15-884: Machine Learning Systems

Federated Learning

Tian Li

tianli@cmu.edu

Carnegie Mellon University
School of Computer Science

Federated Learning

Privacy-preserving <u>training</u> in heterogeneous, (potentially) massive networks

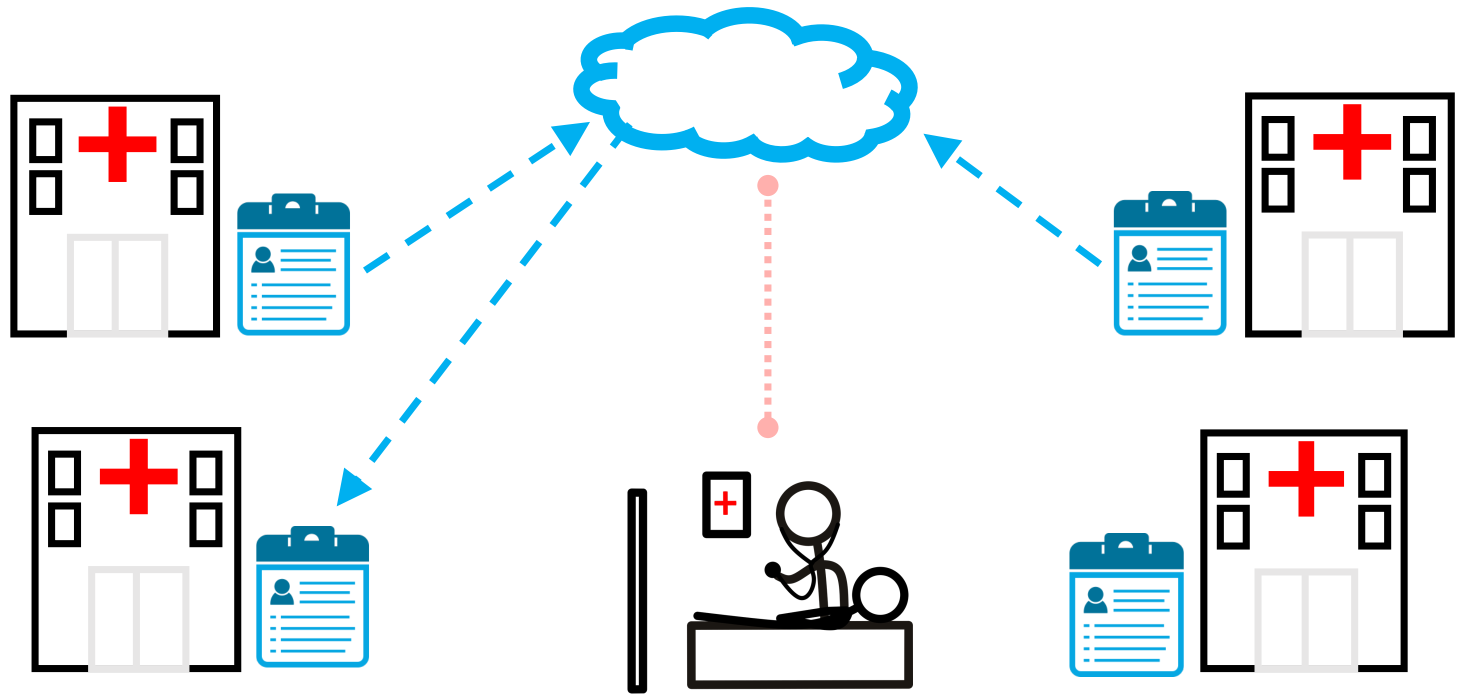
Federated Learning

Privacy-preserving training in heterogeneous, (potentially) massive networks

Networks of remote devices

cross-device setting

Networks of isolated organizations



cross-silo setting

Example Applications

- Anomaly detection in IoT devices
- Adapting to pedestrian behavior on autonomous vehicles
- Personalized healthcare on wearable devices
- Predictive maintenance for industrial machines

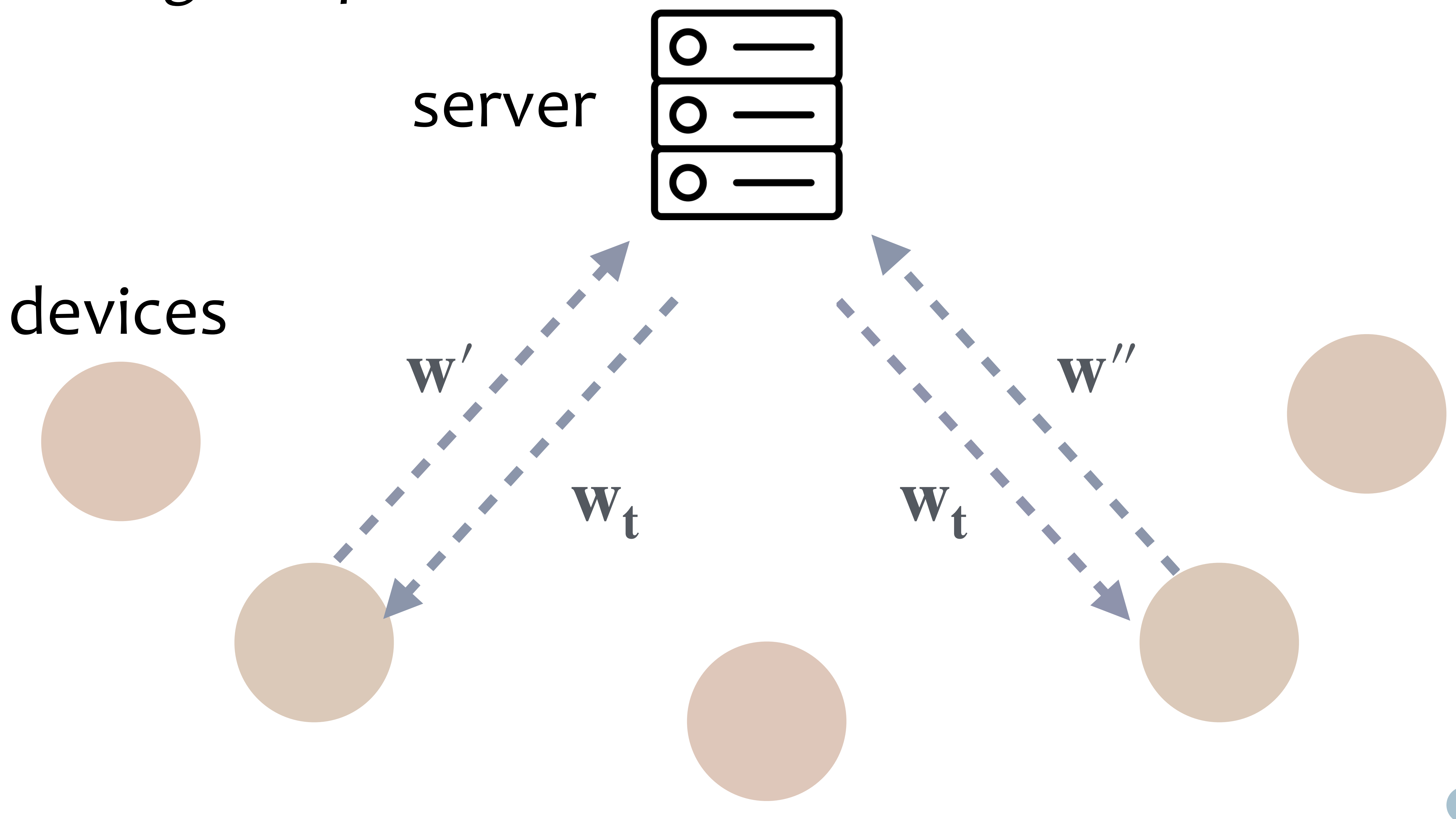
Assumptions: (1) local data is important (2) labels are available (3) privacy is a concern

Workflow & Challenges

Objective: $$\min_{w} f(w) = \sum_{k=1}^{N} p_k F_k(w)$$

loss on device k

Training setup:



Systems heterogeneity
variable hardware, network connectivity, power, etc

Statistical heterogeneity
highly non-identically distributed data

Expensive communication
massive, slow networks

Privacy & security
user privacy constraints

Federated Optimization: Challenges

Systems and statistical heterogeneity (non-identical data) can bias the optimization procedure;
can affect the modeling approach

Systems heterogeneity
variable hardware, network connectivity, power, etc

Statistical heterogeneity
highly non-identically distributed data

Expensive communication
massive, slow networks

Privacy & security
user privacy constraints

Federated Optimization: Challenges

1) reduce the size of messages per round
2) reduce the communication rounds
3) reduce the number of selected devices per round

Systems heterogeneity
variable hardware, network connectivity, power, etc

Statistical heterogeneity
highly non-identically distributed data

Expensive communication
massive, slow networks

Privacy & security
user privacy constraints

Federated Optimization: Challenges

1) keep data on local devices
2) differentially private mechanisms
3) crypto-based methods

(not the focus today)

Systems heterogeneity
variable hardware, network connectivity, power, etc

Statistical heterogeneity
highly non-identically distributed data

Expensive communication
massive, slow networks

Privacy & security
user privacy constraints

How does heterogeneity affect federated **optimization methods**?

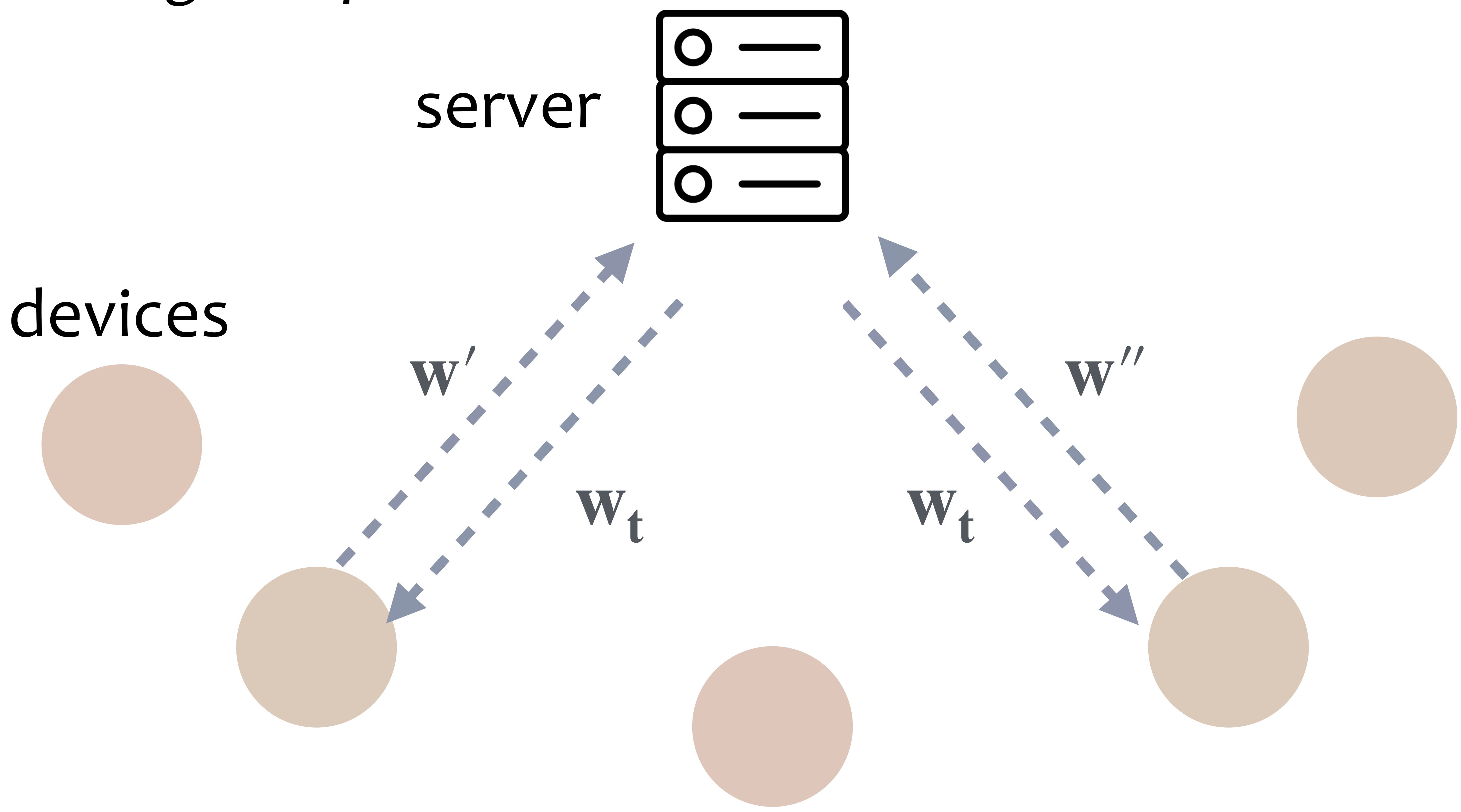
Federated Optimization in Heterogeneous Networks
Li, Sahu, Sanjabi, Zaheer, Talwalkar, Smith, MLSys 2020

