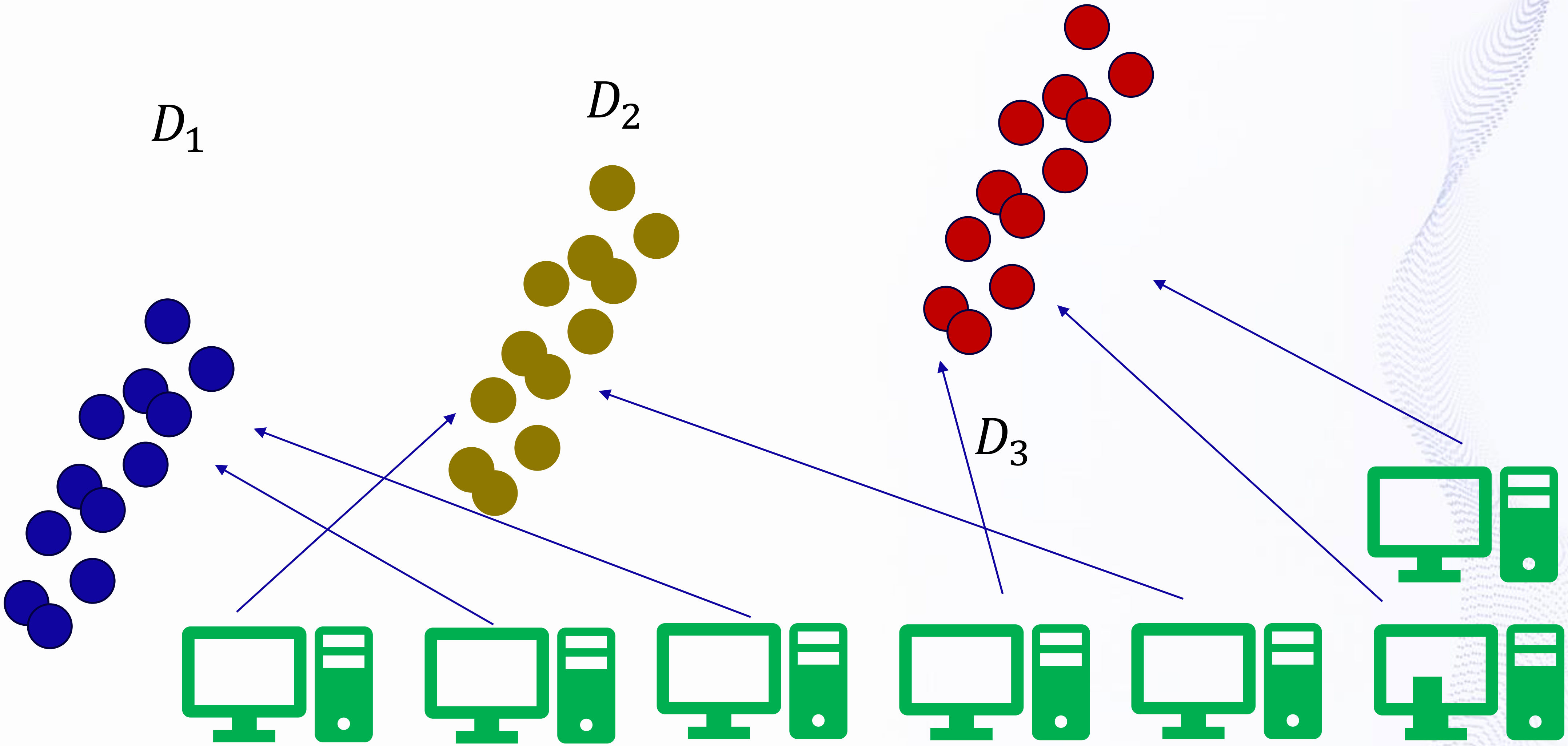
- fine-tuning with a low-rank adaptors
- optimizing over lower number of parameters
- low memory overhead for optimizers





Collaborative and Efficient Personalization with Mixtures of Adaptors

- assume the FL task but with a twist that we have n workers such that each worker belong to 1 of data groups $\in \{D_1, D_2, \dots, D_K\}$ (i.e. multi-task learning)
- **GOAL**: learn only a LoRA adaptors in FL way



