FedMGDA+: Federated Learning Meets Multi-objective Optimization

Presenter: Zeou Hu

Joint work with Kiarash Shaloudegi, Guojun Zhang and Yaoliang Yu

PhD seminar at University of Waterloo,

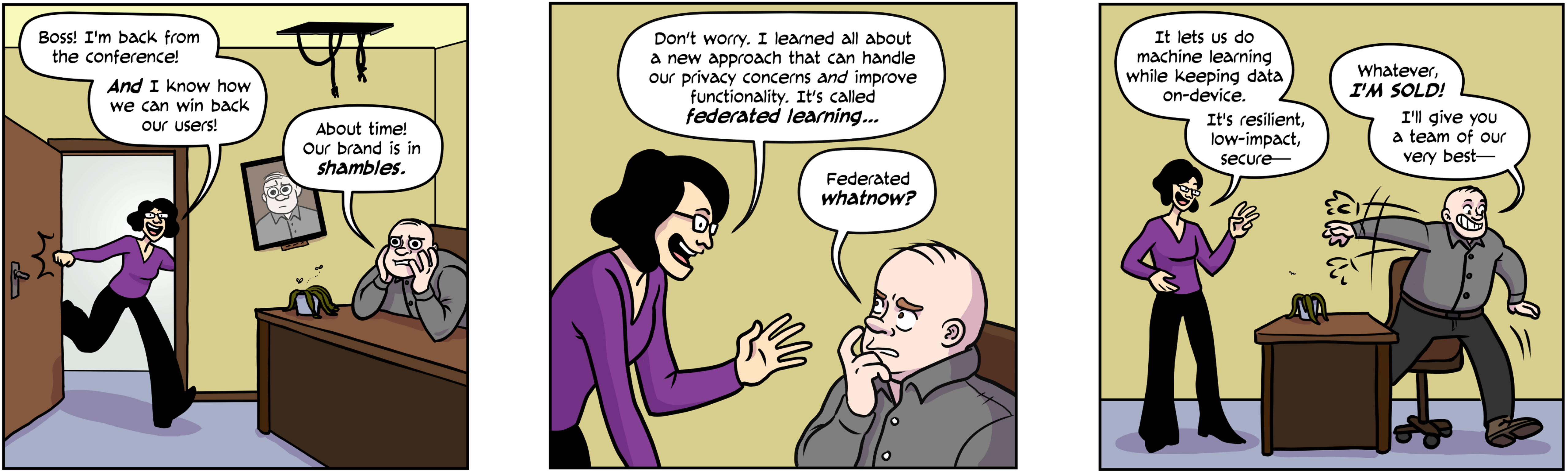
Cheriton School of Computer Science

Date: 24/June

UNIVERSITY OF WATERLOO

HUAWEI

Federated Learning



An online comic from Google AI
https://federated.withgoogle.com/

Motivation

- Large scale networks of connected devices provide access to an unprecedented amount of data.
- Smartphones, wearables, smart-homes, self-driving and ... collect data that are often private in nature.
- Traditional machine learning methods require all data to be collected in a central server.
- Several challenges in practice for collecting data:
  - Data privacy;
  - Data security;
  - Communication costs.

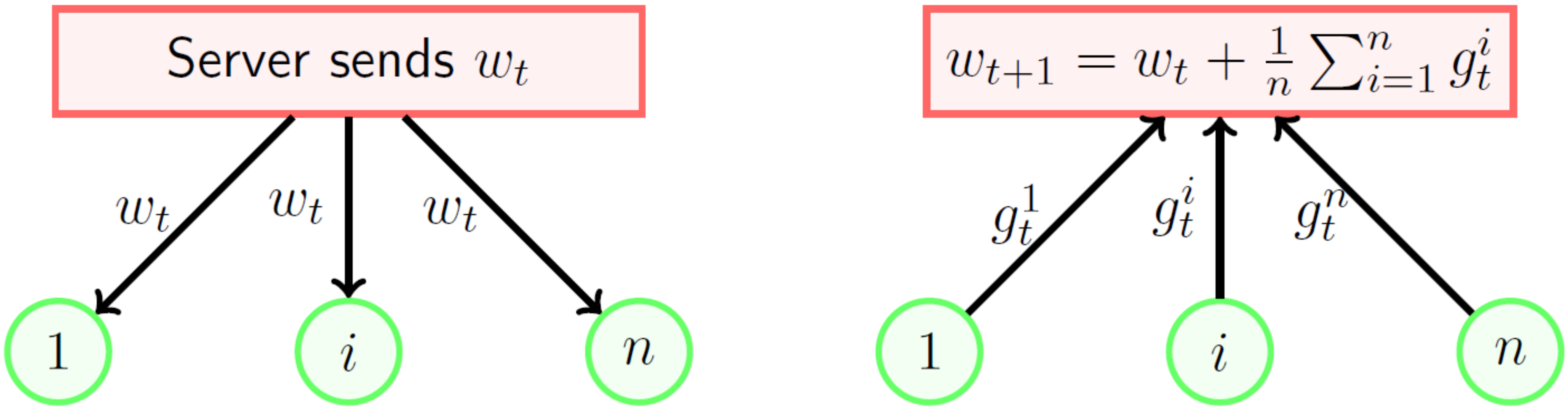
Background: Federated Learning

- Federated learning provides a platform for the edge devices to train a central model without sharing their local data.
- Federated learning has some unique features that distinguish it from the rest of distributed optimization problems:
  - Massively distributed;
  - Non-i.i.d. distribution of data;
  - Limited communication;
  - Unbalanced data.

Credit to (Li et al., 2019)

- Application: smartphones & terminal devices, networking traffic management, connected vehicles, and ...

Federated Learning - FedAvg

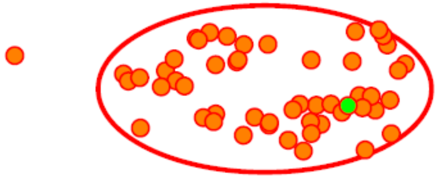


- Each user takes several steps of gradient descent.
- Centralized training: expectation of each gradient update over the data distribution is an unbiased estimate of the true gradient.
- Federated learning: each gradient update is an unbiased estimate of the gradient with respect to its local data.
- It is a biased estimate when it comes to the whole data (i.e., putting the data of all the clients together).

Federated Learning - Challenges

- Statistical heterogeneity of data over different clients
  - Different users generate different types of data.
  - Posing significant difficulty in formulating the goal in precise mathematical terms (Mohri et al., 2019).

- Robustness against adversarial attack
  - There is no mechanism to check the validity of the gradient updates.
  - ~~Data~~ Model poisoning attack ($w_{adv} = w + m \times \delta$).



- Ensuring fairness among users

- Reducing communication costs

Problem Formulation

Conventional FL objective (e.g. FedAvg, FedProx and etc.):

$$\min_{\mathbf{w}} f(\mathbf{w}) = \sum_{i=1}^{m} \lambda_i f_i(\mathbf{w})$$

averaging

where

$$f_i(\mathbf{w}) := \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\mathcal{L}\left(\mathbf{w}, \mathbf{x}_i, y_i\right) \right]$$

typical choice $\lambda_i = \frac{|\mathcal{D}_i|}{\sum_i |\mathcal{D}_i|}$ **match centralized training**

McMahan et al. **Communication-Efficient Learning of Deep Networks from Decentralized Data**. AISTATS 2017

Problem Formulation

Conventional FL objective (e.g. FedAvg, FedProx and etc.):

$$\min_{\mathbf{w}} f(\mathbf{w}) = \sum_{i=1}^{m} \lambda_i f_i(\mathbf{w})$$

averaging

where

$$f_i(\mathbf{w}) := \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\mathcal{L}\left(\mathbf{w}, \mathbf{x}_i, y_i\right) \right]$$

Rare data? Uncertain amount of data?

Fairness? Adversary?

McMahan et al. **Communication-Efficient Learning of Deep Networks from Decentralized Data**. AISTATS 2017

AFL objective

$$\min_{\mathbf{w}} \max_{i=1,\dots,m} f_i(\mathbf{w})$$

worst case

ensure more fairness

Mohri et al. **Agnostic federated learning**.
International Conference on Machine Learning, 2019.

q-FFL objective

$$\min_{\mathbf{w}} f_q(\mathbf{w}) := \sum_{i=1}^{m} \frac{\lambda_i}{q+1} f_i(\mathbf{w})^{q+1}$$

reweighting

fairness can be tuned

$q = 0$, **FedAvg**

$q = \infty$, **AFL**

Li et al., **Fair Resource Allocation in Federated Learning**. ICLR 2020

q-FFL objective

$$\min_{\mathbf{w}} f_q(\mathbf{w}) := \boxed{\sum_{i=1}^{m} \frac{\lambda_i}{q+1} f_i(\mathbf{w})^{q+1}}$$

reweighting

***Special case of Kolmogorov generalized mean**

$$\mathsf{A}_s(f) := s^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} s(f_i)\right)$$