Federated Optimization: Formulation

Objective: $$\min_{w} f(w) = \sum_{k=1}^{N} p_k F_k(w)$$

loss on device k

Training setup:



Typically solving an empirical risk minimization (ERM) objective:

$$\min_{w} \sum_{k=1}^{N} p_k \sum_{i=1}^{n_k} \ell(h(x_k^{(i)}; w), y_k^{(i)})$$

Federated Optimization: Formulation

Risk:

$$R(h) = \mathbb{E}_{k \sim Q} \mathbb{E}_{(x,y) \sim P_k} [\ell(h(x; w), y)]$$

Empirical Risk:

$$R_{\text{emp}}(h) = \sum_{k=1}^{N} p_k \sum_{i=1}^{n_k} \ell(h(x_k^{(i)}; w), y_k^{(i)})$$

Typically solving an empirical risk minimization (ERM) objective:

$$\min_{w} \sum_{k=1}^{N} p_k \sum_{i=1}^{n_k} \ell(h(x_k^{(i)}; w), y_k^{(i)})$$

Optimization for FL: Federated Averaging (FedAvg*)

At each communication round:

- Server randomly selects a subset of devices & sends the current global model w^t
- Each selected device k updates w^t for E epochs of SGD to optimize F_k & sends the new local model back
- Server aggregates local models to form a new global model w^{t+1}

- Simple method
- Using local updates can lead to much faster convergence empirically
- Works well in many settings (especially non-convex)

* McMahan, H. Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." AISTATS, 2017.

[Aside] How does FedAvg Differ from Distributed SGD?

Local updating is not new*

- one-shot averaging
- ADMM
- COCOA
- Local SGD

Federated settings defer in terms of:

- heterogeneous data
- partial device participation
- often for non-convex objectives

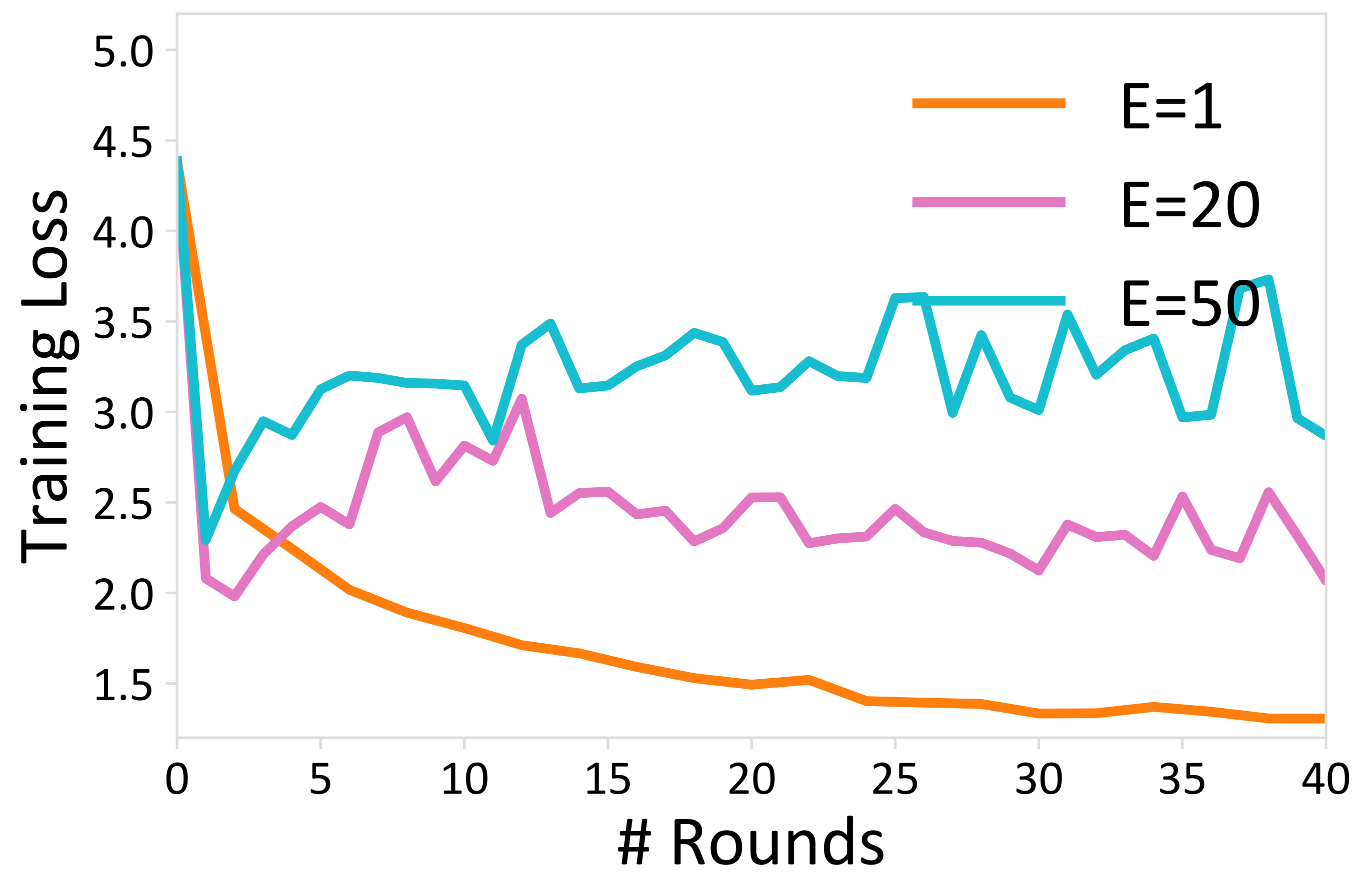
* [Zhang, Duchi, Wainwright, Communication-Efficient Algorithms for Statistical Optimization, JMLR 2013]
* [Boyd et al, Distributed Optimization and Statistical Learning via ADMM, FnT in ML, 2010]
* [Jaggi & Smith et al, Communication-Efficient Distributed Dual Coordinate Ascent, NeurIPS 2014]
* [MacDonald et al, Efficient large-scale distributed training of conditional maxent models, NeurIPS 2009]

Challenge: Heterogeneity

statistical heterogeneity

highly non-identically distributed data

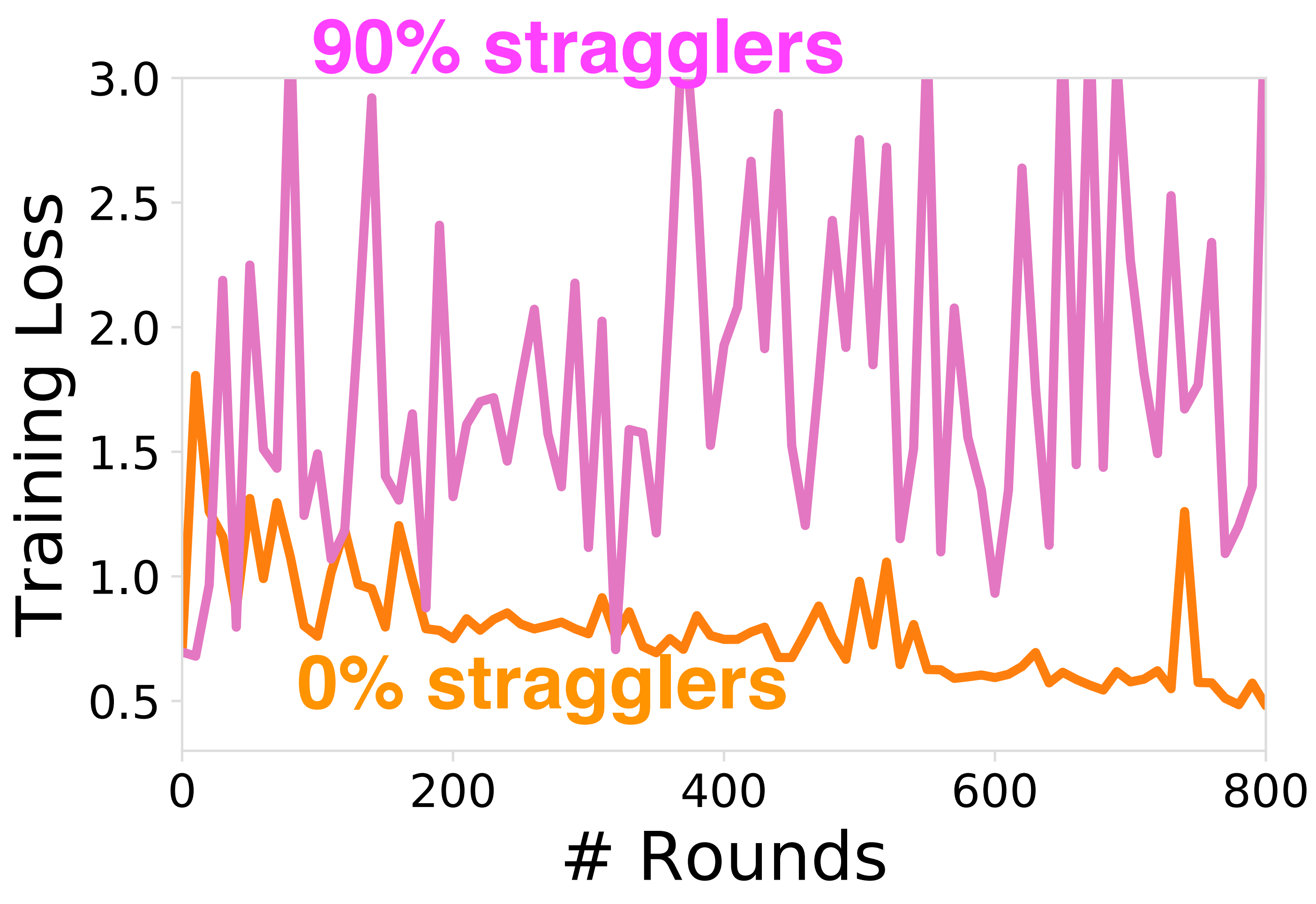
too much local work can hurt convergence