Collaborative and Efficient Personalization with Mixtures of Adaptors

- we allow only a low-rank (adaptor) personalization

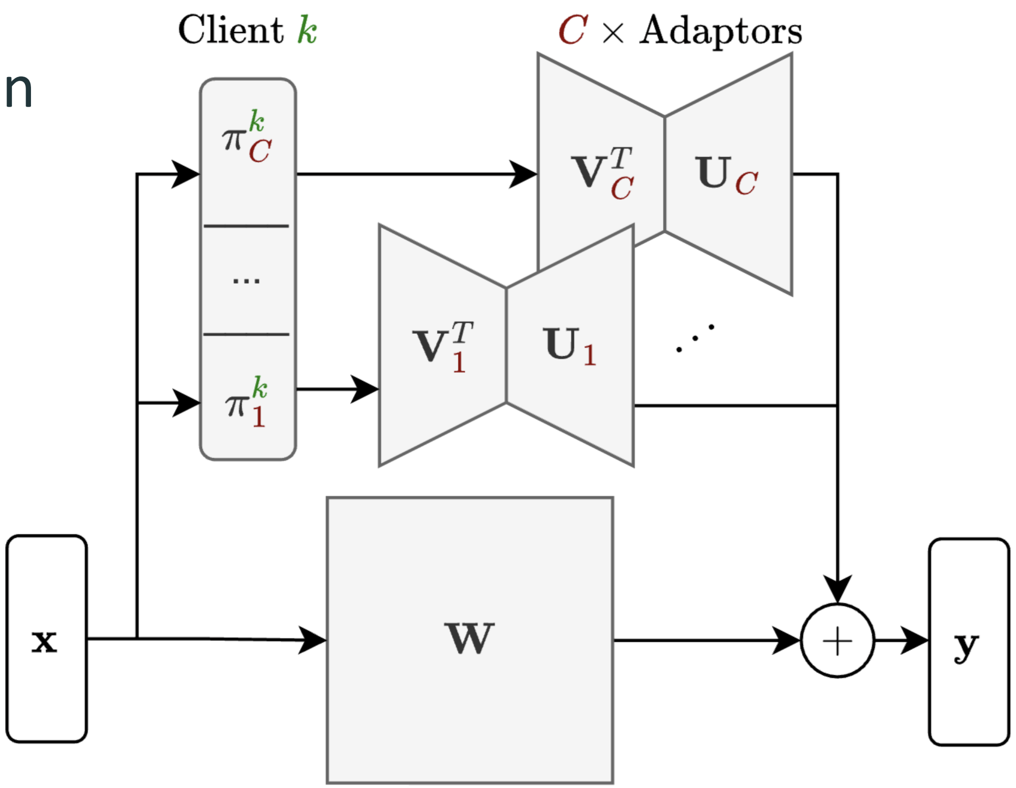
$$W_i = W + \sum_i^K \pi_i^j U_j V_j^T$$

weight for worker i

global weight

mixture of low-rank adaptors

for each worker:
$\pi_i \in \Delta \subset R^K$



Collaborative and Efficient Personalization with Mixtures of Adaptors, Abdulla Jasem Almansoori, Samuel Horváth, M.T., 2024

Formulation & Algorithm

$$\min_{\mathbf{u},\{\mathbf{a}_c\}_{c=1}^{C},\{\boldsymbol{\pi}^k\}_{k=1}^{K}} \quad \sum_{c=1}^{C} \mathbb{E}_{k\sim\mathcal{K}}\left[\boldsymbol{\pi}_c^k f^k(\mathbf{u},\mathbf{a}_c)\right] \qquad \text{(MFL-WS)}$$