Zhang et al., **Proportional Fairness in Federated Learning**. arxiv 2022

Inspiration

Goal: collectively optimize individual objective functions

$$f_1,\ f_2, \ldots,\ f_m$$

Inspiration

Goal: collectively optimize individual objective functions

$$\left(f_1, \; f_2, \ldots, \; f_m \right)$$

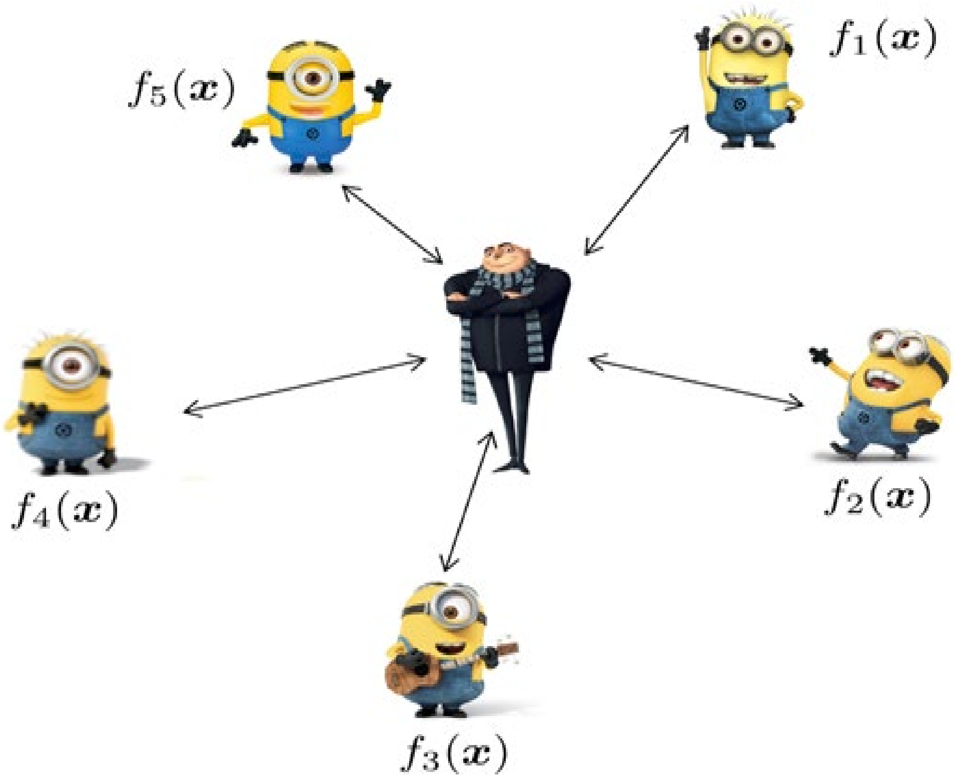
Multi-objective Formulation

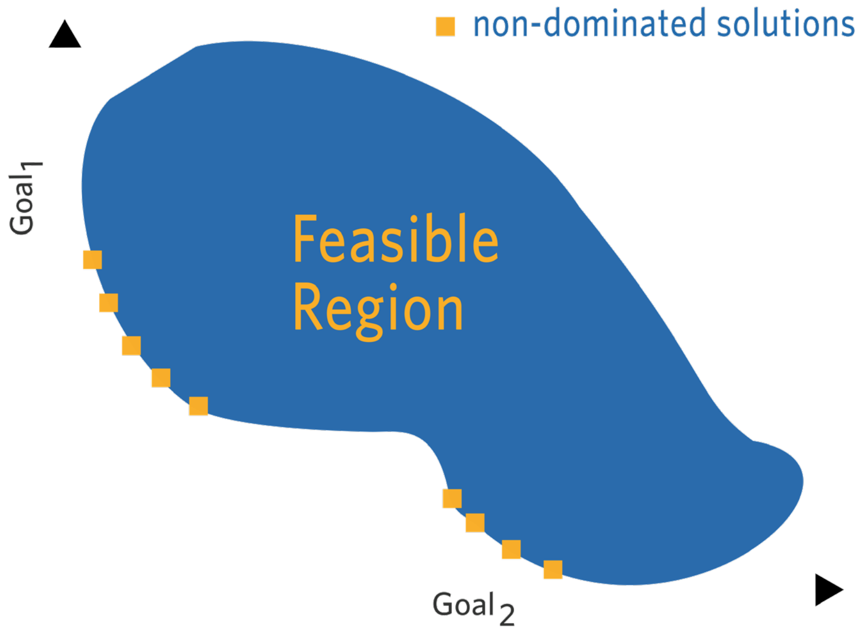
$$\min_{\mathbf{w}} \mathbf{f}(\mathbf{w}) := (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))$$

vector

What does this mean?

Multi-Objective Optimization (MOO)





https://users.ece.cmu.edu/~yuejiec/MachineLearning.html

Background: MOO

$$\min_{\mathbf{w}} \mathbf{f}(\mathbf{w}) := (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))$$

Minimum is defined wrt the partial ordering

$$\mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{z}) \iff \forall i,\ \ f_i(\mathbf{w}) \leq f_i(\mathbf{z})$$

Background: MOO

$$\min_{\mathbf{w}} \mathbf{f}(\mathbf{w}) := (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))$$

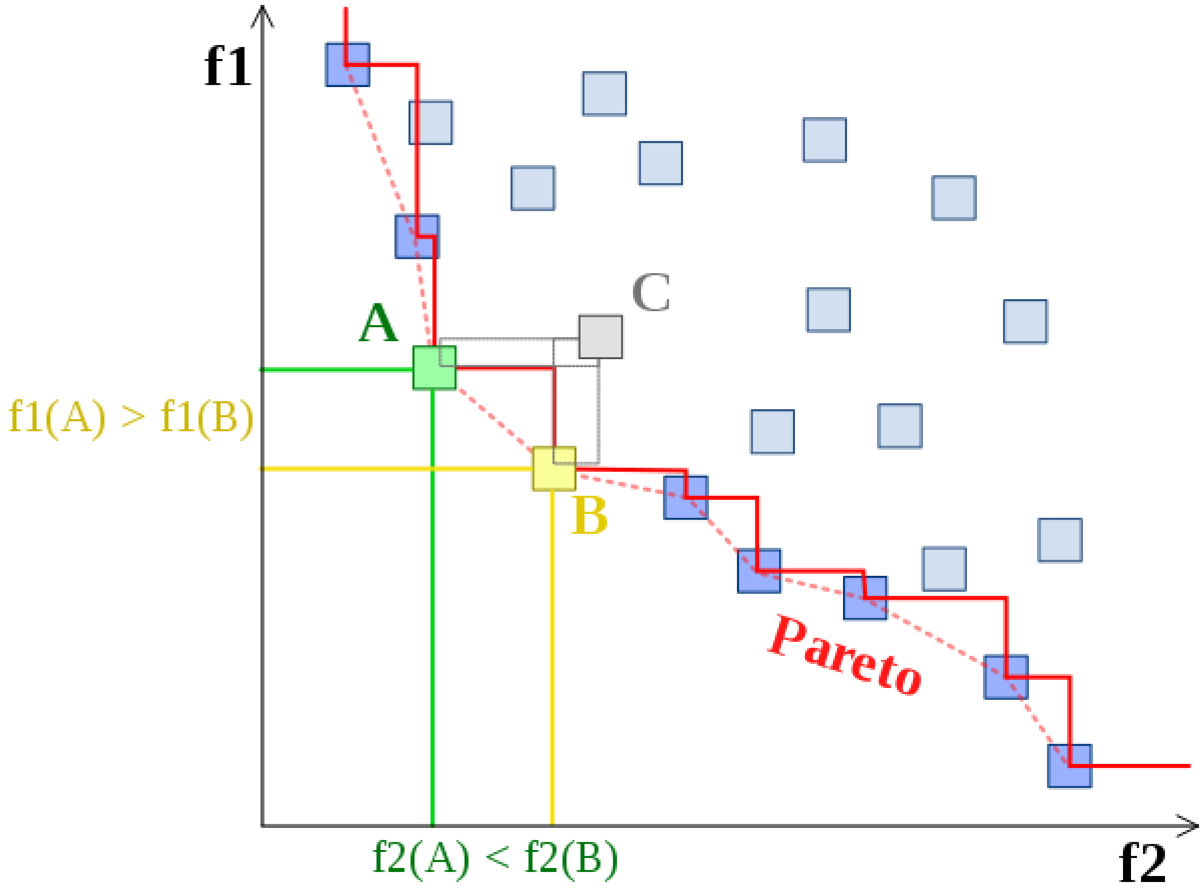
Minimum is defined wrt the partial ordering

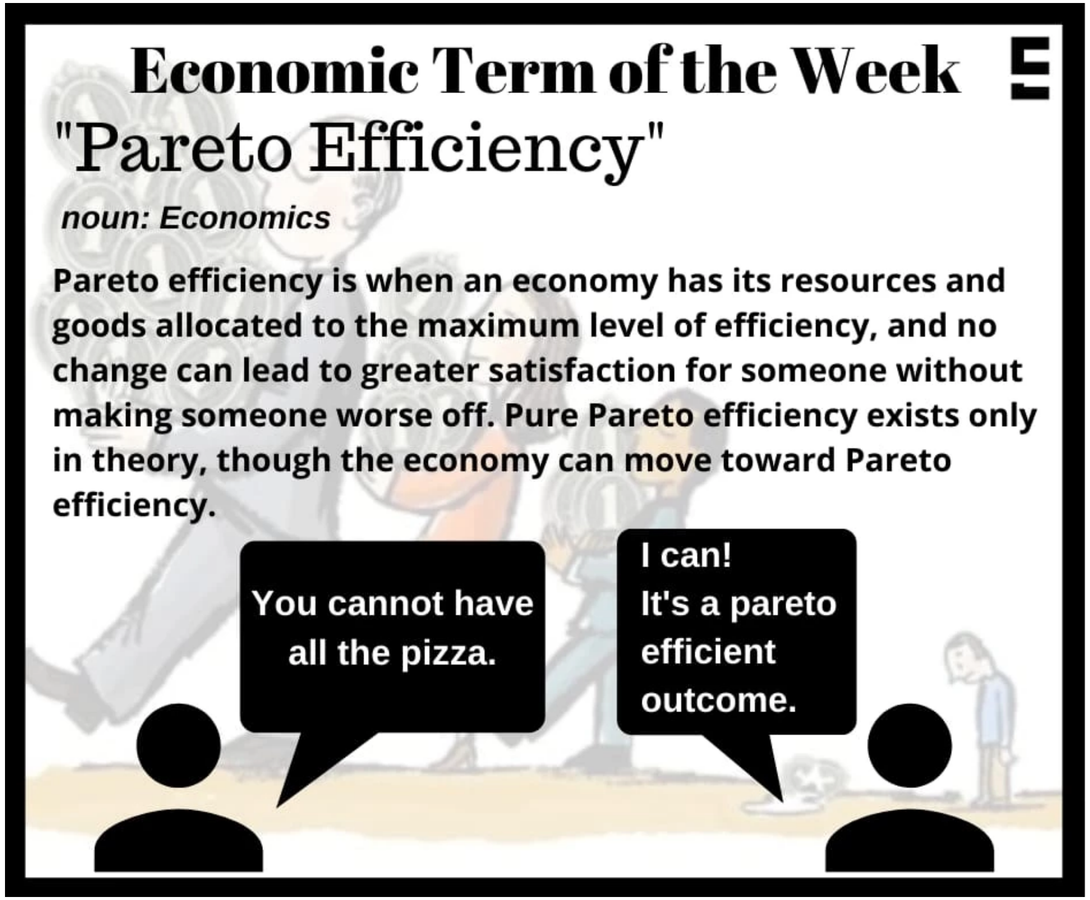
$$\mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{z}) \iff \forall i,\ f_i(\mathbf{w}) \leq f_i(\mathbf{z})$$

“dominates”

possible that w and z are not comparable

Pareto Optimality





$$\min_{\mathbf{w}} \mathbf{f}(\mathbf{w}) := (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))$$

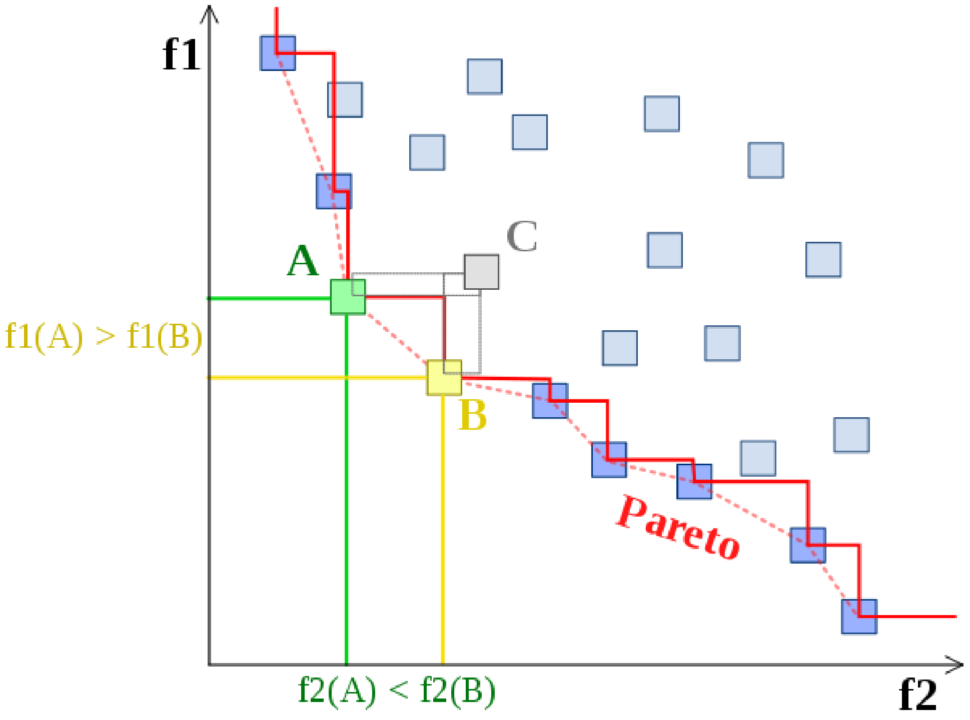
- $\mathbf{w}^*$ is a Pareto optimal solution if its objective value $\mathbf{f}(\mathbf{w}^*)$ is a minimum element wrt the partial ordering