systems heterogeneity

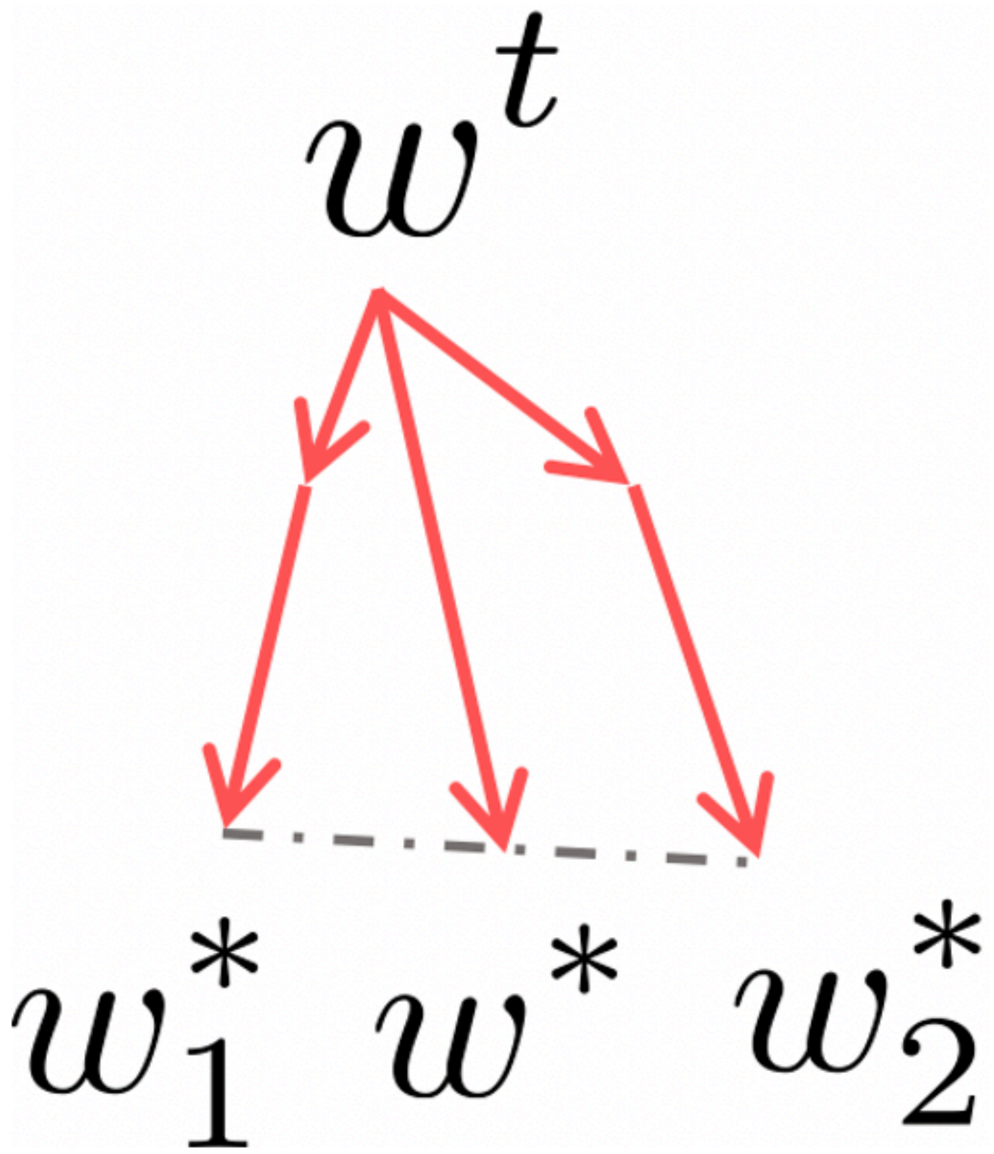
stragglers

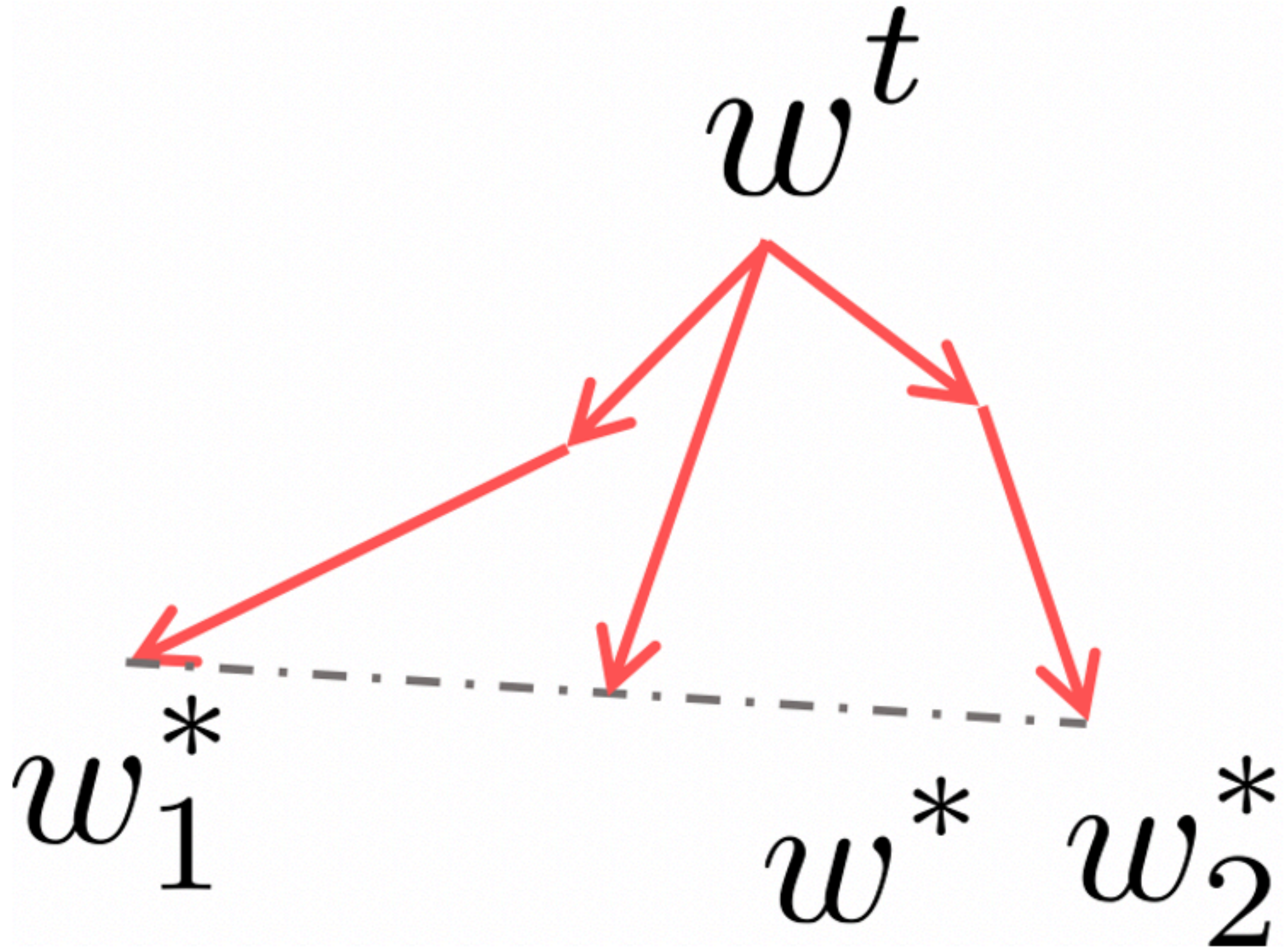
dropping slow devices can exacerbate convergence issues



Bonawitz, Keith, et al. "Towards Federated Learning at Scale: System Design." MLSys, 2019.

Challenge: Heterogeneity

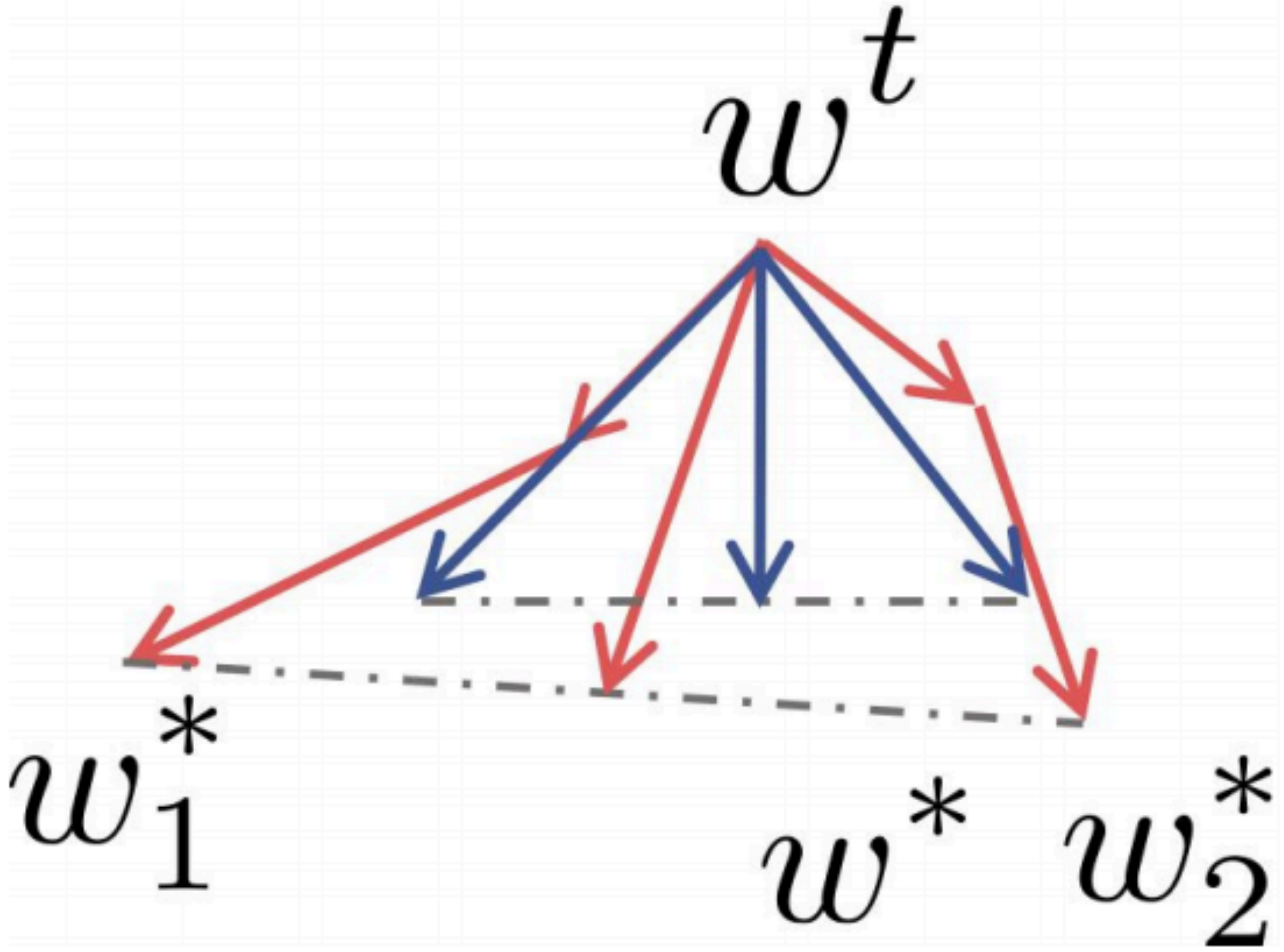




FedProx: A Framework For Federated Optimization

Modified Local Subproblem: $$\min_{w_k} F_k(w_k) + \underbrace{\frac{\mu}{2} \| w_k - w^t \|^2}_{\text{a proximal term}}$$

- The proximal term explicitly limits the impact of heterogeneous local updates
- Don't drop devices: instead [safely] incorporate partial work
- Generalization of FedAvg; Allows for any local solver
- Theoretical guarantees (with a dissimilarity assumption)



FedProx: Convergence Analysis

- High-level: **converges** despite non-IID data, local updating, and partial device participation
- Introduces notion of **B-dissimilarity** in to characterize statistical heterogeneity:

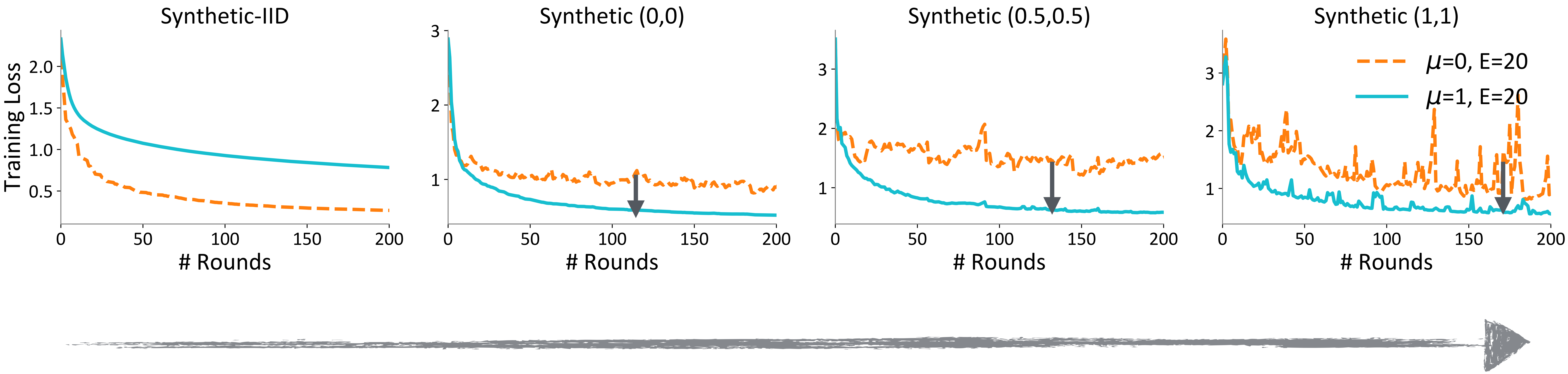
$$\mathbb{E}\left[\|\nabla F_k(w)\|^2\right] \leq \|\nabla f(w)\|^2 B^2$$

IID data: $B = 1$
non-IID data: $B > 1$

** used in other contexts, e.g., gradient diversity to quantify the benefits of scaling distributed SGD*

Yin, Dong, et al. "Gradient Diversity: a Key Ingredient for Scalable Distributed Learning." AISTATS, 2018.