$$\text{s.t.} \quad \boldsymbol{\pi}^k \in \Delta^{C-1}, \forall k \in [K].$$

Algorithm 1 Simple FLoRAL Averaging

1: Let $\mathbf{w}_{c,t}^k = (\mathbf{u}_t^k, \mathbf{a}_{c,t}^k)$
2: **for** $\tau = 0, H, 2H, \cdots, \lfloor \frac{T-1}{H} \rfloor$ **do** ▷ Comm. rounds
3: Sample clients $S_\tau \sim \mathcal{K}$
4: **for all** $k \in S_\tau$ **in parallel do**
5: **for** $t = \tau, \cdots, \tau + H - 1$ **do** ▷ Local epoch
6: $\hat{\boldsymbol{\pi}}_{c,t}^k = \frac{\exp(\theta_{c,t}^k)}{\sum_{c=1}^{C}\exp(\theta_{c,t}^k)}$
7: $\theta_{c,t+1}^k = \theta_{c,t}^k - \eta_t \nabla_{\theta_{c,t}^k} f^k(\sum_{c=1}^{C} \hat{\boldsymbol{\pi}}_{c,t}^k \mathbf{w}_{c,t}^k)$
8: $\mathbf{w}_{c,t+1}^k = \mathbf{w}_{c,t}^k - \eta_t \nabla_{\mathbf{w}_{c,t}^k} f^k(\sum_{c=1}^{C} \hat{\boldsymbol{\pi}}_{c,t}^k \mathbf{w}_{c,t}^k)$
9: **end for**
10: **end for**
11: $\mathbf{u}_{\tau+H}^k \leftarrow \frac{\sum_{k\in S_\tau} N^k \mathbf{u}_{\tau+H}^k}{\sum_{k\in S_\tau} N^k}$ ▷ Synchronize base layers
12: $\mathbf{a}_{c,\tau+H}^k \leftarrow \frac{\sum_{k\in S_\tau} \hat{\boldsymbol{\pi}}_{c,\tau+H}^k N^k \mathbf{a}_{c,\tau+H}^k}{\sum_{k\in S_\tau} \hat{\boldsymbol{\pi}}_{c,\tau+H}^k N^k}$ ▷ Synchronize adaptors
13: **end for**

Collaborative and Efficient Personalization with Mixtures of Adaptors, Abdulla Jasem Almansoori, Samuel Horváth, M.T., 2024

Experiments

Method	π^*	MNIST				CIFAR-10			
		Full		Reduced		Full		Reduced	
		R	LS	R	LS	R	LS	R	LS
FedAvg		91.5$_{0.6}$	25.8$_{2.4}$	78.2$_{0.6}$	23.2$_{0.9}$	64.4$_{0.3}$	21.9$_{0.4}$	45.6$_{0.3}$	18.7$_{0.4}$
Local Adaptor		86.6$_{0.3}$	84.5$_{1.8}$	47.4$_{5.4}$	32.0$_{2.3}$	66.3$_{0.5}$	68.8$_{0.5}$	33.5$_{0.5}$	30.8$_{0.8}$
Ensemble	✗	92.0$_{0.1}$	93.8$_{0.5}$	66.7$_{5.3}$	86.4$_{0.4}$	*71.0*$_{2.8}$	46.4$_{9.2}$	42.4$_{0.9}$	41.7$_{4.6}$
Ensemble	✓	**95.8**$_{0.3}$	**95.6**$_{0.3}$	**88.2**$_{1.4}$	**87.6**$_{1.3}$	**73.7**$_{0.2}$	**73.3**$_{0.1}$	45.0$_{0.9}$	**45.1**$_{0.8}$
FLoRAL(1%)	✗	91.3$_{0.6}$	89.7$_{3.2}$	73.1$_{3.7}$	46.0$_{9.9}$	65.5$_{0.4}$	62.8$_{8.8}$	45.2$_{0.3}$	*44.2*$_{0.9}$
FLoRAL(1%)	✓	93.9$_{0.8}$	93.7$_{0.2}$	*87.5*$_{2.1}$	**87.6**$_{0.5}$	68.9$_{0.2}$	*72.2*$_{0.2}$	**47.8**$_{0.9}$	44.1$_{0.6}$
FLoRAL(10%)	✗	91.8$_{1.0}$	93.1$_{0.9}$	75.7$_{2.3}$	70.8$_{7.1}$	65.1$_{0.3}$	56.2$_{5.5}$	44.5$_{0.4}$	42.1$_{0.2}$
FLoRAL(10%)	✓	*94.5*$_{0.6}$	*94.2*$_{0.2}$	87.0$_{0.7}$	*86.9*$_{0.5}$	69.3$_{0.5}$	72.1$_{0.5}$	*47.2*$_{0.3}$	42.7$_{0.3}$