Equivalently, $\forall \mathbf{w},\ \mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{w}^*) \Rightarrow \mathbf{f}(\mathbf{w}) = \mathbf{f}(\mathbf{w}^*)$

$$\min_{\mathbf{w}} \mathbf{f}(\mathbf{w}) := (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))$$

- $\mathbf{w}^*$ is a Pareto optimal solution if its objective value $\mathbf{f}(\mathbf{w}^*)$ is a minimum element wrt the partial ordering

Equivalently, $\forall \mathbf{w},\ \mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{w}^*) \Rightarrow \mathbf{f}(\mathbf{w}) = \mathbf{f}(\mathbf{w}^*)$



Pareto optimal:

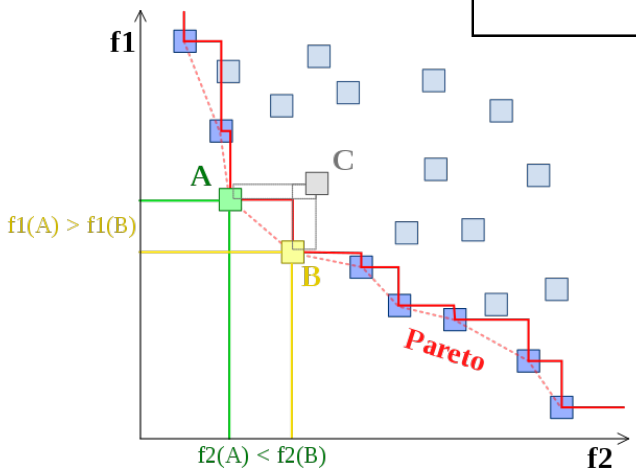
"You can be better than me in some aspects,
But you can't be better than me in all aspects"

$$\min_{\mathbf{w}} \mathbf{f}(\mathbf{w}) := (f_1(\mathbf{w}), f_2(\mathbf{w}), \ldots, f_m(\mathbf{w}))$$

- $\mathbf{w}^*$ is a Pareto optimal solution if its objective value $\mathbf{f}(\mathbf{w}^*)$ is a minimum element wrt the partial ordering

Equivalently, $\forall \mathbf{w},\ \mathbf{f}(\mathbf{w}) \leq \mathbf{f}(\mathbf{w}^*) \Rightarrow \mathbf{f}(\mathbf{w}) = \mathbf{f}(\mathbf{w}^*)$

not possible to <u>improve</u> any component objective without <u>compromising</u> some other objective



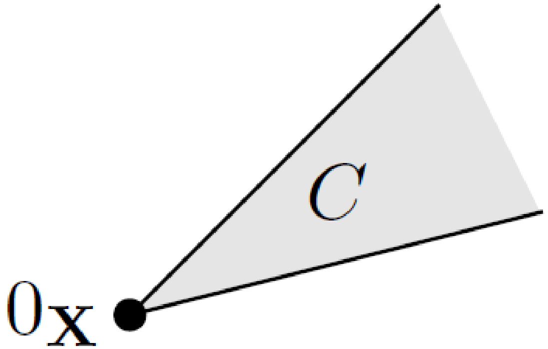
⬇

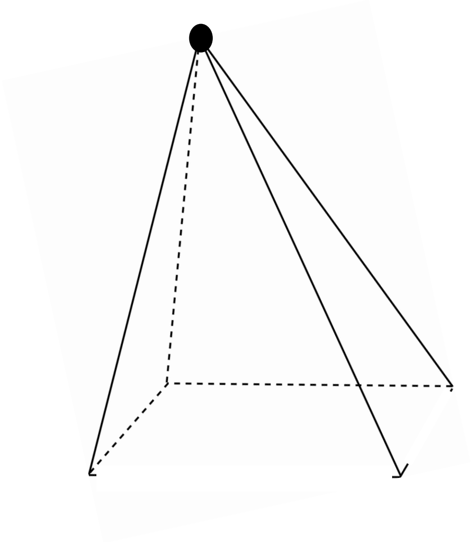
Fairness

Ordering Cone

characterization of partial ordering

Cones



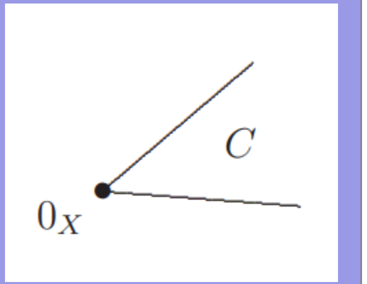


Ordering Cone

characterization of partial ordering

Theorem (Jahn, 2009)

Let $\mathbf{X}$ be a real linear space.



1. If $\leq$ is a partial ordering on $\mathbf{X}$, then the set

$$C := \{x \in \mathbf{X} | 0_{\mathbf{X}} \leq x\}$$

is a convex cone. If, in addition, $\leq$ is antisymmetric, the C is pointed.

2. If C is a convex cone in $\mathbf{X}$, then the binary relation

$$\leq_C := \{(x, y) \in \mathbf{X} \times \mathbf{X} | y - x \in C\}$$

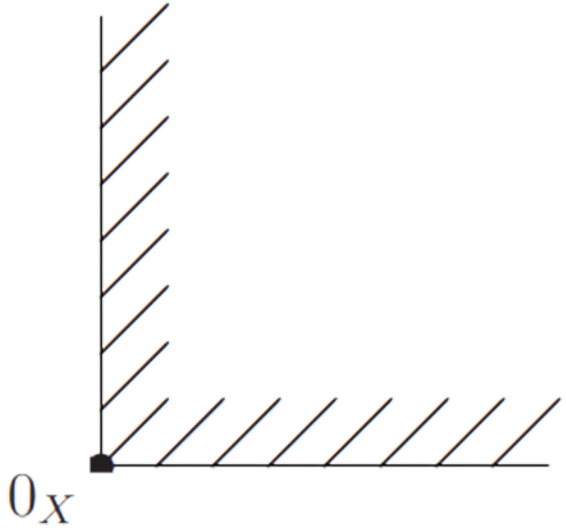
is a partial ordering on X. If, in addition, C is pointed, then $\leq_C$ is antisymmetric.

Cone that Induces MOO

Natural ordering cone (Jahn, 2009)

For $\mathbf{X} = \mathbb{R}^n$ the ordering cone of the component-wise partial ordering on $\mathbb{R}^n$ is given by

$$C := \{x \in \mathbb{R}^n | x_i \geq 0 \text{ for all } i \in \{1, \ldots, n\}\} = \mathbb{R}^n_+.$$

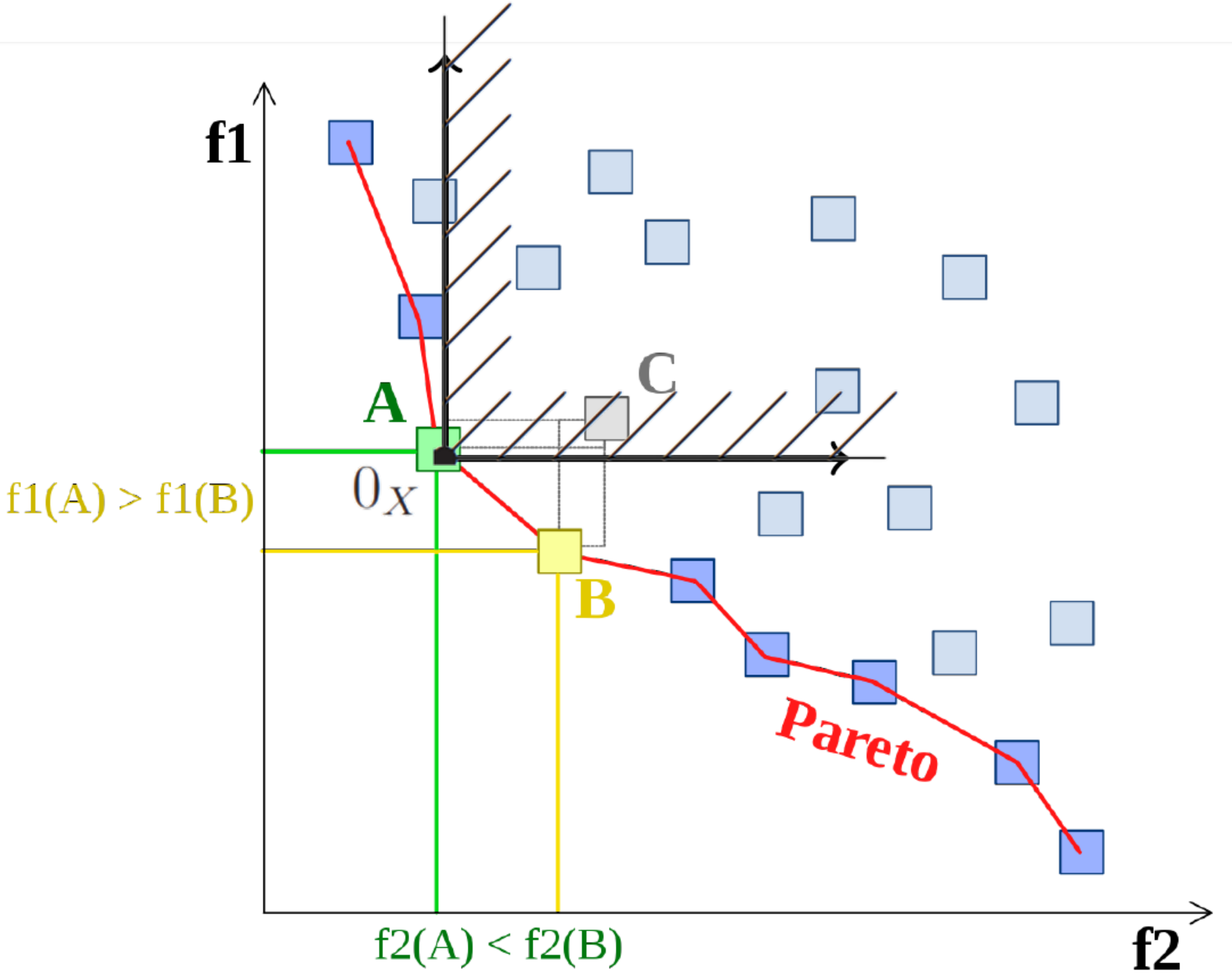


nonnegative orthant

Natural ordering cone (Jahn, 2009)

For $\mathbf{X} = \mathbb{R}^n$ the ordering cone of the component-wise partial ordering on $\mathbb{R}^n$ is given by