Impact of Statistical Heterogeneity



Increasing heterogeneity leads to worse convergence

Setting μ > 0 can help to combat this

How does heterogeneity affect federated optimization methods?

- Heterogeneity can lead to:
  - Slower convergence, reduced stability, divergence
- Critical to analyze and evaluate federated methods with:
  - Non-IID data, partial / variable participation

Can we equalize performance across heterogeneous networks?

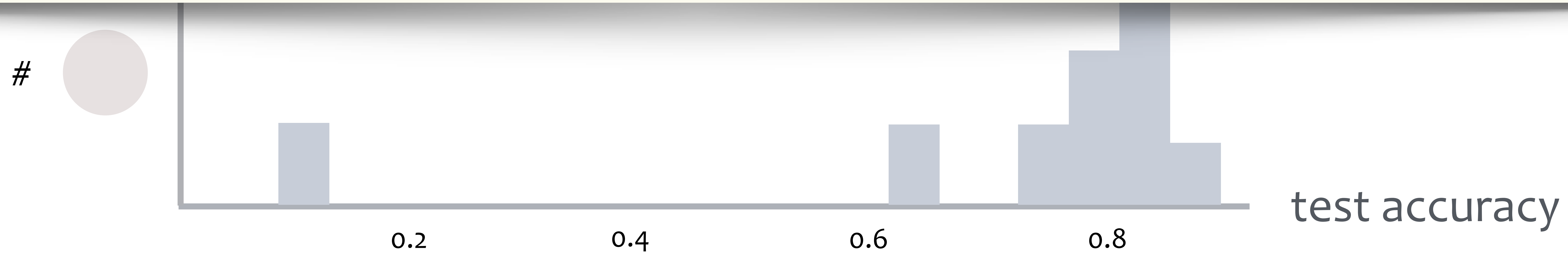
Fair Resource Allocation in Federated Learning
Li, Sanjabi, Beirami, Smith, ICLR 2020

FL: Traditional Empirical Risk Minimization

ERM: $$\min_{w} \left(p_1 F_1 + p_2 F_2 + \cdots + p_N F_N \right)$$

no accuracy guarantees for individual devices

Can we encourage a more **fair** (i.e., more **uniform**) distribution of the model performance across devices?



Fair Resource Allocation Objective

q-FFL: $$\min_{w} \frac{1}{q+1}\left(p_1 F_1^{q+1} + p_2 F_2^{q+1} + \cdots + p_N F_N^{q+1}\right)$$

- Inspired by α-fairness for fair resource allocation in wireless networks
- A tunable **framework** ($q = 0$: previous objective; $q = \infty$: minimax fairness*)

**Fairness without Demographics in Repeated Loss Minimization*, Hashimoto et al, ICML 2018
**Agnostic Federated Learning*, Mohri, Sivek, Suresh, ICML 2019

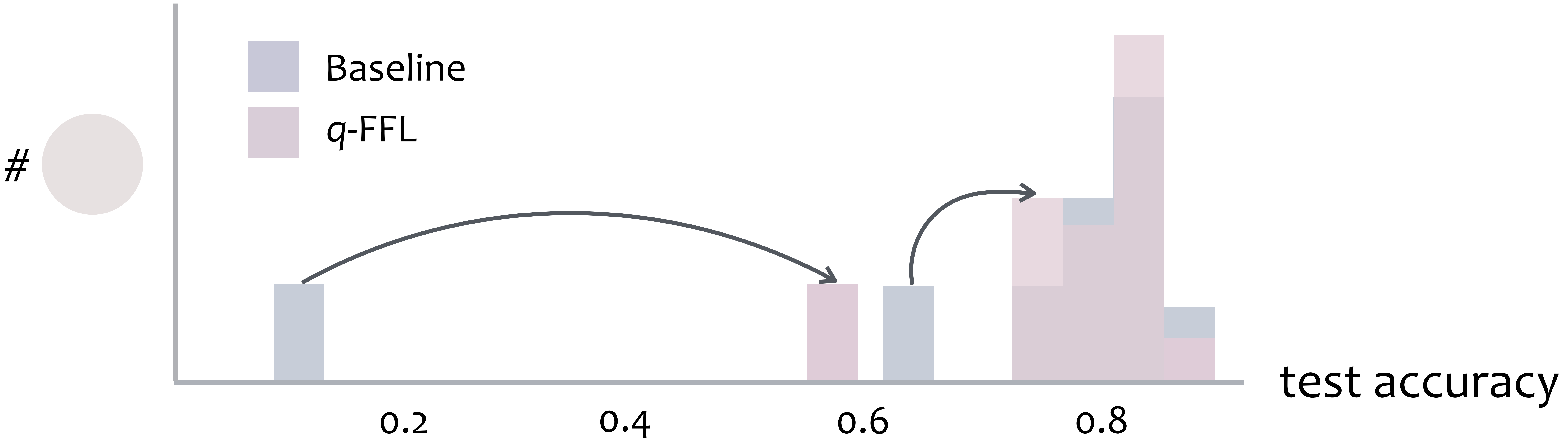
Fair Resource Allocation Objective

q-FFL: $$\min_{w} \frac{1}{q+1}\left(p_1 F_1^{q+1} + p_2 F_2^{q+1} + \cdots + p_N F_N^{q+1}\right)$$

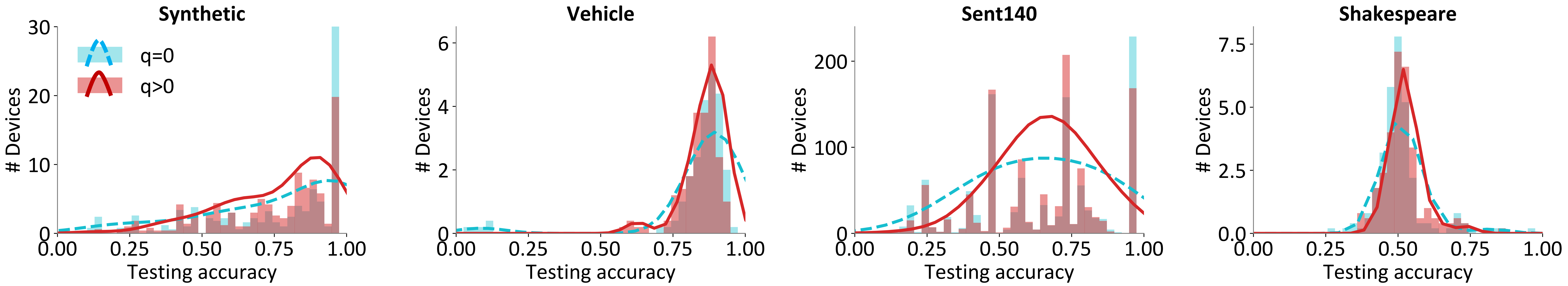
- Theory
  - Generalization guarantees (recover the known case of $q \to \infty$)
  - Increasing q results in more 'uniform' accuracy distributions (in terms of various uniformity measures such as variance)

Fair Resource Allocation Objective

q-FFL: $$\min_{w} \frac{1}{q+1}\left(p_1 F_1^{q+1} + p_2 F_2^{q+1} + \cdots + p_N F_N^{q+1}\right)$$

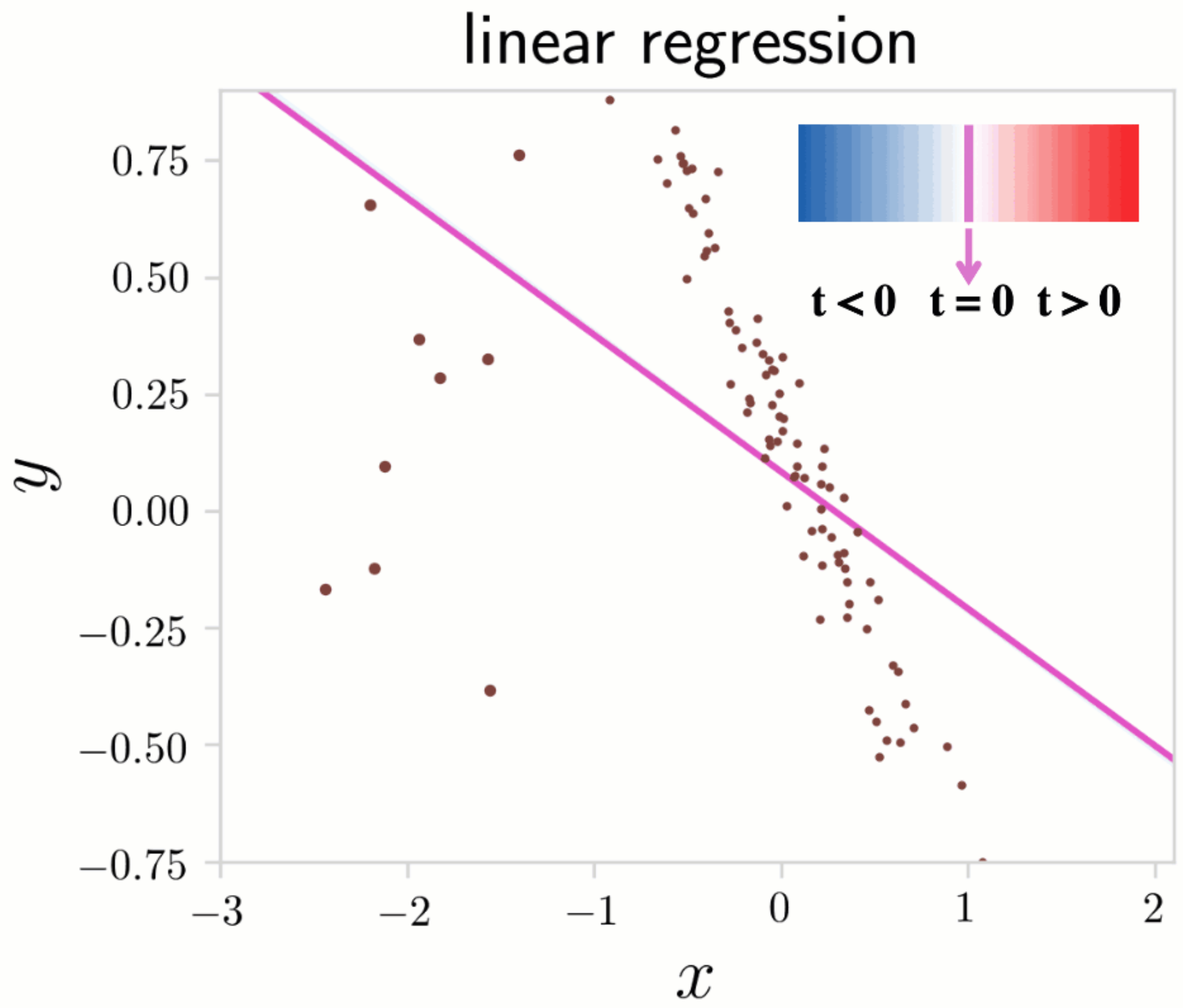


Empirical Results



on average, cut variance of accuracy by 45% while maintaining mean accuracy

Tilted ERM (TERM) Objective



Empirical Risk Minimization

$$\min_w \frac{1}{n} \sum_{i=1}^{n} f(x_i; w)$$

Tilted ERM

$$\min_w \frac{1}{t} \log \left(\frac{1}{n} \sum_{i=1}^{n} e^{t f(x_i; w)} \right)$$

TERM can increase or decrease the influence of outliers to enable fairness or robustness