MeritFed: Merit-Based Federated Learning For Diverse Datasets

…allows FL agent to find whose updates are beneficial for training ML model

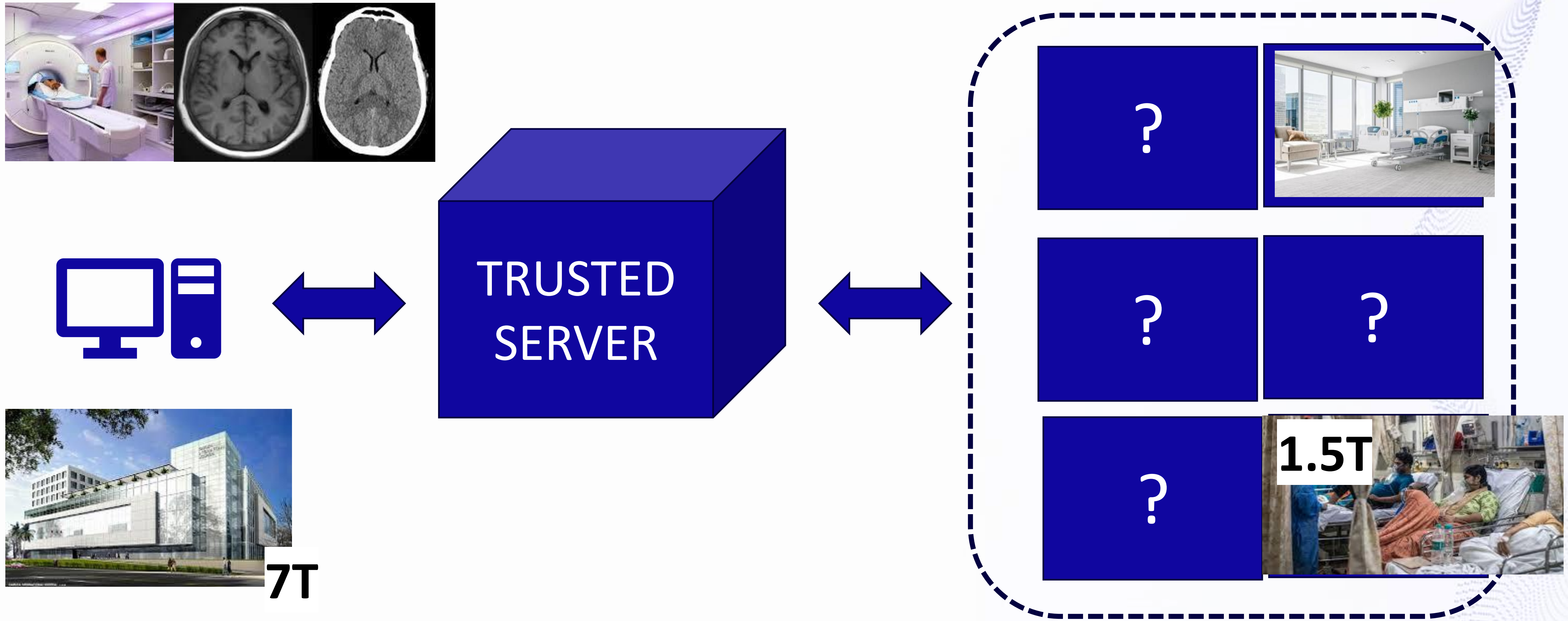
Federated Learning Can Find Friends That Are Advantageous