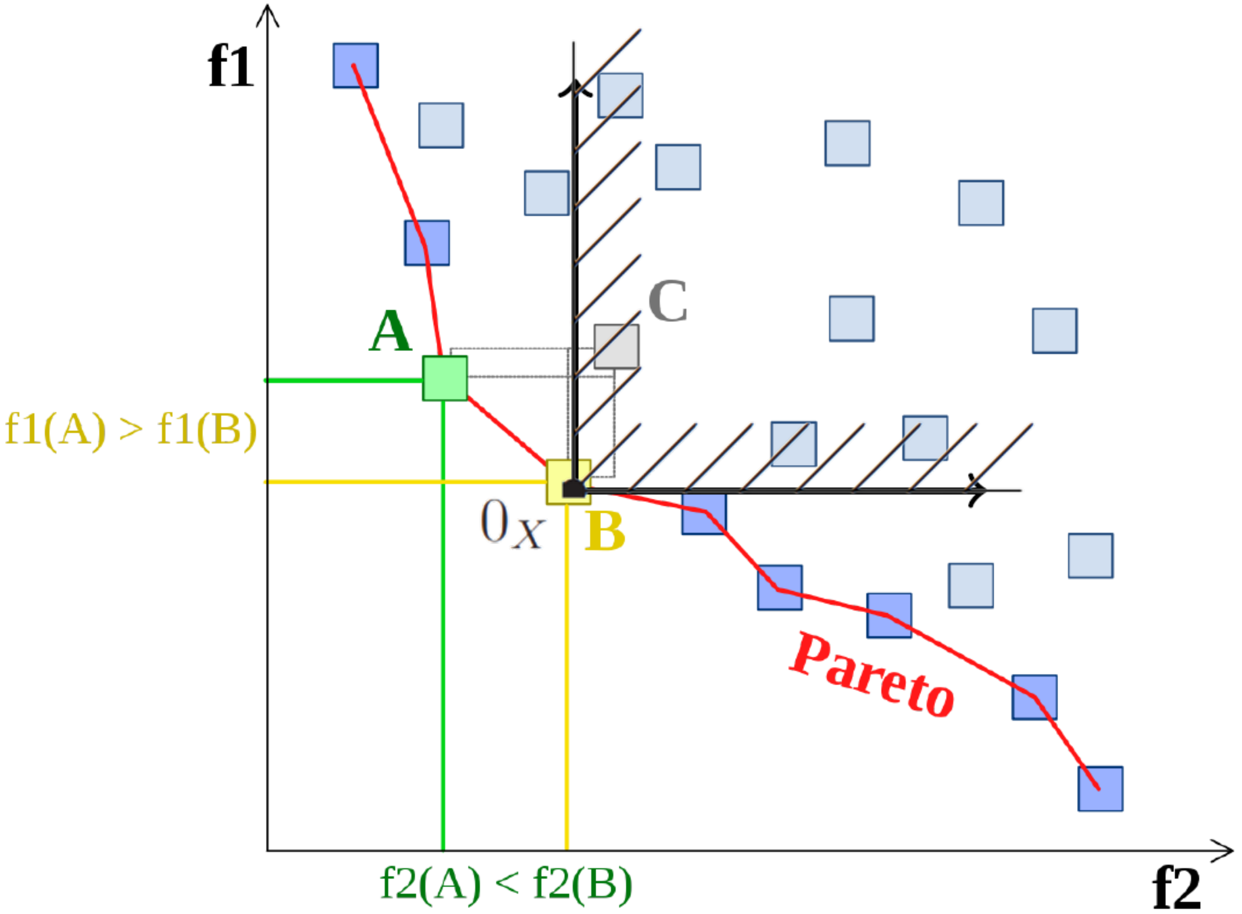
$$C := \{x \in \mathbb{R}^n | x_i \geq 0 \text{ for all } i \in \{1, \ldots, n\}\} = \mathbb{R}^n_+.$$



Natural ordering cone (Jahn, 2009)

For $\mathbf{X} = \mathbb{R}^n$ the ordering cone of the component-wise partial ordering on $\mathbb{R}^n$ is given by

$$C := \{x \in \mathbb{R}^n | x_i \geq 0 \text{ for all } i \in \{1, \ldots, n\}\} = \mathbb{R}^n_+.$$



Pareto Stationary

- All objective functions are continuously differentiable but not necessarily convex (to accommodate deep models).
- Finding a Pareto optimal solution in this setting is quite challenging.
- Instead, we will contend with Pareto stationary solutions, namely those that satisfy an intuitive first order necessary condition.

Definition: Pareto-stationary (Mukai, 1980)

We call x^* Pareto-stationary iff there exists some convex combination of the gradients $\{\nabla f_i(x^*)\}$ that equals zero.

Lemma (Mukai, 1980)

Any Pareto optimal solution is Pareto stationary. Conversely, if all functions are convex, then any Pareto stationary solution is weakly Pareto optimal.

"Pareto stationary vs. Pareto optimal is analogous to local vs. global optimal"

Solving MOO with scalarization

Weighted sum

$$\min_{\mathbf{w}} \sum_{i=1}^{m} \lambda_i f_i(\mathbf{w})$$

λ **fixed throughout**

Different weights leads to different Pareto stationary solutions

Lagrangian reformulation

Epsilon constraint

$$\min_{\mathbf{w}} f_\iota(\mathbf{w})$$
$$\text{s.t. } f_i(\mathbf{w}) \le \epsilon_i, \forall i \neq \iota$$

ϵ **fixed throughout**

Solving MOO with minimax

Chebyshev approach

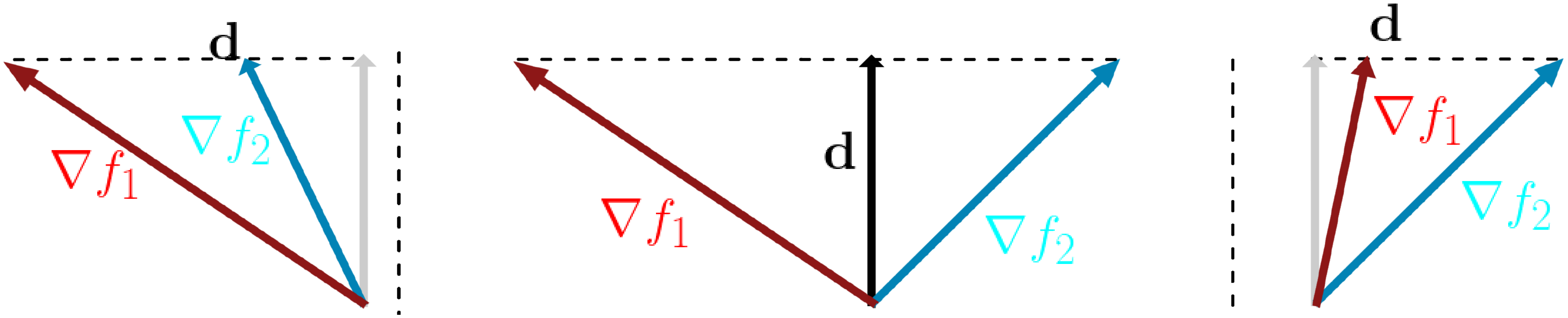
$$\min_{\mathbf{w}} \max_{\boldsymbol{\lambda} \in \Delta} \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{w}) - \mathbf{s})$$

$\mathbf{s}$ → fixed vector

s = 0 is essentially AFL

$$\min_{\mathbf{w}} \max_{\boldsymbol{\lambda} \in \Delta} \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{w}) \equiv \min_{\mathbf{w}} \max_{i=1,\ldots,m} f_i(\mathbf{w})$$

Multiple Gradient Descent Algorithm (MGDA)



finds the min-norm element d
in the convex hull spanned by gradients

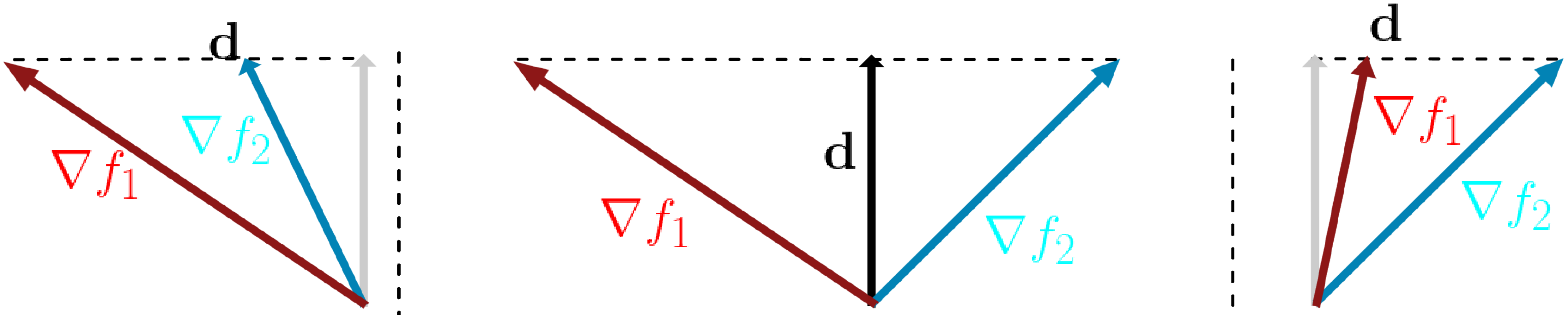
$$\mathbf{d} = \sum_i \lambda_i^* \nabla f_i(\mathbf{w})$$

$$\boldsymbol{\lambda}^* = \operatorname{argmin}_{\boldsymbol{\lambda}\in\Delta} \left\| \sum_i \lambda_i \nabla f_i(\mathbf{w}) \right\|^2$$

then descent along (negative) d

Jean-Antoine Désidéri. “Multiple-gradient descent algorithm (MGDA) for multiobjective optimization ” 2012

Multiple Gradient Descent Algorithm (MGDA)



finds the min-norm element d
in the convex hull spanned by gradients

-d is a descent direction that is common to all objectives

Jean-Antoine Désidéri. "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization " 2012

Primal-Dual interpretation of MGDA

Primal

$$\min_{\mathbf{d}} \max_{i=1,\dots,m} \langle \mathbf{d}, \nabla f_i(\mathbf{w}) \rangle + \frac{1}{2}\|\mathbf{d}\|^2$$

reformulation

$$\min_{\mathbf{d}} \alpha + \frac{1}{2}\|\mathbf{d}\|^2 \qquad \text{s.t.} \ \ \langle \mathbf{d}, \nabla f_i(\mathbf{w}) \rangle \leq \alpha, \ \forall i$$

Dual

$$\min_{\boldsymbol{\lambda} \in \Delta} \left\| \sum_i \lambda_i \nabla f_i(\mathbf{w}) \right\|^2$$

used in implementation

Jörg Fliege and Benar Fux Svaiter. "Steepest descent methods for multicriteria optimization" 2000

New Insights

Recall Chebyshev approach: $\min_{\mathbf{w}} \max_{\boldsymbol{\lambda}\in\Delta} \boldsymbol{\lambda}^\top(\mathbf{f}(\mathbf{w}) - \mathbf{s})$

Don't fix s?

adaptive 'centering'

$$\tilde{\mathbf{w}}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \max_{\boldsymbol{\lambda}\in\Delta} \boldsymbol{\lambda}^\top \left(\mathbf{f}(\mathbf{w}) - \mathbf{f}\left(\tilde{\mathbf{w}}_t\right)\right)$$

Apply quadratic bound

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \max_{\boldsymbol{\lambda}\in\Delta} \boldsymbol{\lambda}^\top J_{\mathbf{f}}^\top(\mathbf{w}_t)(\mathbf{w} - \mathbf{w}_t) + \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_t\|^2,$$

Swapping min and max

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{d}_t, \quad \mathbf{d}_t = J_{\mathbf{f}}(\mathbf{w}_t)\boldsymbol{\lambda}_t^*,$$
$$\text{where} \quad \boldsymbol{\lambda}_t^* = \operatorname*{argmin}_{\boldsymbol{\lambda}\in\Delta} \|J_{\mathbf{f}}(\mathbf{w}_t)\boldsymbol{\lambda}\|^2.$$

MGDA

Connections

$$\min_{\mathbf{w}} \max_{\boldsymbol{\lambda} \in \Delta} \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{w}) - \mathbf{s})$$

Chebyshev approach

$$s = \begin{cases} 0 & \textbf{AFL} \\ \mathbf{f}(\tilde{\mathbf{w}}_t) & \textbf{MGDA} \end{cases}$$

Connections

$$\min_{\mathbf{w}} \max_{\boldsymbol{\lambda}\in\Delta} \boldsymbol{\lambda}^\top(\mathbf{f}(\mathbf{w}) - \mathbf{s})$$