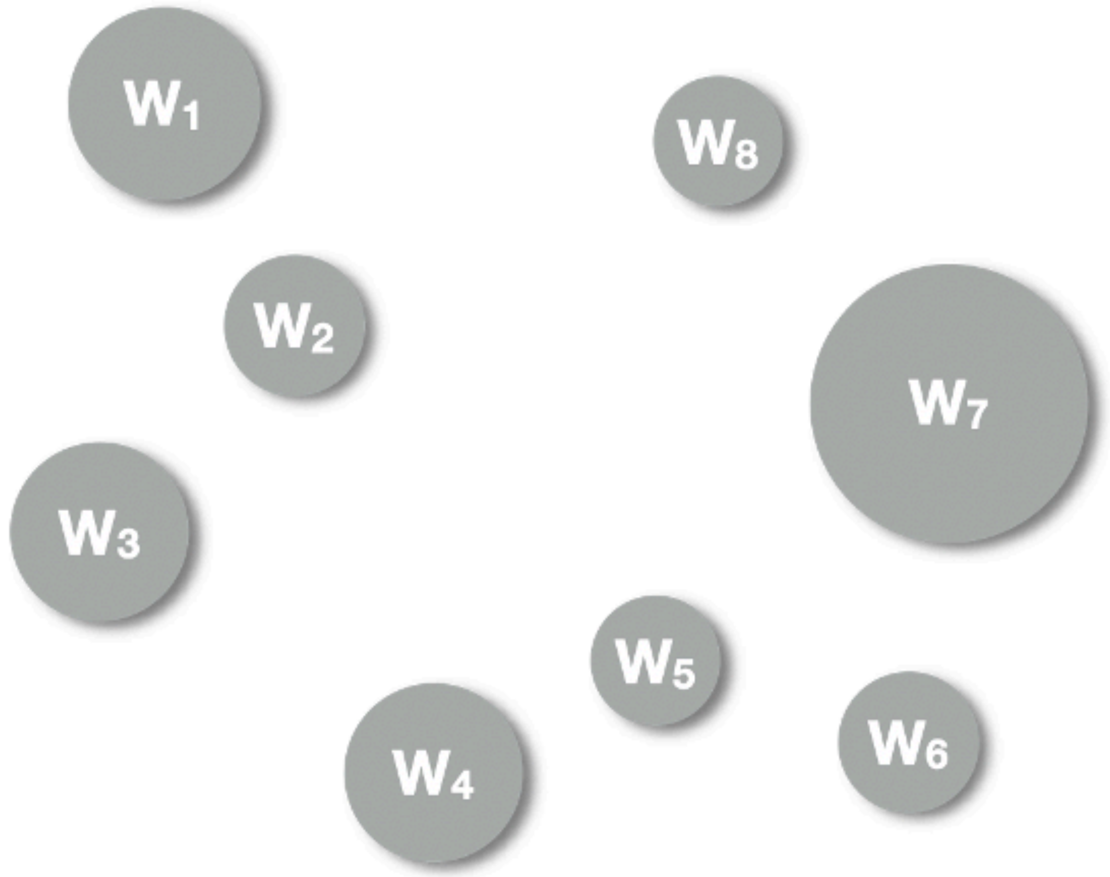
Can we equalize performance across heterogeneous networks?

- Vanilla ERM may deliver poor quality of service in heterogeneous networks
- q-FFL/TERM allows for flexible trade-off between fairness and accuracy

How to model federated data?

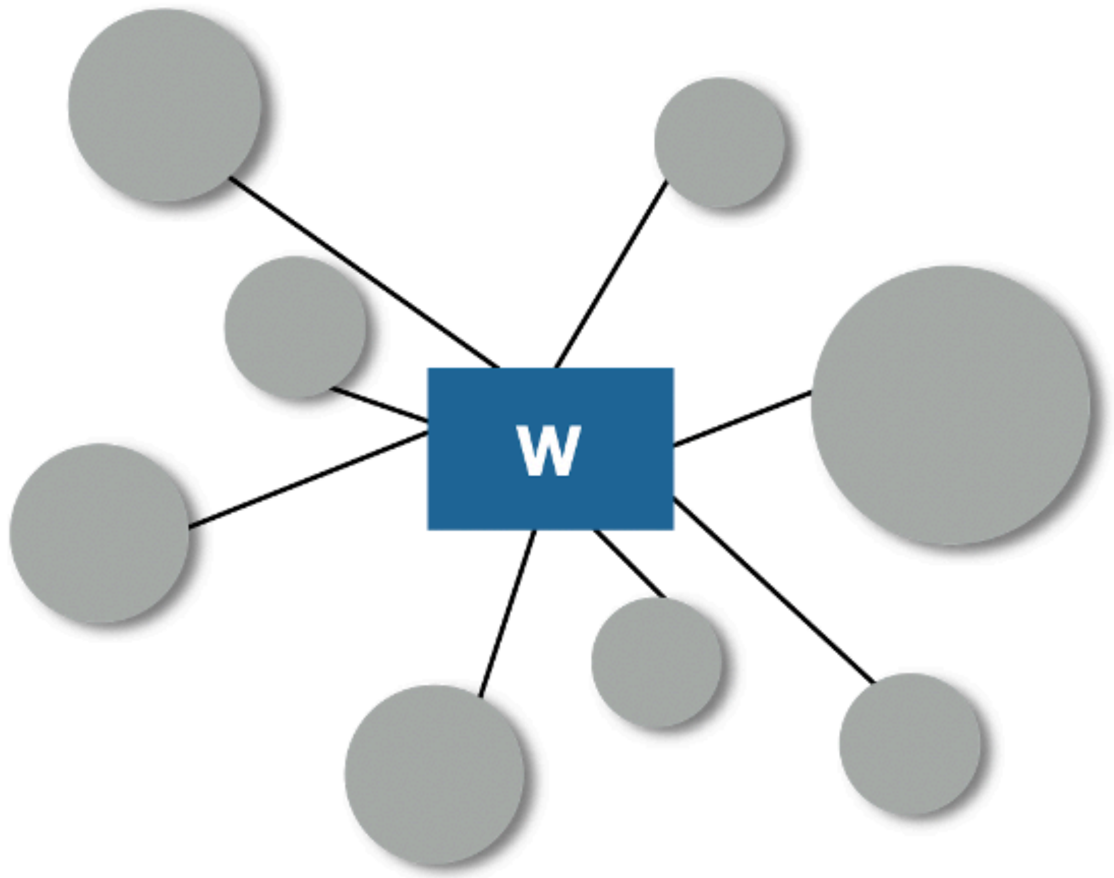
Personalization for Federated Learning

local



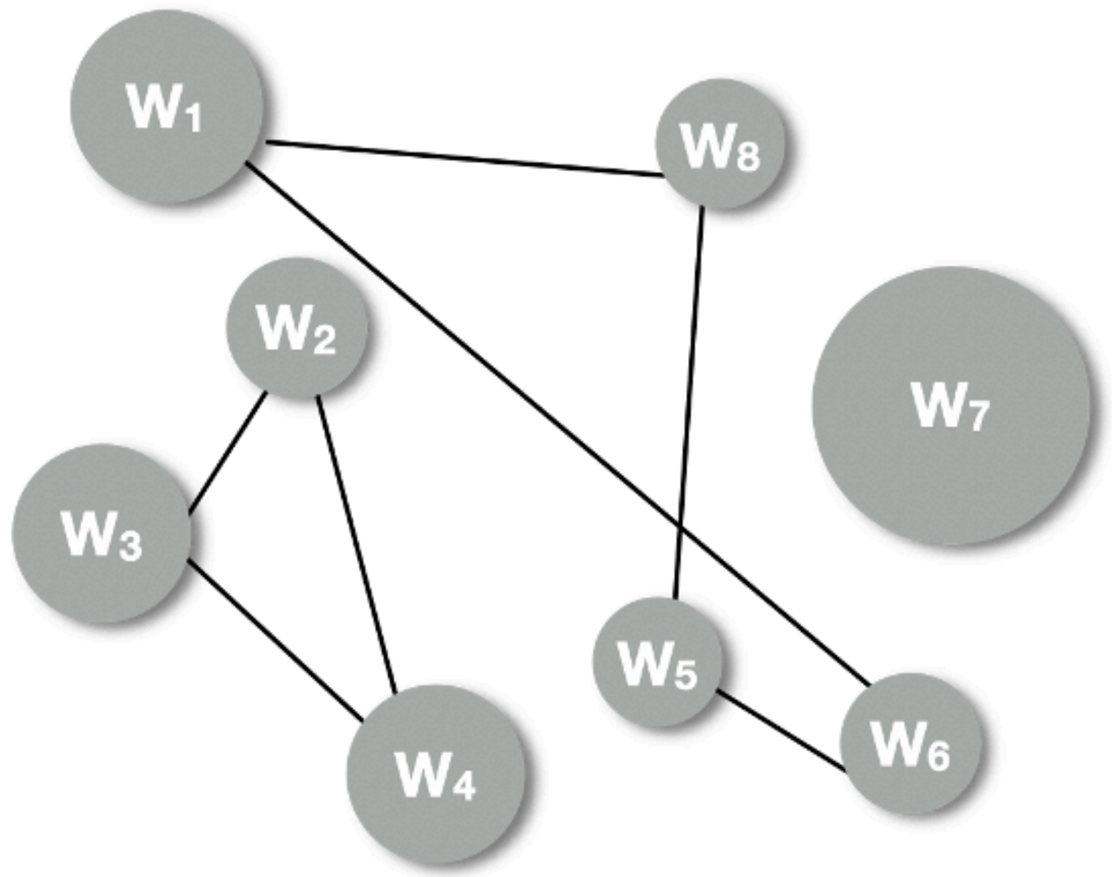
personalized models
not learn from peers

global



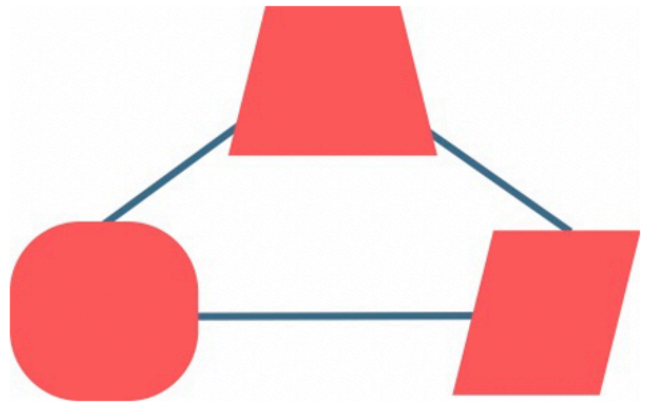
non-personalized models
learn from peers

??



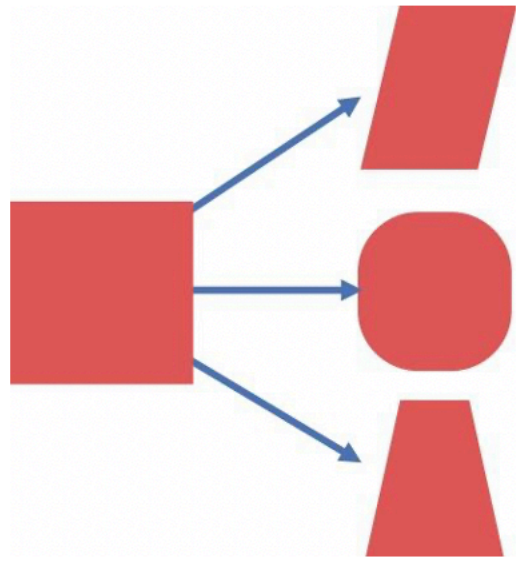
personalized models
learn from peers

Approaches for Personalization



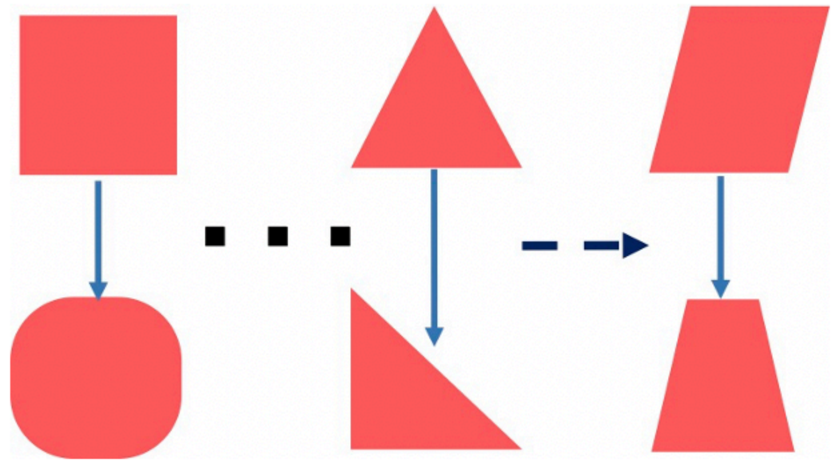
Multi-task Learning

Jointly learn shared, yet personalized models



Fine-tuning

- Learn a global model, then "fine-tune"/adapt it on local data
- See also: transfer learning, domain adaptation



Meta-learning

- Learn initialization over multiple tasks, then train locally

Meta-learning & Federated learning

Algorithm 1 Connects FL and MAML (left), Reptile Batch Version(middle), and FedAvg (right).