Nazarii Tupitsa, Samuel Horváth, MT, Eduard Gorbunov, 2024



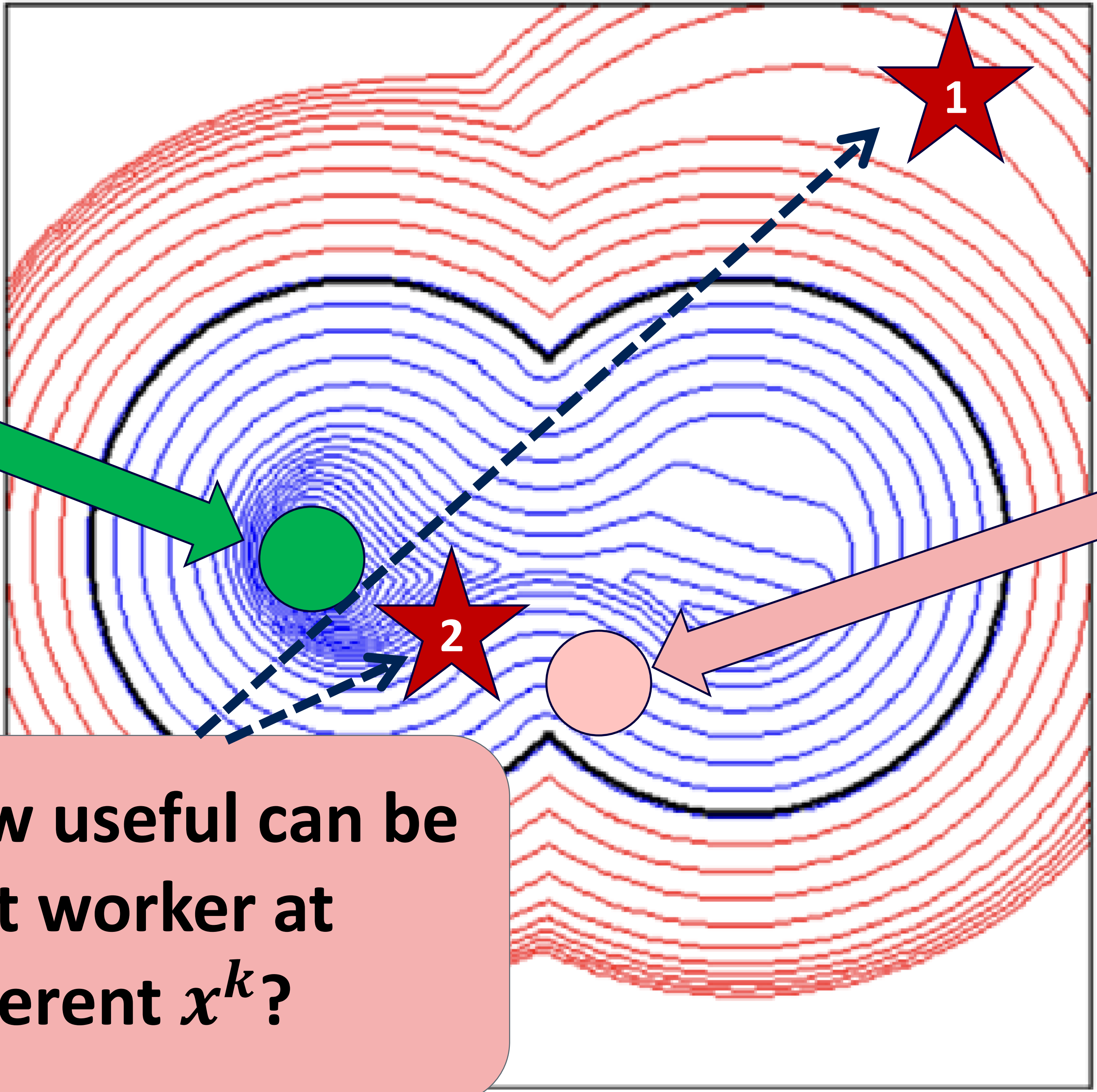
Collaboration as a service

- workers are available for collaboration for a fee (they do not care about training model for their use, just to utilize their data for profit)!



Federated Learning Can Find Friends That Are Advantageous, arxiv 2402.05050

MeritFed



our optimal solution

some worker solution

How useful can be that worker at different x^k?

MeritFed

MBZUAI

our testing loss!

The Bi-level optimization formulation

$$\min_{w \in \Delta} f(x^*(w)) \qquad x^*(w) \in \operatorname{argmin}_{x \in R^d} \sum_{i=1}^{n} w_i f_i(x)$$

Algorithm 1 `MeritFed`: Merit-based Federated Learning for Diverse Datasets

---

Input: Starting point $x^0 \in \mathbb{R}^d$, stepsize $\gamma > 0$
for $t = 0, \ldots$ **do**
  server sends x^t to each worker
  for all workers $i = 1, \ldots, n$ **in parallel do**
    compute stochastic gradient $g_i(x^t, \boldsymbol{\xi}_i)$ from local data and **send** $g_i(x^t, \boldsymbol{\xi}_i)$ to the server
  end for
  $$w^{t+1} \approx \arg\min_{w \in \Delta_1^n} f\left(x^t - \gamma \sum_{i=1}^{n} w_i g_i(x^t, \boldsymbol{\xi}_i)\right)$$
  $$x^{t+1} = x^t - \gamma \sum_{i=1}^{n} w_i^{t+1} g_i(x^t, \boldsymbol{\xi}_i).$$
end for