Chebyshev approach

s = 0	**AFL**	✘
s = $\mathbf{f}(\tilde{\mathbf{w}}_t)$	**MGDA**	✓

invariance to additive perturbation

FedMGDA+

Federated Learning Meets Multi-Objective Optimization

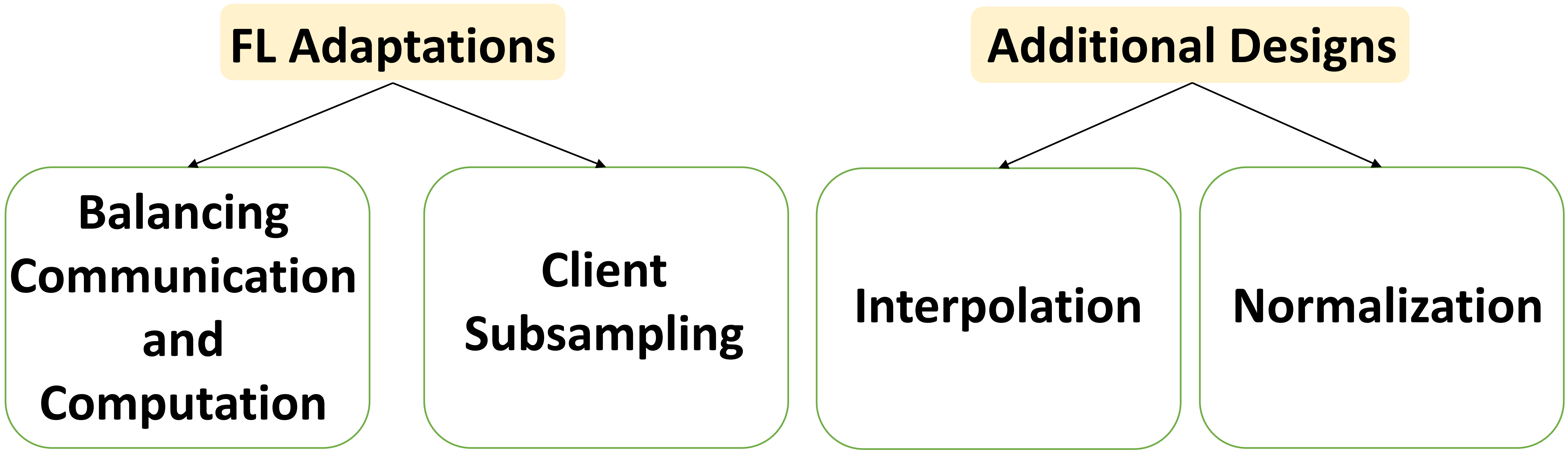
Incentive

Fairness

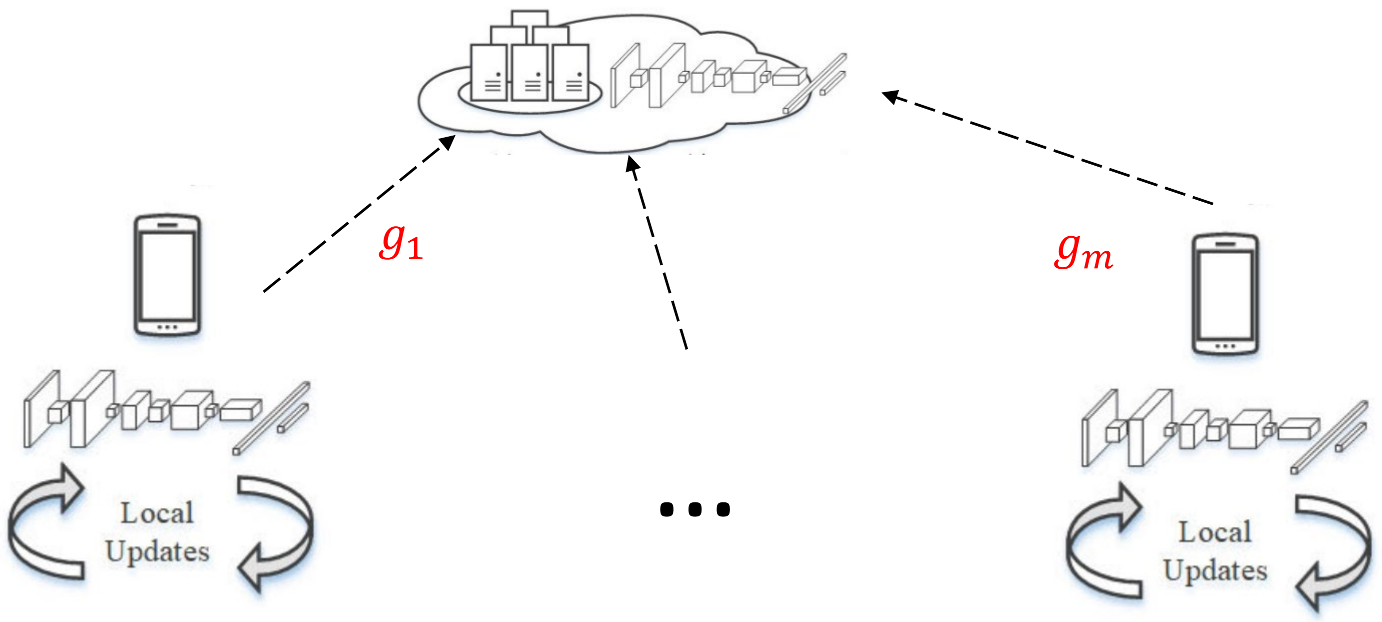
Robustness

Hu et al., **Federated learning meets multi-objective optimization**. IEEE Transactions on Network Science and Engineering, 2022

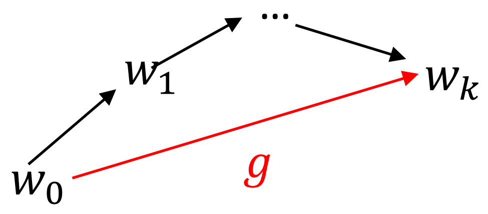
From MGDA to FedMGDA+



Balancing communication and on-device computation



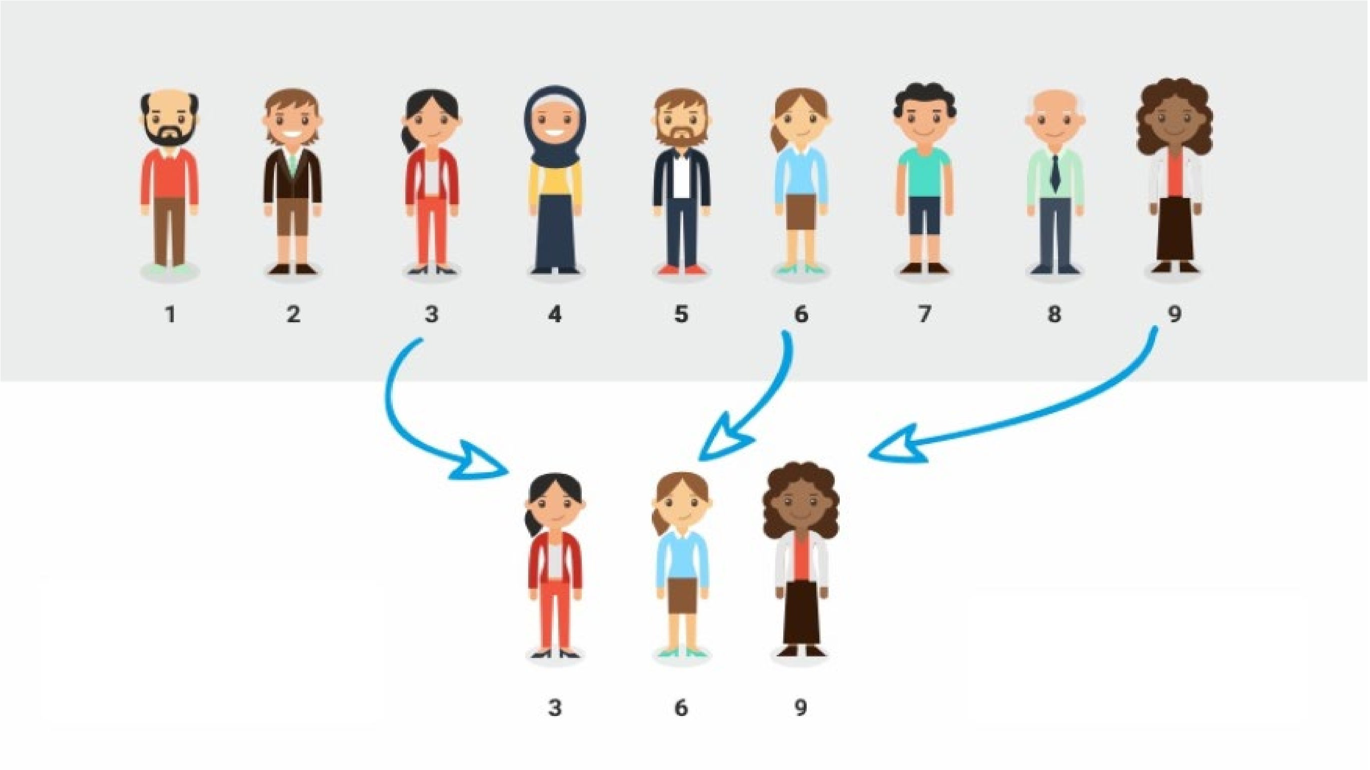
Allow multiple local updates before communicating



More local updates,
Less global communications.

Client Subsampling

- Common practice in FL
- Alleviate non-iid
- Enhance throughput



MGDA provides incentive for users to participate!

Interpolation

$$\boldsymbol{\lambda}_t^* = \operatorname*{argmin}_{\boldsymbol{\lambda}\in\Delta, \|\boldsymbol{\lambda}-\boldsymbol{\lambda}_0\|_\infty \le \epsilon} \|J_{\mathbf{f}}(\mathbf{w}_t)\boldsymbol{\lambda}\|^2.$$

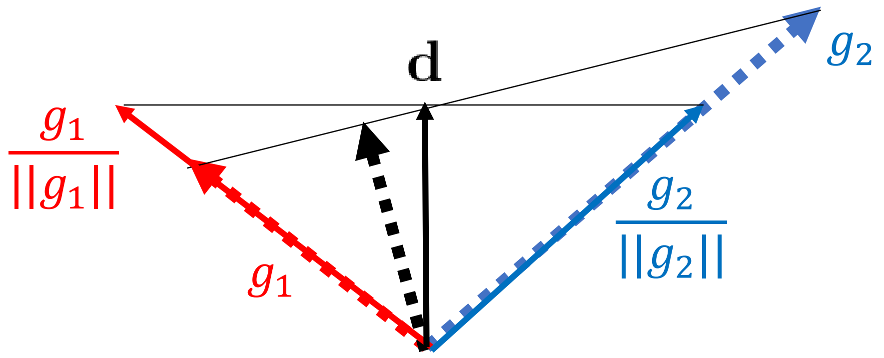
Balancing average performance and fairness

$\epsilon = 0$, FedAvg → **average performance**

$\epsilon = 1$, FedMGDA → **fairness**

Normalization

- Normalizing the (sub)gradient can sometimes ease step size tuning[i]
- Normalization does not change the 'common descent' property of MGDA
- Robustness against multiplicative inflation attack



i. Kurt Anstreicher et al., **Two "well-known" properties of subgradient optimization**. Mathematical Programming 2009

Algorithm: FedMGDA+

Algorithm 1: FedMGDA+

for $t = 1, 2, \ldots$ **do**
- choose a subset I_t of $\lceil pm \rceil$ clients/users
- **for** $i \in I_t$ **do**
  - $\mathbf{g}_i \leftarrow \textsc{ClientUpdate}(i, \mathbf{w}_t)$
  - $\bar{\mathbf{g}}_i := \mathbf{g}_i / \|\mathbf{g}_i\|$ // normalize
- $\boldsymbol{\lambda}^* \leftarrow \operatorname{argmin}_{\boldsymbol{\lambda} \in \Delta, \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_\infty \le \epsilon} \| \sum_i \lambda_i \bar{\mathbf{g}}_i \|^2$
- $\mathbf{d}_t \leftarrow \sum_i \lambda_i^* \bar{\mathbf{g}}_i$ // common direction
- choose (global) step size η_t
- $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \mathbf{d}_t$