OuterLoop/Server learning rate α
InnerLoop/Client learning rate β
Initial model parameters θ
while not done **do**
  Sample batch of tasks/clients $\{T_i\}$
  for Sampled task/client T_i **do**
    if FL **then**
      $g_i, w_i = ClientUpdate(\theta, T_i, \beta)$
    else if MAML **then**
      $g_i = InnerLoop(\theta, T_i, \beta)$
    end if
  end for
  if FL **then**
    $\theta = ServerUpdate(\theta, \{g_i, w_i\}, \alpha)$
  else if MAML **then**
    $\theta = OuterLoop(\theta, \{g_i\}, \alpha)$
  end if
end while

Require: : Reptile Step K.
  function $InnerLoop(\theta, T_i, \beta)$
    Sample K-shot data $D_{i,k}$ from T_i.
    $\theta_i = \theta$
    for each local step i from 1 to K **do**
      $\theta_i = \theta_i - \beta \nabla_\theta L(\theta_i, D_{i,k})$
    end for
    Return $g_i = \theta_i - \theta$
  end function
Require: : Meta Batch Size M.
  function $OuterLoop(\theta, \{g_i\}, \alpha)$
    $\theta = \theta + \alpha \frac{1}{M} \sum_{i=1}^{M} g_i$
    Return θ
  end function

Require: FedAvg Local Epoch E.
  function $ClientUpdate(\theta, T_i, \beta)$
    Split local dataset into batches B
    $\theta_i = \theta$
    for each local epoch i from 1 to E **do**
      for batch $b \in B$ **do**
        $\theta_i = \theta_i - \beta \nabla_\theta L(\theta_i, b)$
      end for
    end for
    Return $g_i = \theta_i - \theta$
  end function
Require: Clients per training round M.
  function $ServerUpdate(\theta, \{g_i, w_i\}, \alpha)$
    $\theta = \theta + \alpha \sum_{i=1}^{M} w_i g_i / \sum_{i=1}^{M} w_i$
    Return θ
  end function

[Jiang et al, Improving federated learning personalization via model agnostic meta learning, 2019]
[Khodak, Balcan, Talwalkar, Adaptive gradient-based meta-learning methods, NeurIPS 2019

Personalization for Practical Constraints

constraints in federated learning

fairness — *representation disparity*

robustness — *against data and model poisoning attacks*

privacy

security

communication

……

competing with each other

$$w^* \in \arg\min_{w} G\left(F_1(w), \ldots, F_k(w)\right)$$

Ditto: Fair and Robust Federated Learning Through Personalization
Li, Hu, Beirami, Smith, ArXiv 2021
Best paper at ICLR Secure ML Workshop

Ditto: Global-regularized Federated MTL

personalization to achieve robustness and fairness simultaneously

for each device k,

Ditto:

$$\min_{v_k} \quad h_k(v_k; w^*) := \underbrace{F_k(v_k)}_{\text{local loss}} + \underbrace{\frac{\lambda}{2}\|v_k - w^*\|^2}_{\text{global-regularized}}$$

$$\text{s.t.} \quad \boxed{w^*} \in \arg\min_{w} G\left(F_1(w), \ldots, F_k(w)\right)$$

- simple form of MTL: ensure personalized models are close to global model
- easy to implement in federated settings
- accurate, robust, and fair

Ditto Solver

solver for the global model w^* + personalization add-on

Algorithm 1: `Ditto` for Personalized FL

Input: K, T, s, λ, η, w^0, $\{v_k^0\}_{k\in[K]}$

for $t = 0, \cdots, T-1$ **do**

Server randomly selects a subset of devices S_t, and sends the current global model w^t to them

for *device* $k \in S_t$ *in parallel* **do**

Solve the local sub-problem of $G(\cdot)$ inexactly starting from w^t to obtain w_k^t:

$$w_k^t \leftarrow \text{UPDATE\_GLOBAL}(w^t, \nabla F_k(w^t))$$

`/* Solve` $h_k(v_k; w^t)$ `*/`

Update v_k for s local iterations:

$$v_k = v_k - \eta(\nabla F_k(v_k) + \lambda(v_k - w^t))$$

Send $\Delta_k^t := w_k^t - w^t$ back

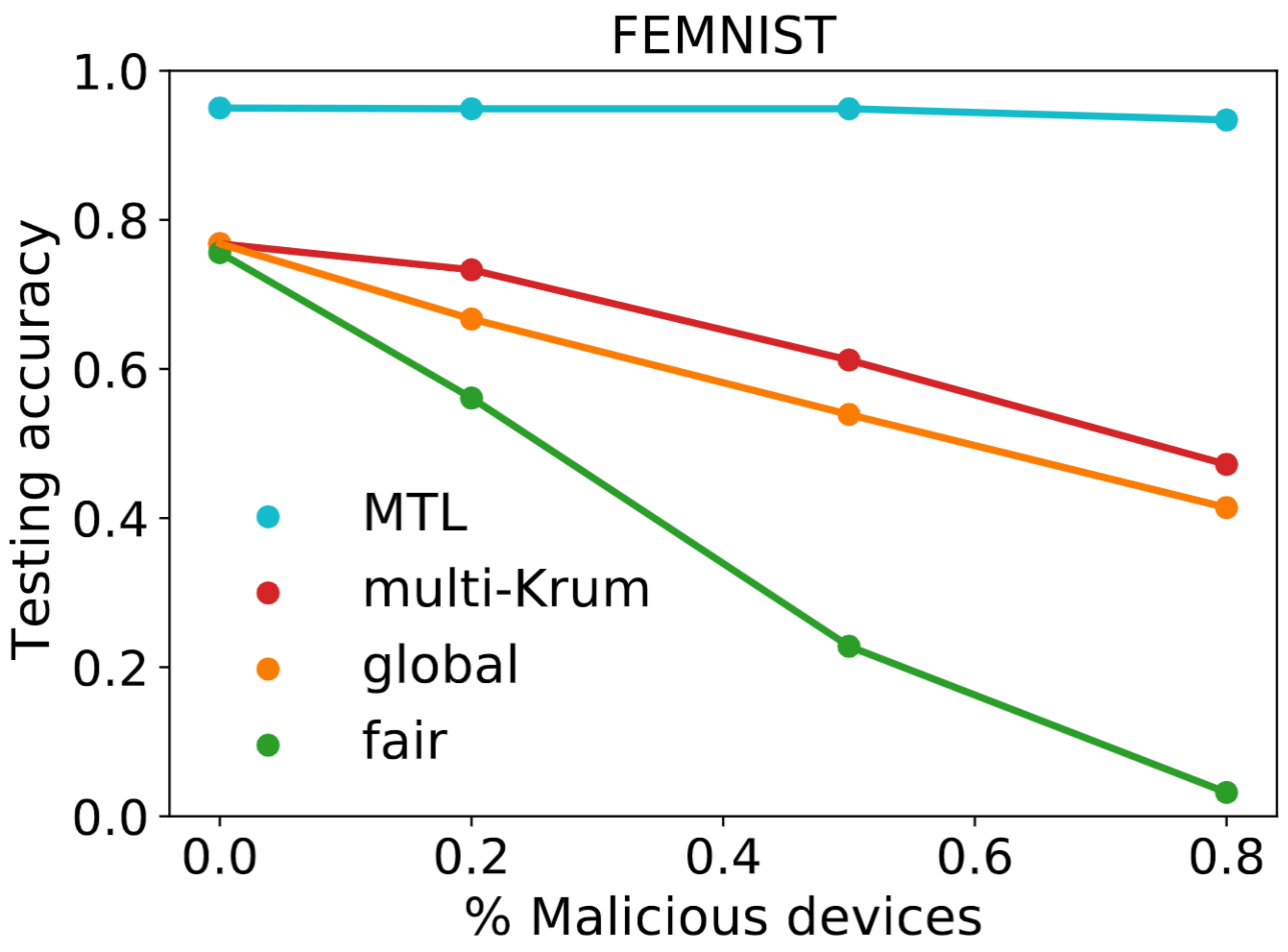
Server aggregates $\{\Delta_k^t\}$:

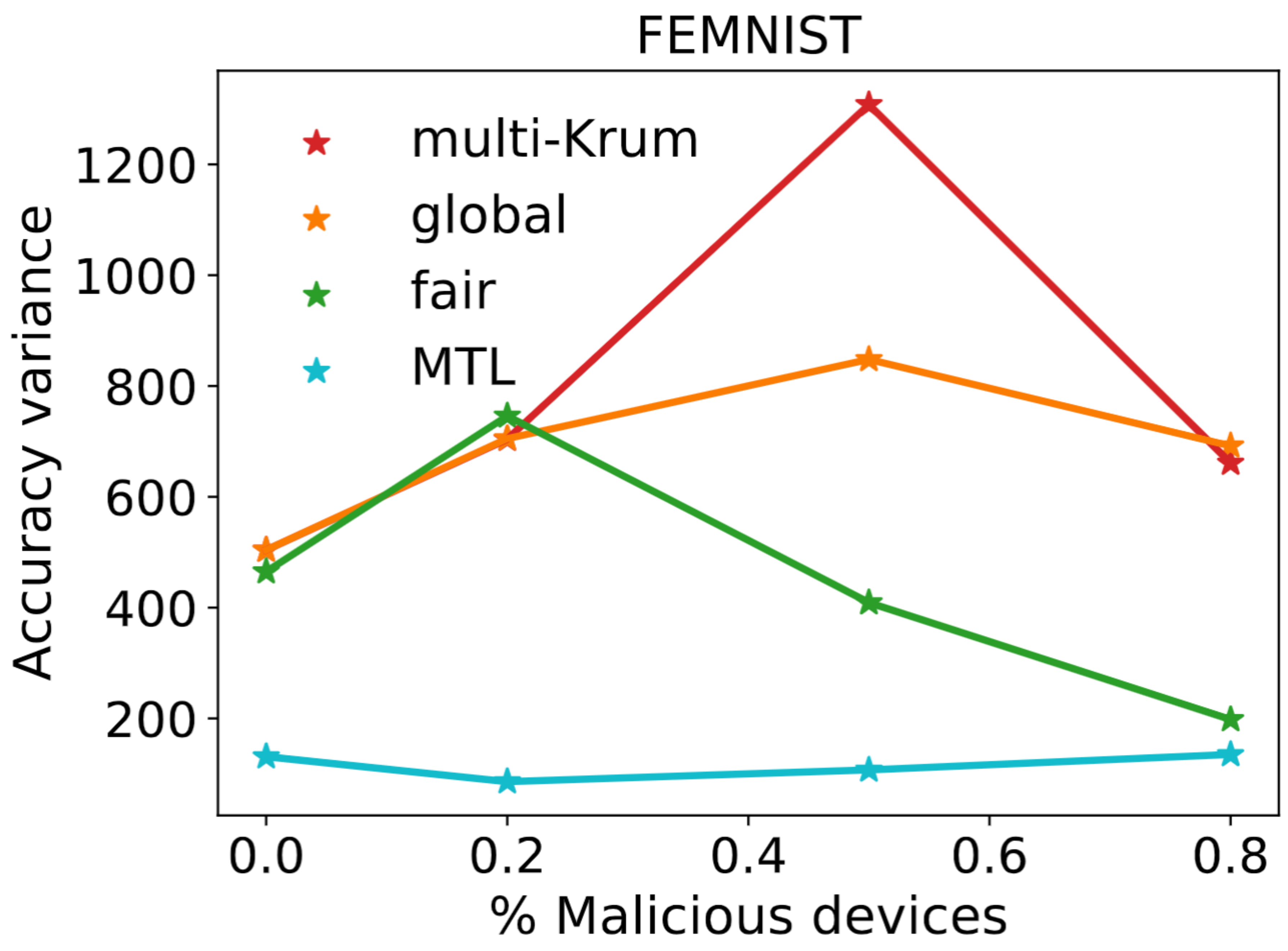
$$w^{t+1} \leftarrow \text{AGGREGATE}\left(w^t, \{\Delta_k^t\}_{k\in\{S_t\}}\right)$$

return $\{v_k\}_{k\in[K]}$ *(personalized models)*, w^T *(global model)*

- a scalable, simple personalization add-on for any federated global solver
- preserves the practical properties of the global FL solver (e.g., communication, privacy)
- with convergence guarantees