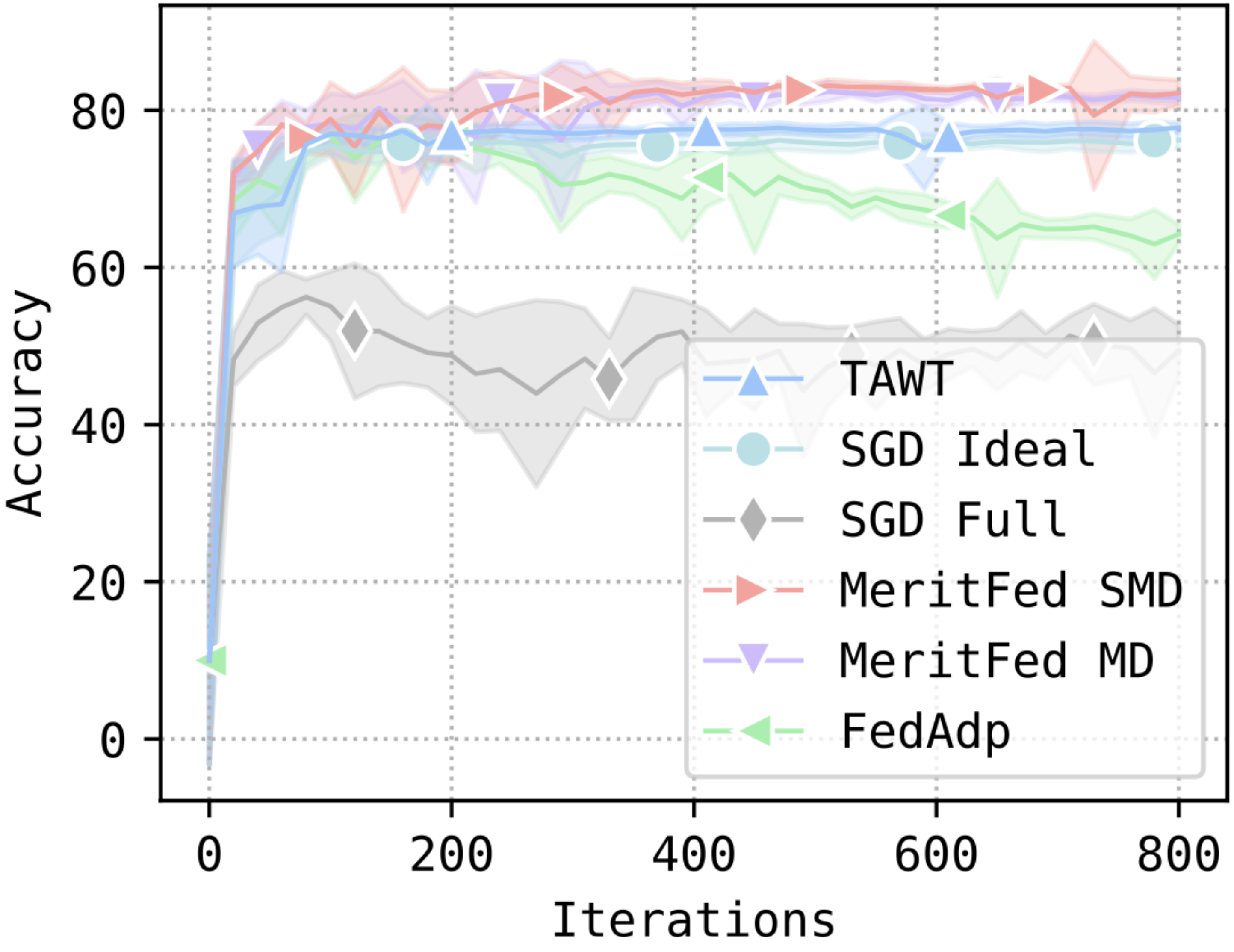
---

finding the best allocation to workers

use zeroth-order Mirror Descent (or its accelerated version)
Duchi et al. (2015); Shamir (2017); Gasnikov et al. (2022):

How can we do it?