Function $\textsc{ClientUpdate}(i, \mathbf{w})$:
- $\mathbf{w}^0 \leftarrow \mathbf{w}$
- **repeat** k epochs
  - // split local data into r batches
  - $\mathcal{D}_i \to \mathcal{D}_{i,1} \cup \cdots \cup \mathcal{D}_{i,r}$
  - **for** $j \in \{1, \ldots, r\}$ **do**
    - $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla f_i(\mathbf{w}; \mathcal{D}_{i,j})$
- **return** $\mathbf{g} := \mathbf{w}^0 - \mathbf{w}$ to server

Subsampling

Normalization

Interpolation

Multiple local epochs

Convergence Results

Theorem 1 (simplified)

Let each user function f_i be L-Lipschitz smooth and M-Lipschitz continuous, and choose step size η_t so that $\sum_t \eta_t = \infty$ and $\sum_t \sigma_t \eta_t < \infty$, where σ_t^2 is the variance of (the stochastic) common direction $\mathbf{d}_t$ under random subsampling. Then, with $r = k = 1$, we have for $\boldsymbol{\lambda}_t = \operatorname{argmin}_{\boldsymbol{\lambda} \in \Delta} \|J_{\mathbf{f}}(\mathbf{w}_t)\boldsymbol{\lambda}\|$:

$$\min_{t=0,\ldots,T} \|J_{\mathbf{f}}(\mathbf{w}_t)\boldsymbol{\lambda}_t\|^2 \to 0.$$

Convergence rate depends on how quickly the variance diminishes, which in turn depends on subsampling and heterogeneity of user objective functions

Convergence Results

Theorem 2 (simplified)

Suppose each user function f_i is convex and M-Lipschitz continuous. Suppose at each round `FedMGDA` includes a strongly convex user function whose weight is bounded away from 0. Then, with the choice $\eta_t = \frac{2}{c(t+2)}$ and $r = k = 1$, we have

$$\|\mathbf{w}_t - \mathbf{w}_t^*\|^2 \leq \frac{4M^2}{c^2(t+3)},$$

and $\mathbf{w}_t - \mathbf{w}_t^* \to 0$ almost surely, where $\mathbf{w}_t^*$ is the nearest Pareto stationary solution to $\mathbf{w}_t$ and c is some constant.

Experiments

- In our work, we conducted experiments on CIFAR-10, FMNIST, EMNIST, Shakespeare and Adult datasets
- Mainly, figures on CIFAR-10 are showed here for illustration purpose

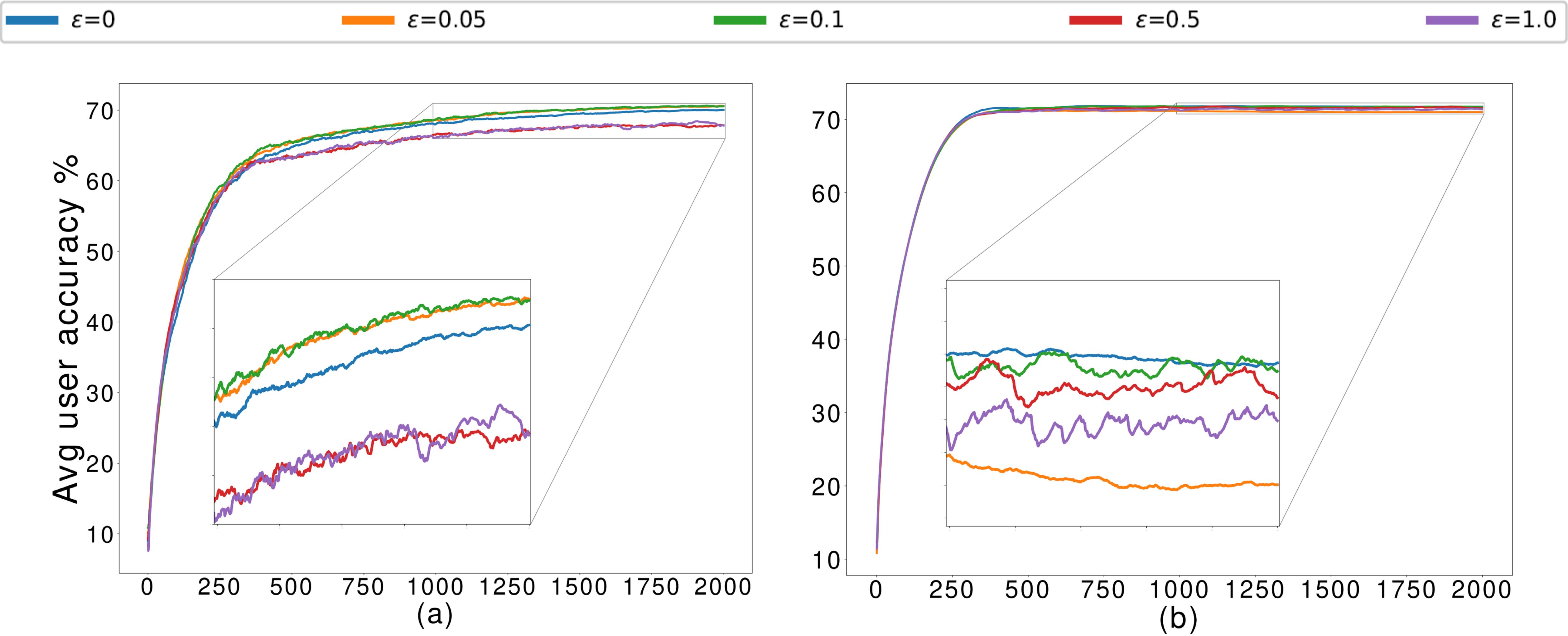
Algorithm: FedMGDA+

Algorithm 1: FedMGDA+

for $t = 1, 2, \ldots$ **do**
  choose a subset I_t of $\lceil pm \rceil$ clients/users → Subsampling
  for $i \in I_t$ **do**
    $\mathbf{g}_i \leftarrow \text{ClientUpdate}(i, \mathbf{w}_t)$
    $\bar{\mathbf{g}}_i := \mathbf{g}_i / \|\mathbf{g}_i\|$ // normalize → Normalization
  $\boldsymbol{\lambda}^* \leftarrow \operatorname{argmin}_{\boldsymbol{\lambda} \in \Delta, \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_\infty \leq \epsilon} \|\sum_i \lambda_i \bar{\mathbf{g}}_i\|^2$ → **Interpolation**
  $\mathbf{d}_t \leftarrow \sum_i \lambda_i^* \bar{\mathbf{g}}_i$ // common direction
  choose (global) step size η_t
  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \mathbf{d}_t$

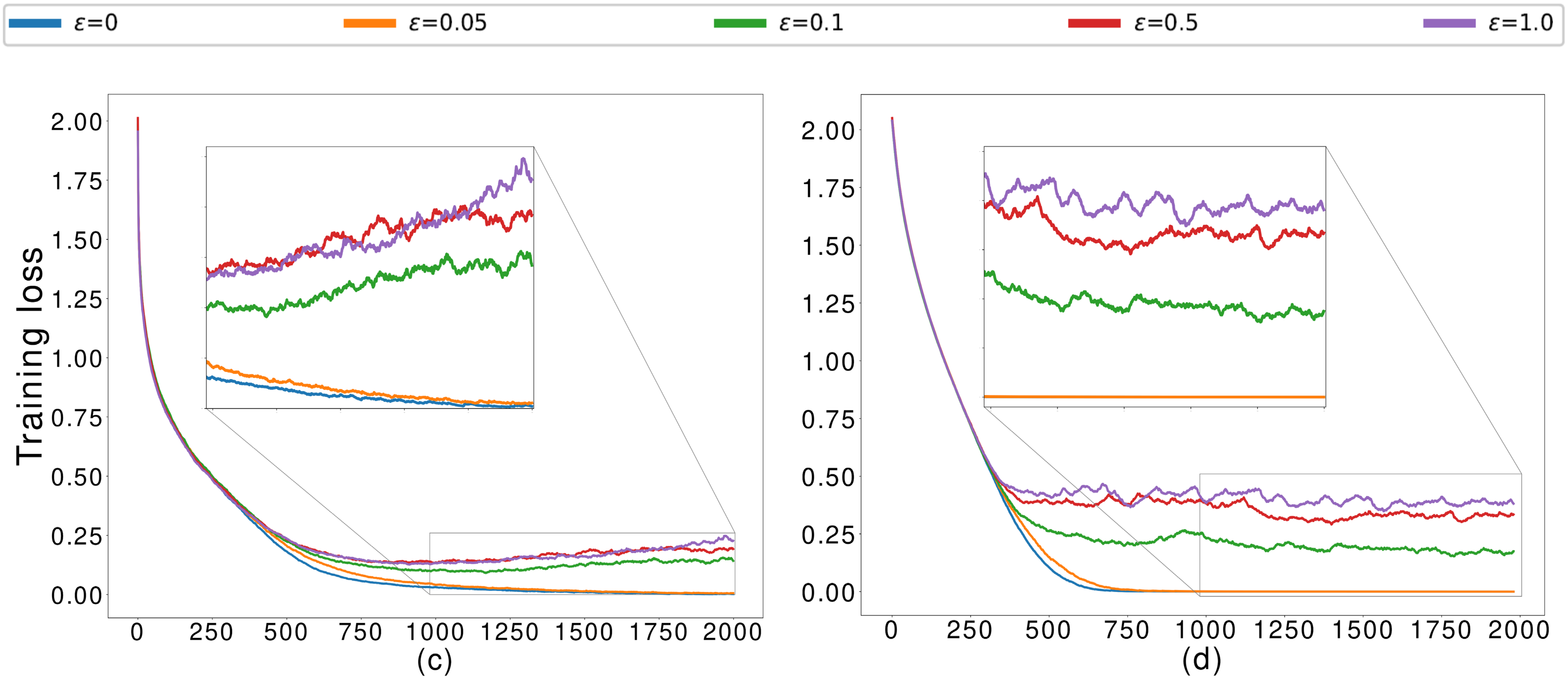
Function $\text{ClientUpdate}(i, \mathbf{w})$:
  $\mathbf{w}^0 \leftarrow \mathbf{w}$
  repeat k epochs → Multiple local epochs
    // split local data into r batches
    $\mathcal{D}_i \rightarrow \mathcal{D}_{i,1} \cup \cdots \cup \mathcal{D}_{i,r}$
    for $j \in \{1, \ldots, r\}$ **do**
      $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla f_i(\mathbf{w}; \mathcal{D}_{i,j})$
  return $\mathbf{g} := \mathbf{w}^0 - \mathbf{w}$ to server