Experiments: Competing Constraints

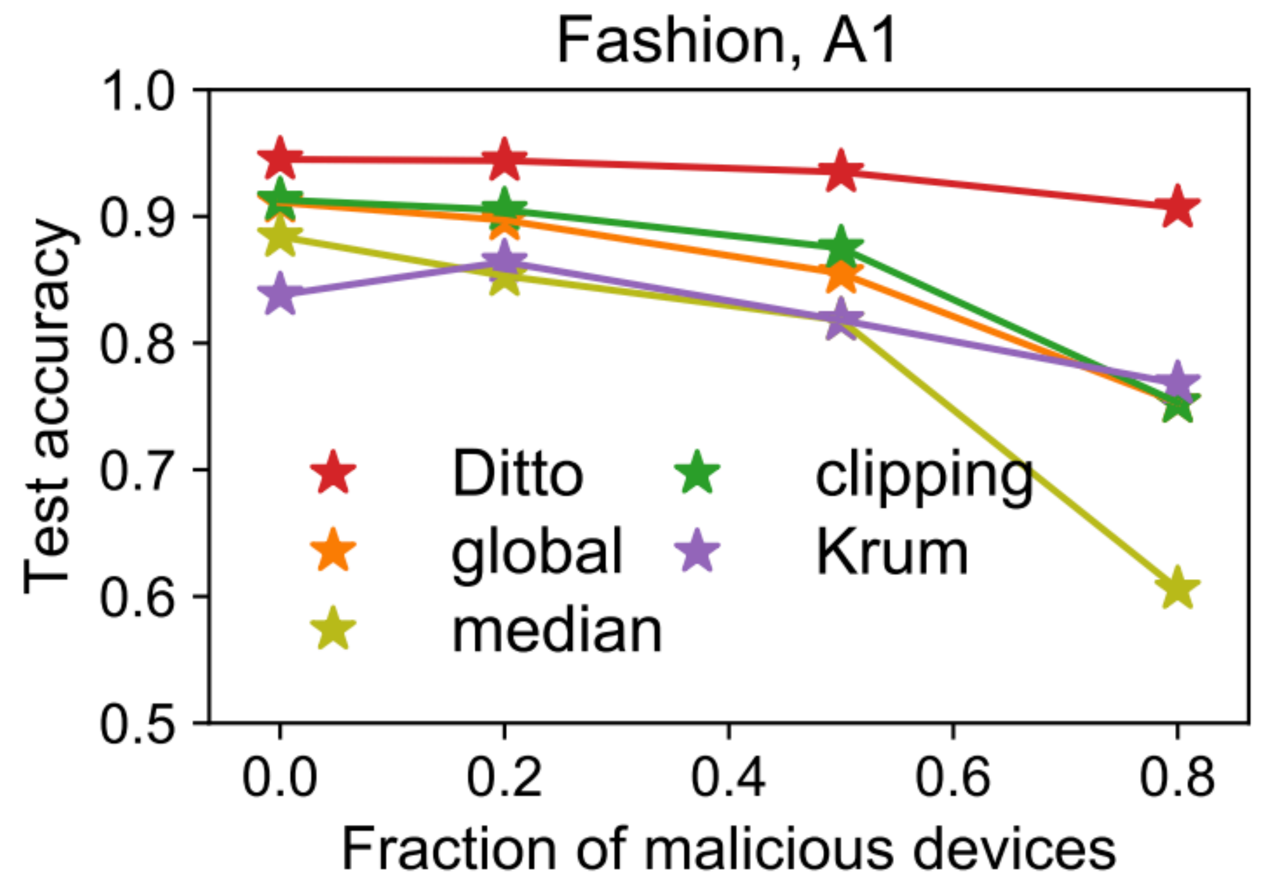


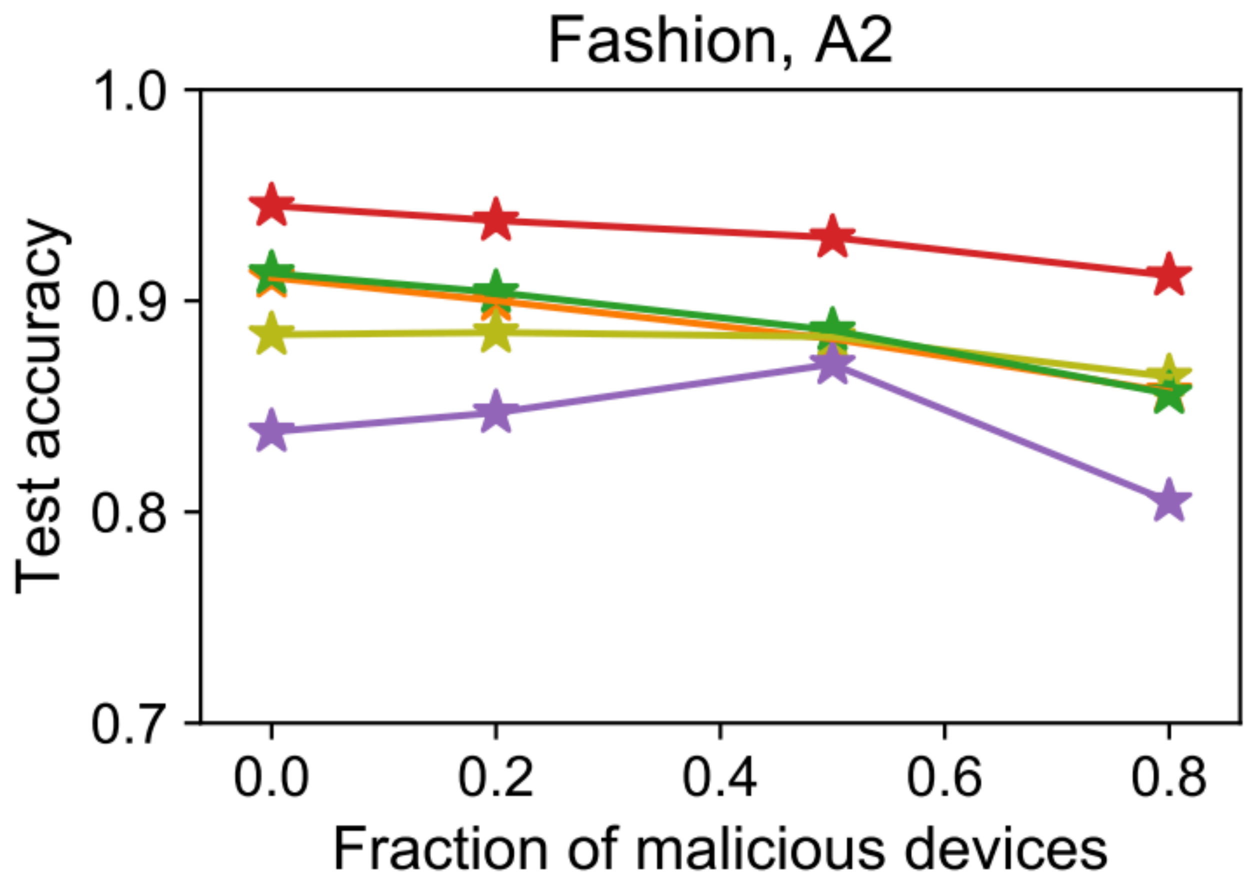


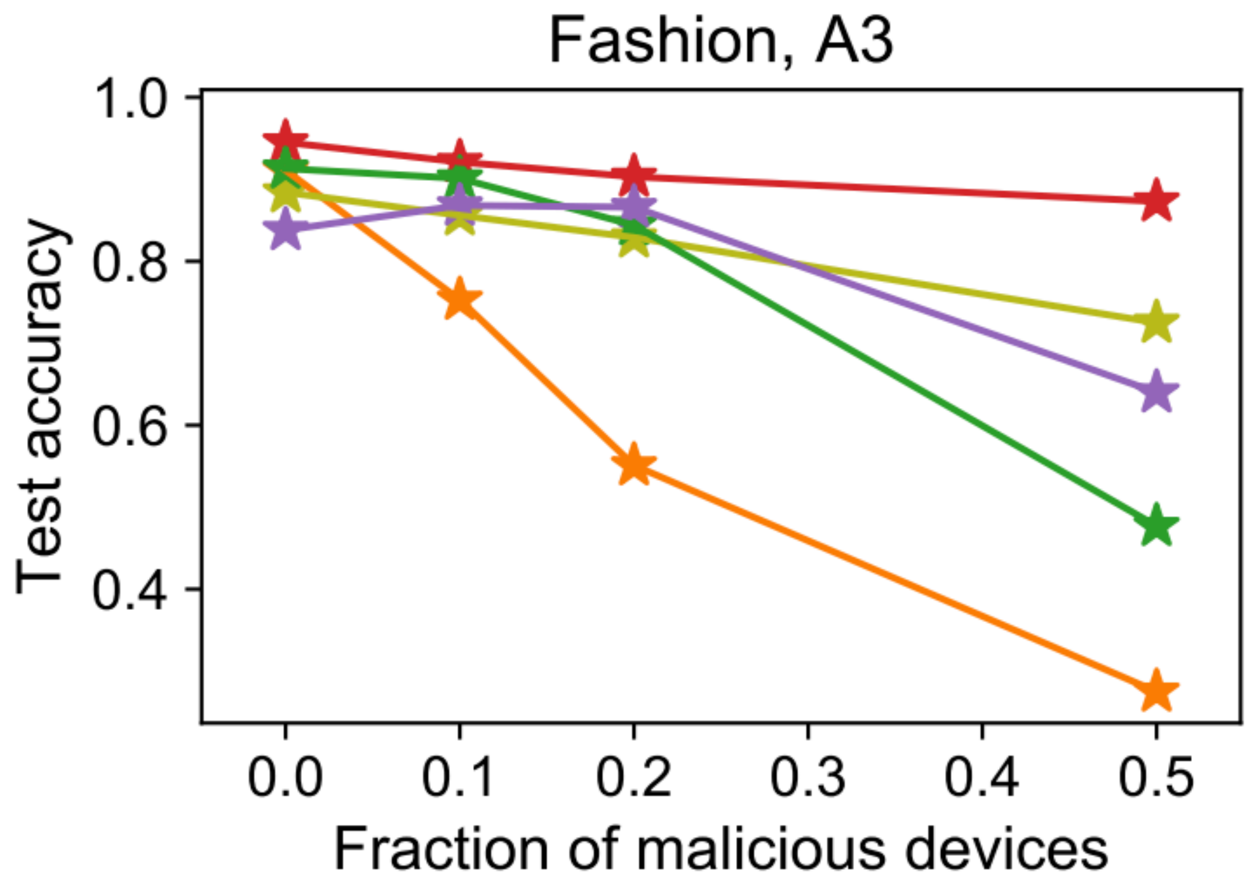
fair methods are not robust

robust methods are not fair (with high variance)

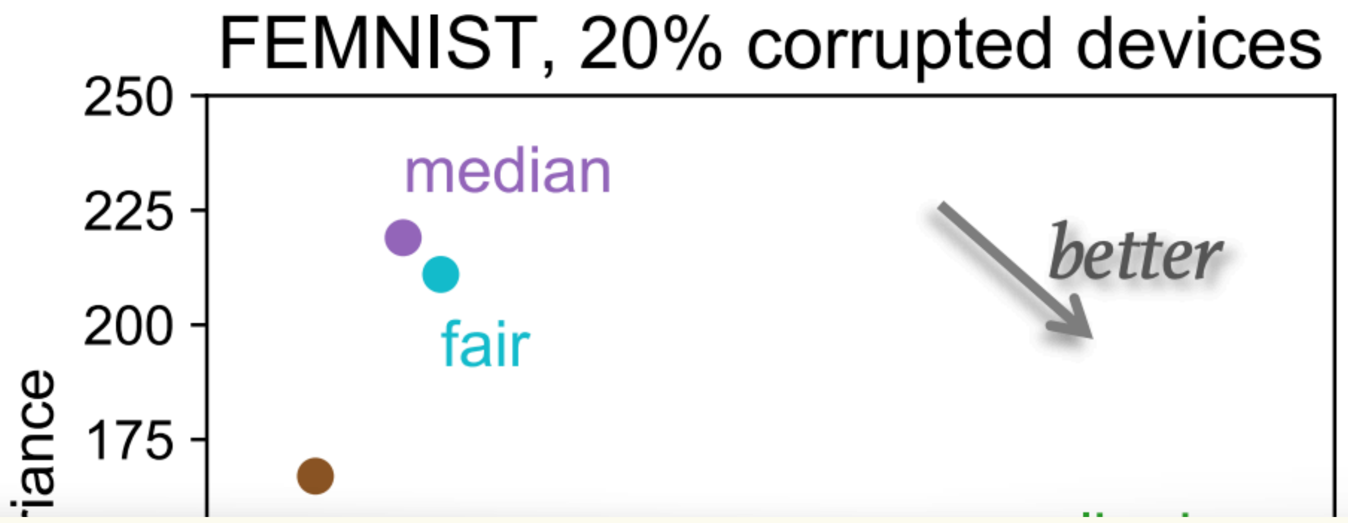
Experiments: Benefits of Personalization