MeritFed - Experiments



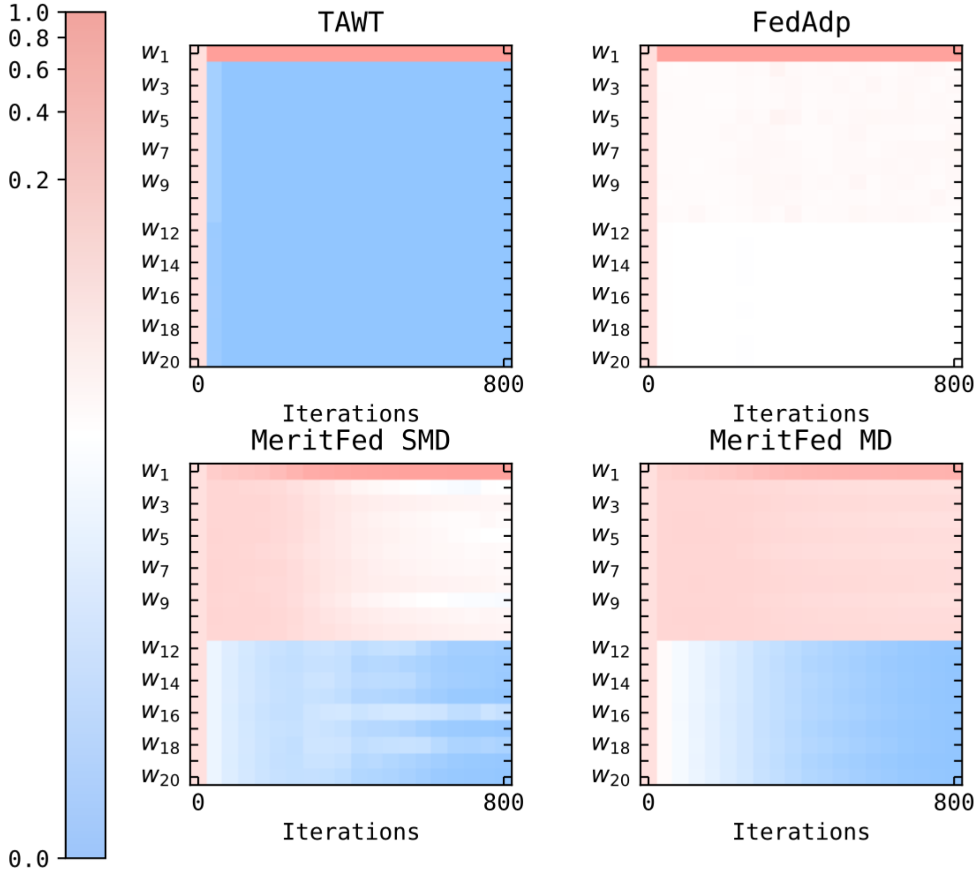


Figure 6: CIFAR10 (extra val.): $\alpha = 0.9$

MeritFed - Experiments

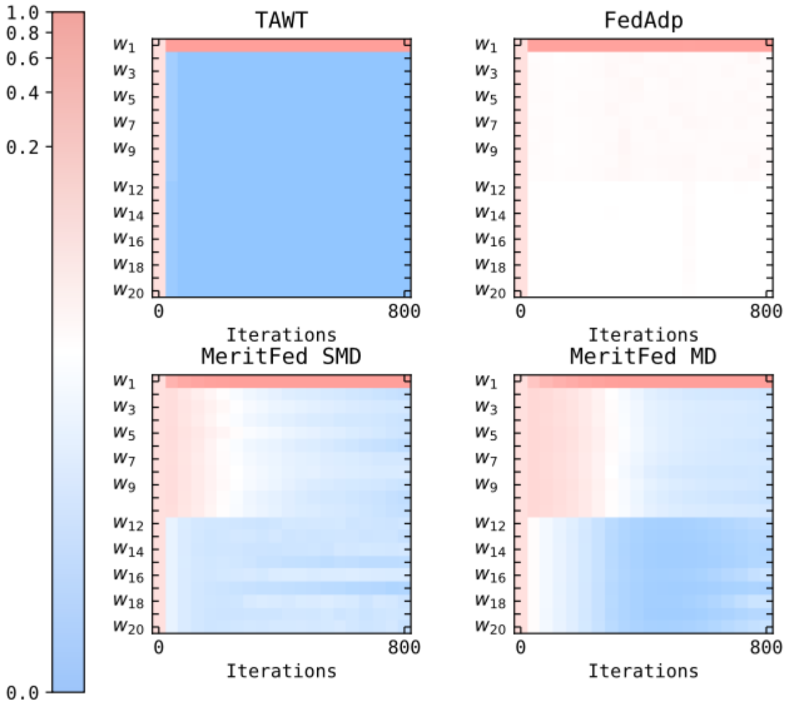


Figure 5: CIFAR10 (extra val.): $\alpha = 0.7$

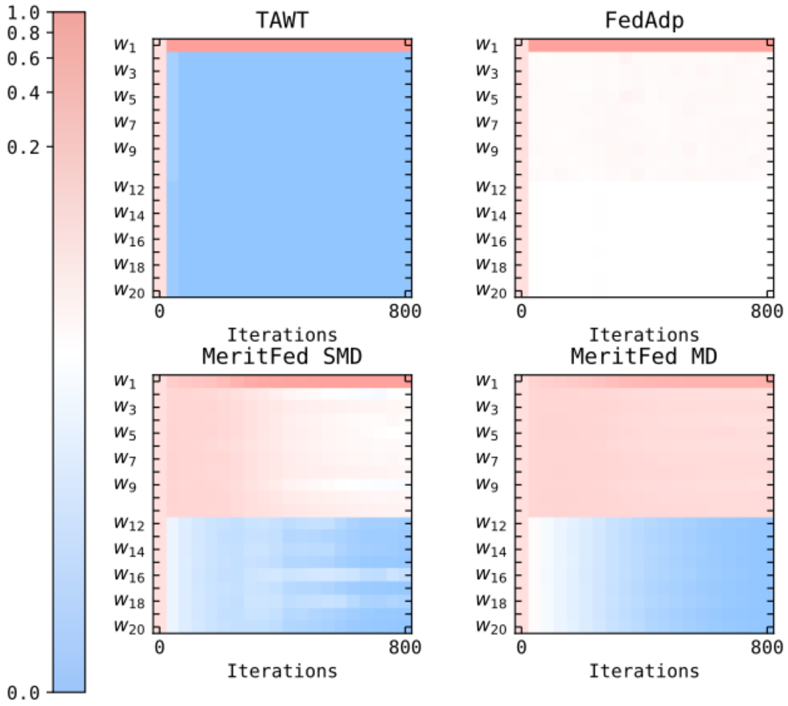