Interpolation



CIFAR-10

Interpolation



CIFAR-10

Bias attack

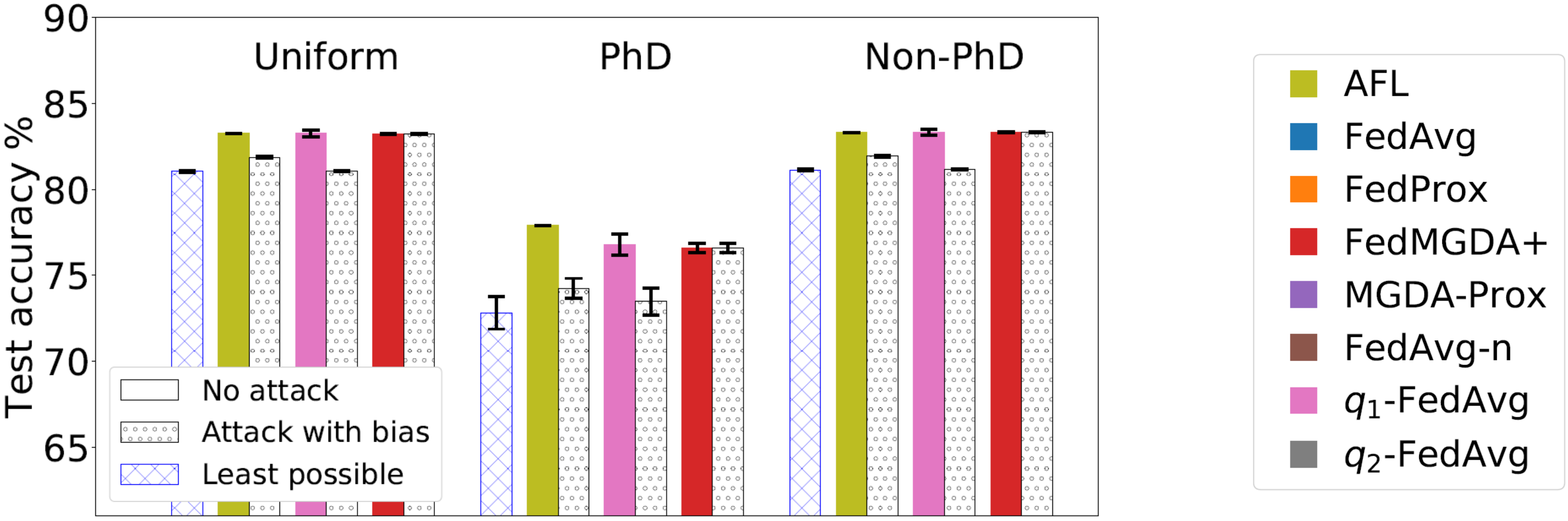
$$\min_{\mathbf{w}} \max_{\boldsymbol{\lambda} \in \Delta} \boldsymbol{\lambda}^\top (\mathbf{f}(\mathbf{w}) - \mathbf{s})$$

Chebyshev approach

$$s = \begin{cases} 0 & \text{AFL} \quad ✗ \\ \mathbf{f}(\tilde{\mathbf{w}}_t) & \text{MGDA} \quad ✓ \end{cases}$$

invariance to additive perturbation

Robustness



Bias attack on Adult dataset

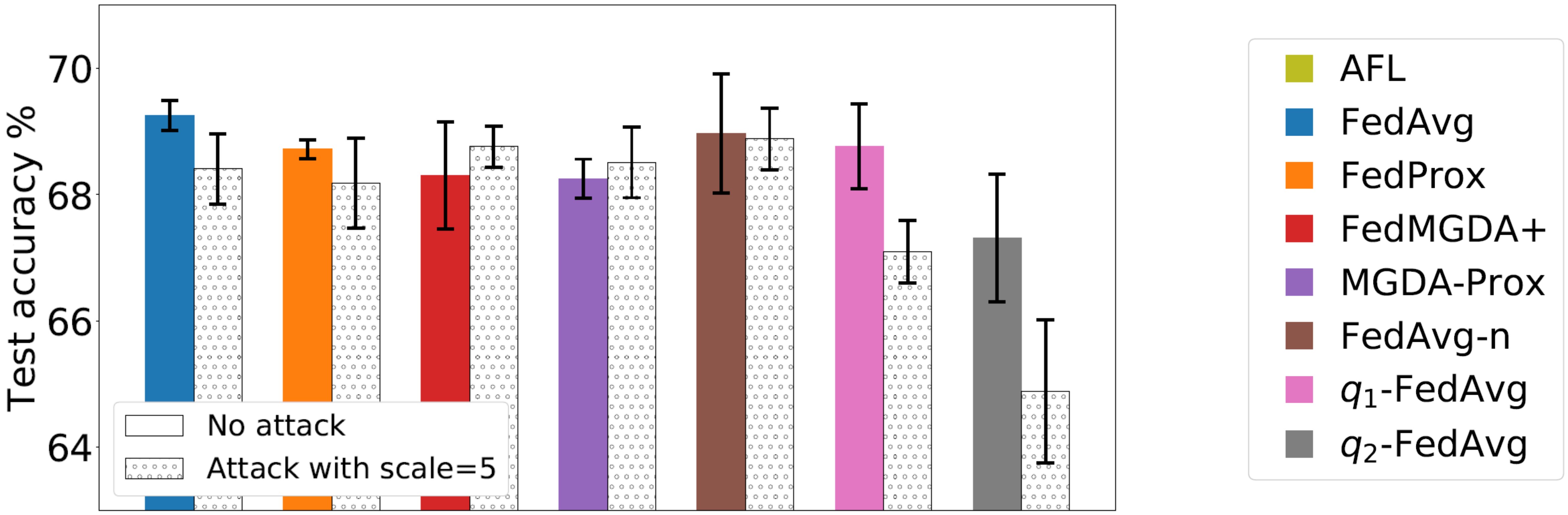
Algorithm: FedMGDA+

Algorithm 1: FedMGDA+

for $t = 1, 2, \ldots$ **do**
- choose a subset I_t of $\lceil pm \rceil$ clients/users → Subsampling
- **for** $i \in I_t$ **do**
  - $\mathbf{g}_i \leftarrow \text{ClientUpdate}(i, \mathbf{w}_t)$
  - $\bar{\mathbf{g}}_i := \mathbf{g}_i / \|\mathbf{g}_i\|$ // normalize → **Normalization (Robustness)**
- $\boldsymbol{\lambda}^* \leftarrow \operatorname{argmin}_{\boldsymbol{\lambda} \in \Delta, \|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_\infty \le \epsilon} \|\sum_i \lambda_i \bar{\mathbf{g}}_i\|^2$ → Interpolation
- $\mathbf{d}_t \leftarrow \sum_i \lambda_i^* \bar{\mathbf{g}}_i$ // common direction
- choose (global) step size η_t
- $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \mathbf{d}_t$

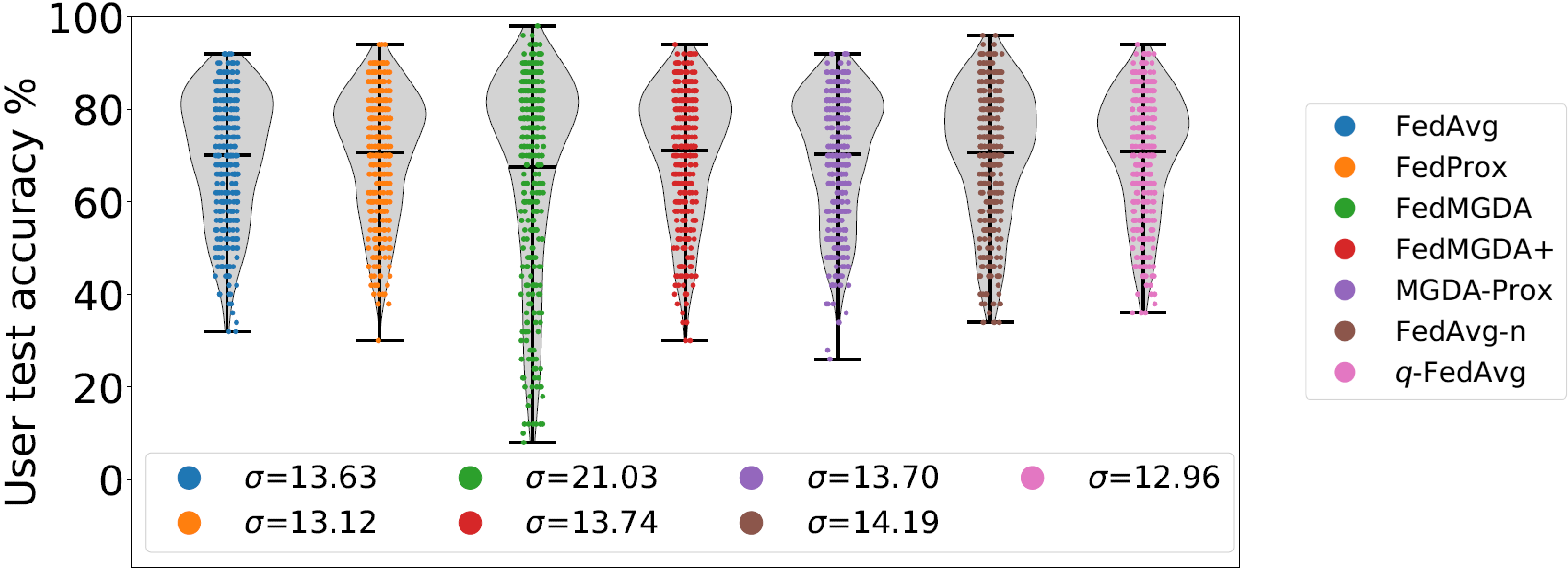
Function $\text{ClientUpdate}(i, \mathbf{w})$:
- $\mathbf{w}^0 \leftarrow \mathbf{w}$
- **repeat** k epochs → Multiple local epochs
  - // split local data into r batches
  - $\mathcal{D}_i \rightarrow \mathcal{D}_{i,1} \cup \cdots \cup \mathcal{D}_{i,r}$
  - **for** $j \in \{1, \ldots, r\}$ **do**
    - $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla f_i(\mathbf{w}; \mathcal{D}_{i,j})$
- **return** $\mathbf{g} := \mathbf{w}^0 - \mathbf{w}$ to server