Ditto is more robust than strong baselines under various attacks



Ditto is also more fair

on average, improve absolute accuracy by ~6% over the strongest robust baseline

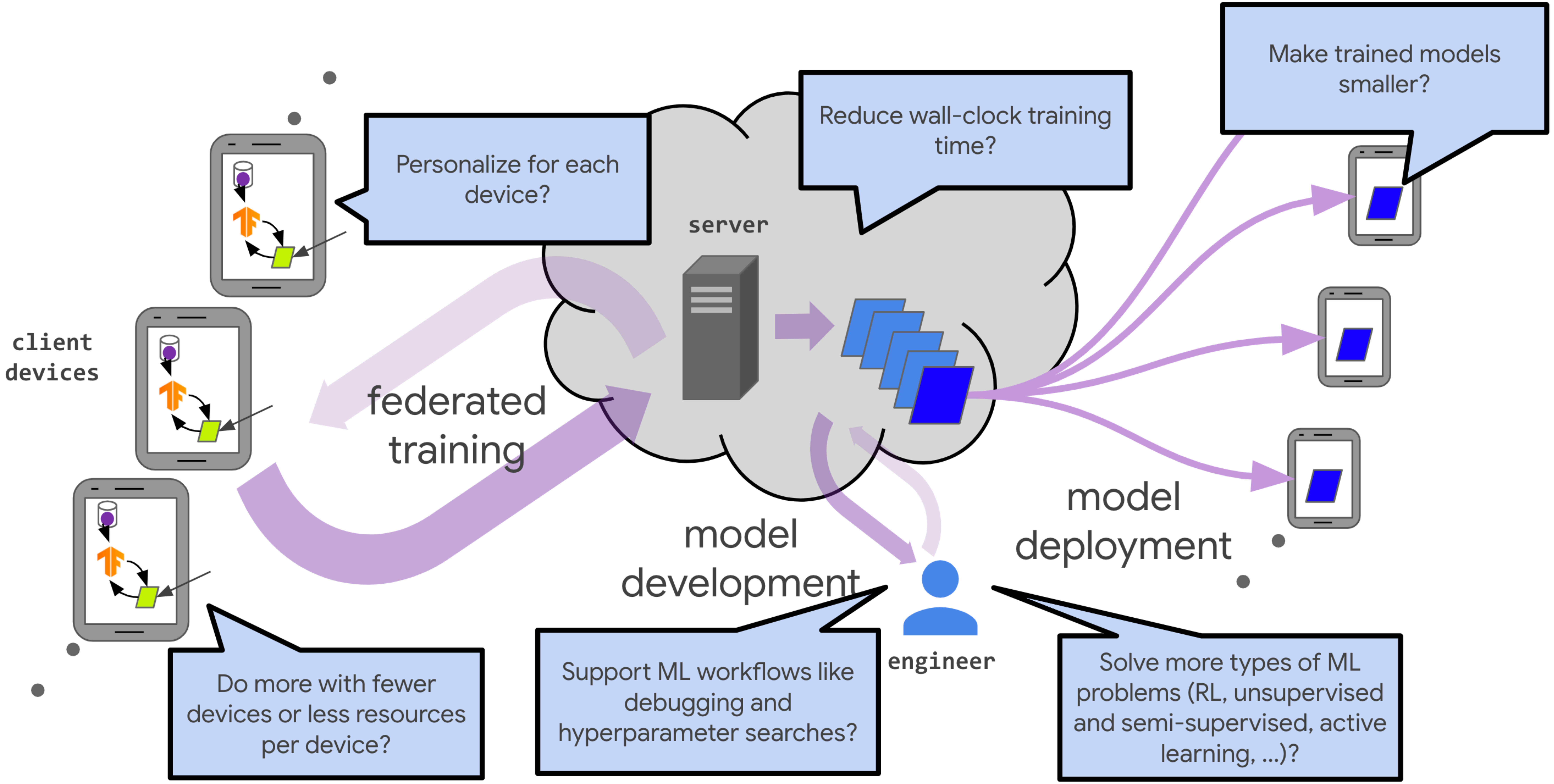
reduce variance by ~10% over STOA fair methods

How to model federated data?

- Personalization is a promising approach (need to be scalable, automated)
- Personalization has additional benefits beyond accuracy, e.g., fairness, robustness, etc.

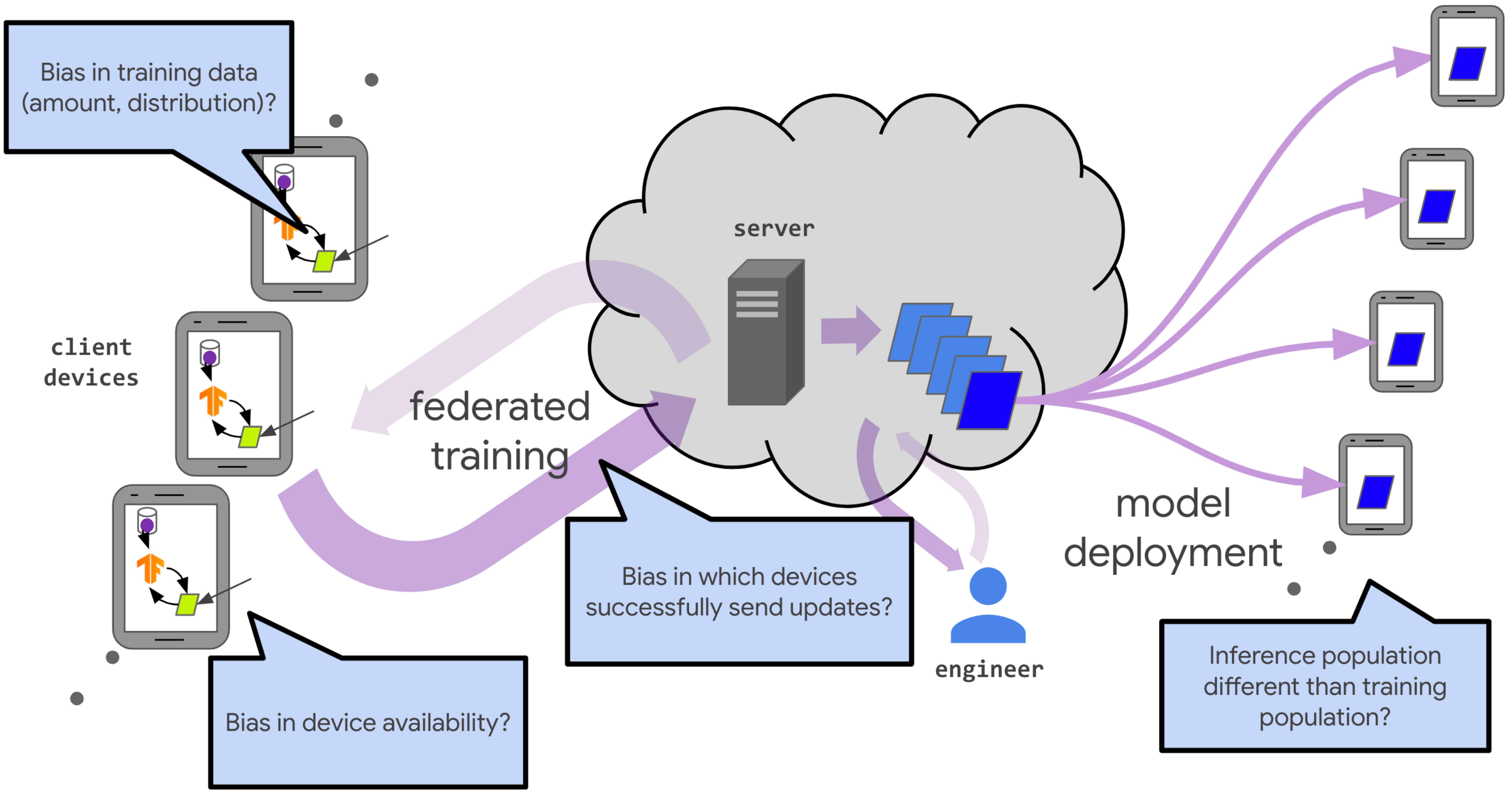
What’s next??

Improving efficiency and effectiveness



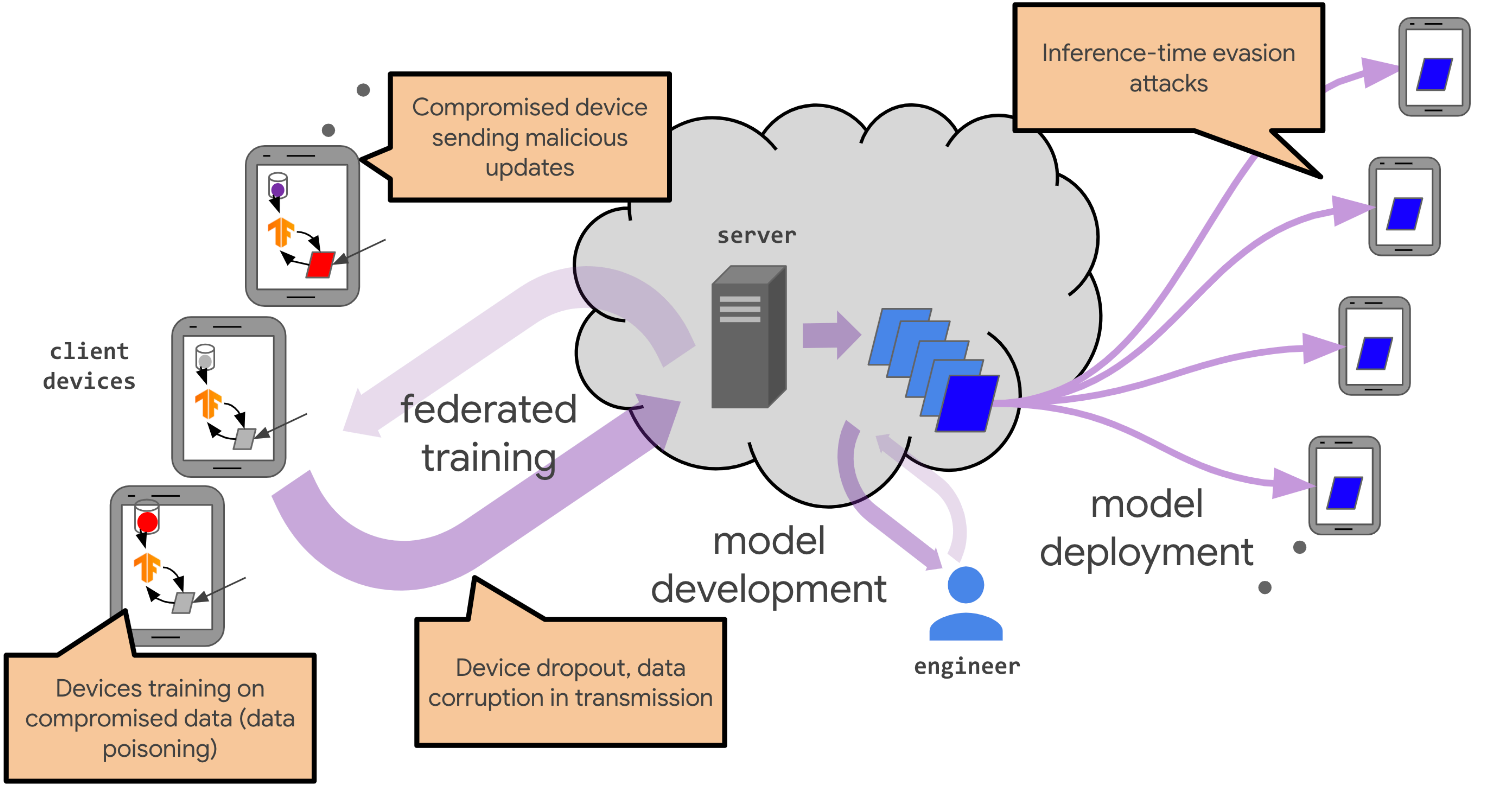
[Credit: B. McMahan, FL Tutorial, NeurIPS 2020]

Ensuring fairness and addressing sources of bias



[Credit: B. McMahan, FL Tutorial, NeurIPS 2020]

Robustness to attacks and failures



[Credit: B. McMahan, FL Tutorial, NeurIPS 2020]

Additional Reading

- **FedAvg: Communication-Efficient Learning of Deep Networks from Decentralized Data**, McMahan et al, AISTATS 2017
- **MOCHA: Federated Multi-Task Learning**, Smith et al, NeurIPS 2017
- [White Paper] **Federated Learning: Challenges, Methods, and Future Directions**, Li et al, IEEE Signal Processing Magazine, 2020
- **NeurIPS 2020 federated learning tutorial**, https://sites.google.com/view/fl-tutorial

Questions?

Tian Li

tianli@cmu.edu