Figure 6: CIFAR10 (extra val.): $\alpha = 0.9$

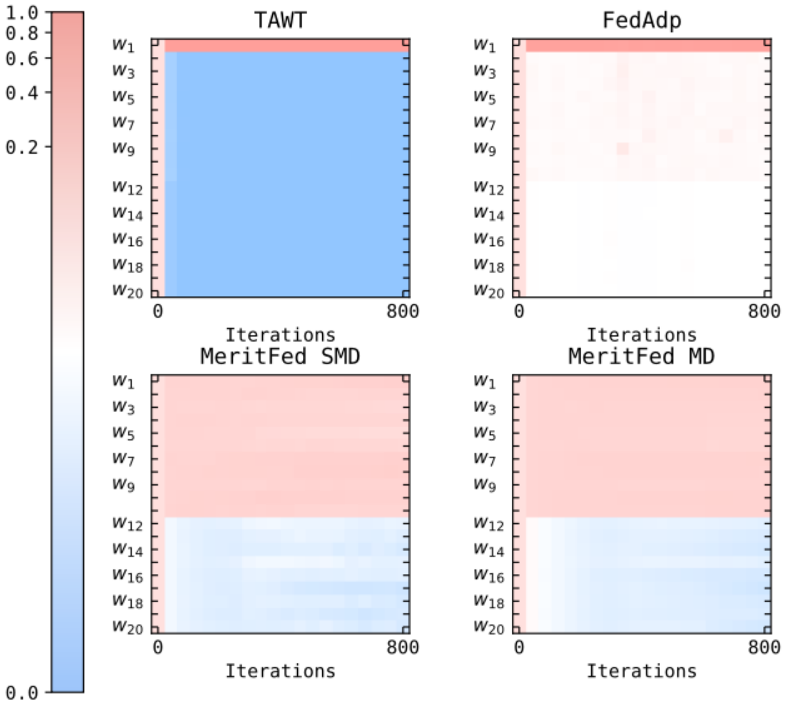


Figure 7: CIFAR10 (extra val.): $\alpha = 0.99$.

MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

Thank you

Mohamed bin Zayed University of Artificial Intelligence
Masdar City, Abu Dhabi, United Arab Emirates

mbzuai.ac.ae