Robustness



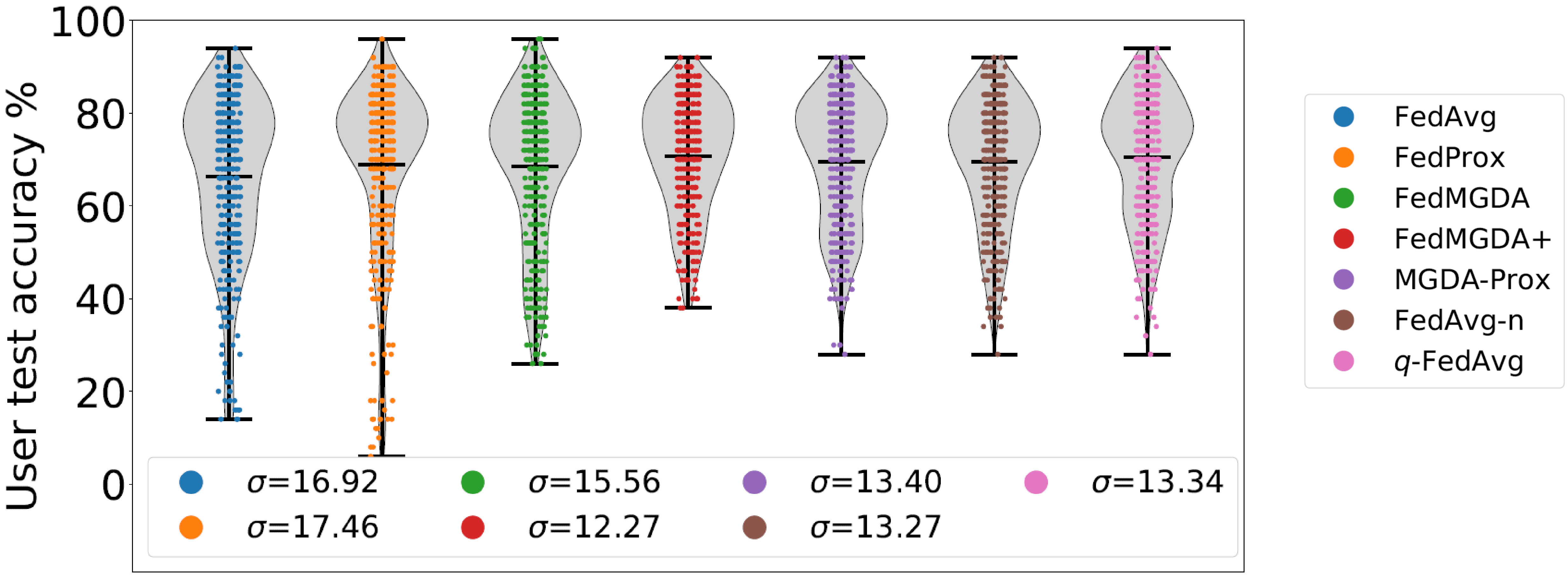
Scaling attack on CIFAR-10

Fairness



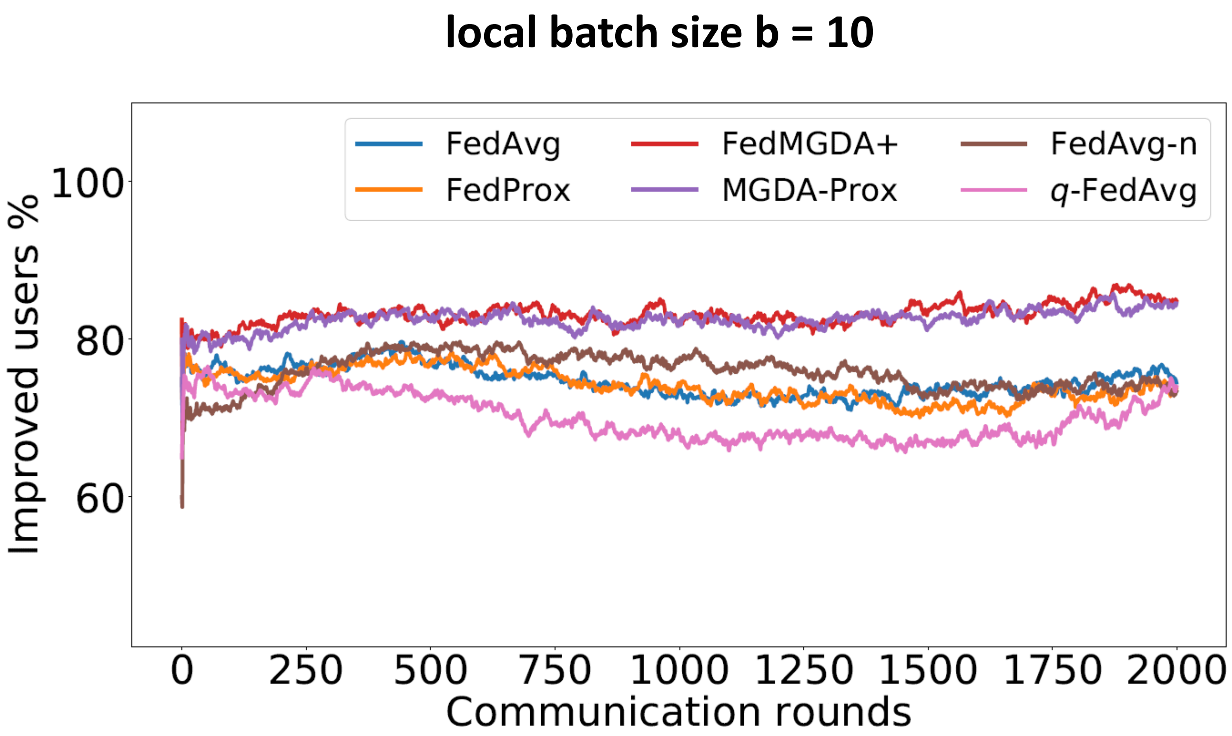
CIFAR-10, small local batch size

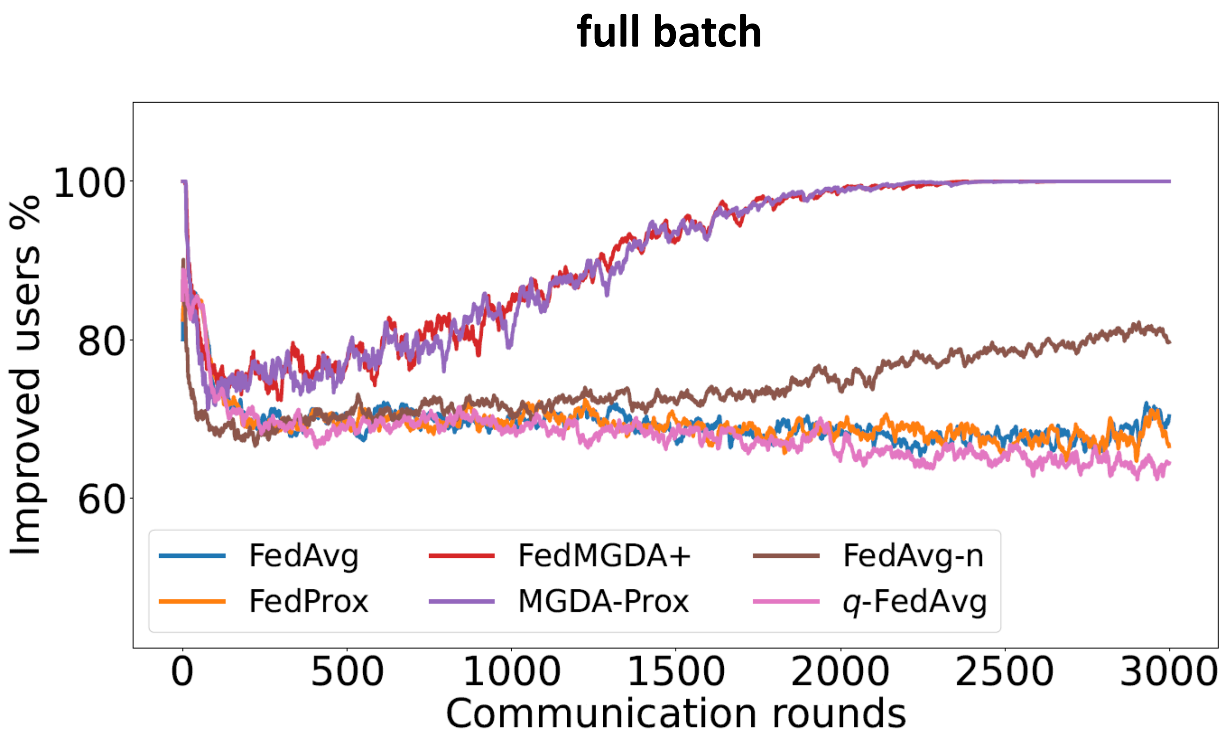
Fairness



CIFAR-10, big local batch size

Improvement Fairness





Percentage of improved participants each round

CIFAR-10

More Tables....

Check appendices of our paper

Algorithm	Average (%)	Std. (%)
FedMGDA	85.73 ± 0.05	14.79 ± 0.12
FedMGDA+	$\mathbf{87.60 \pm 0.20}$	$\mathbf{13.68 \pm 0.19}$
MGDA-Prox	87.59 ± 0.19	13.75 ± 0.18
FedAvg	84.97 ± 0.44	15.25 ± 0.36
FedAvg-n	87.57 ± 0.09	13.74 ± 0.11
FedProx	84.97 ± 0.45	15.26 ± 0.35
q-FedAvg	84.97 ± 0.44	15.25 ± 0.37

EMNIST

Algorithm			Average (%)	Std. (%)	Worst 5% (%)	Best 5% (%)
Name	η	decay				
FedMGDA			67.59 ± 0.65	21.03 ± 2.40	22.95 ± 7.27	$\mathbf{90.50 \pm 0.87}$
FedMGDA+	1.0	0	69.06 ± 1.08	14.10 ± 1.61	44.38 ± 5.90	87.55 ± 0.84
FedMGDA+	1.0	1/10	69.87 ± 0.87	14.33 ± 0.61	42.42 ± 3.61	87.05 ± 0.95
FedMGDA+	1.5	1/10	$\mathbf{71.15 \pm 0.62}$	13.74 ± 0.49	44.48 ± 1.64	88.53 ± 0.85
FedMGDA+	1.0	1/40	68.68 ± 1.25	17.23 ± 1.60	34.40 ± 6.23	88.07 ± 0.04
FedMGDA+	1.5	1/40	71.05 ± 0.82	13.53 ± 0.77	$\mathbf{46.50 \pm 2.96}$	88.53 ± 0.85
Name	η	decay				
MGDA-Prox	1.0	0	66.98 ± 1.52	15.46 ± 3.15	39.42 ± 10.35	87.60 ± 2.18
MGDA-Prox	1.0	1/10	70.39 ± 0.96	13.70 ± 1.08	46.43 ± 2.17	87.50 ± 0.87
MGDA-Prox	1.5	1/10	69.45 ± 0.77	14.98 ± 1.61	40.42 ± 5.88	87.05 ± 1.00
MGDA-Prox	1.0	1/40	69.01 ± 0.51	16.24 ± 0.74	36.92 ± 4.12	88.53 ± 0.85
MGDA-Prox	1.5	1/40	69.53 ± 0.70	15.90 ± 1.79	36.43 ± 7.42	87.53 ± 2.14
Name	η	decay				
FedAvg			70.11 ± 1.27	13.63 ± 0.81	45.45 ± 2.21	88.00 ± 0.00
FedAvg-n	1.0	0	67.69 ± 1.15	16.97 ± 2.33	37.98 ± 6.61	89.55 ± 2.61
FedAvg-n	1.0	1/10	69.66 ± 1.22	15.11 ± 1.14	40.42 ± 1.71	88.55 ± 0.84
FedAvg-n	1.5	1/10	70.62 ± 0.82	14.19 ± 0.49	43.48 ± 2.17	89.03 ± 1.03
FedAvg-n	1.0	1/40	70.31 ± 0.29	14.97 ± 0.96	42.48 ± 2.56	88.55 ± 2.15
FedAvg-n	1.5	1/40	70.47 ± 0.70	13.88 ± 0.96	44.95 ± 4.07	88.03 ± 0.04
Name	μ					
FedProx	0.01		70.77 ± 0.70	13.12 ± 0.47	46.43 ± 2.95	88.50 ± 0.87
FedProx	0.1		70.69 ± 0.58	13.42 ± 0.43	45.42 ± 2.14	87.55 ± 1.64
FedProx	0.5		68.89 ± 0.83	14.10 ± 1.08	43.95 ± 4.52	88.00 ± 0.00
Name	q	L				
q-FedAvg	0.1	0.1	70.40 ± 0.41	$\mathbf{12.43 \pm 0.24}$	46.48 ± 2.14	87.50 ± 0.87
q-FedAvg	0.5	0.1	70.58 ± 0.73	13.60 ± 0.47	$\mathbf{46.50 \pm 2.96}$	88.05 ± 1.38
q-FedAvg	1.0	0.1	70.27 ± 0.61	13.31 ± 0.46	45.95 ± 1.38	87.55 ± 0.90
q-FedAvg	0.1	1.0	70.95 ± 0.83	12.70 ± 0.74	46.45 ± 4.07	87.00 ± 1.00
q-FedAvg	0.5	1.0	70.98 ± 0.52	12.96 ± 0.63	45.95 ± 1.45	88.00 ± 0.00
q-FedAvg	1.0	1.0	69.98 ± 0.67	13.15 ± 1.12	45.95 ± 2.49	87.53 ± 0.82

Table 8: Test accuracy of users on CIFAR-10 with local batch size $b = 10$, fraction of users $p = 0.1$, local learning rate $\eta = 0.01$, total communication rounds 2000. The reported statistics are averaged across 4 runs with different random seeds.

Hu et al., **Federated learning meets multi-objective optimization**. IEEE Transactions on Network Science and Engineering, 2022

Thank you for your attention!