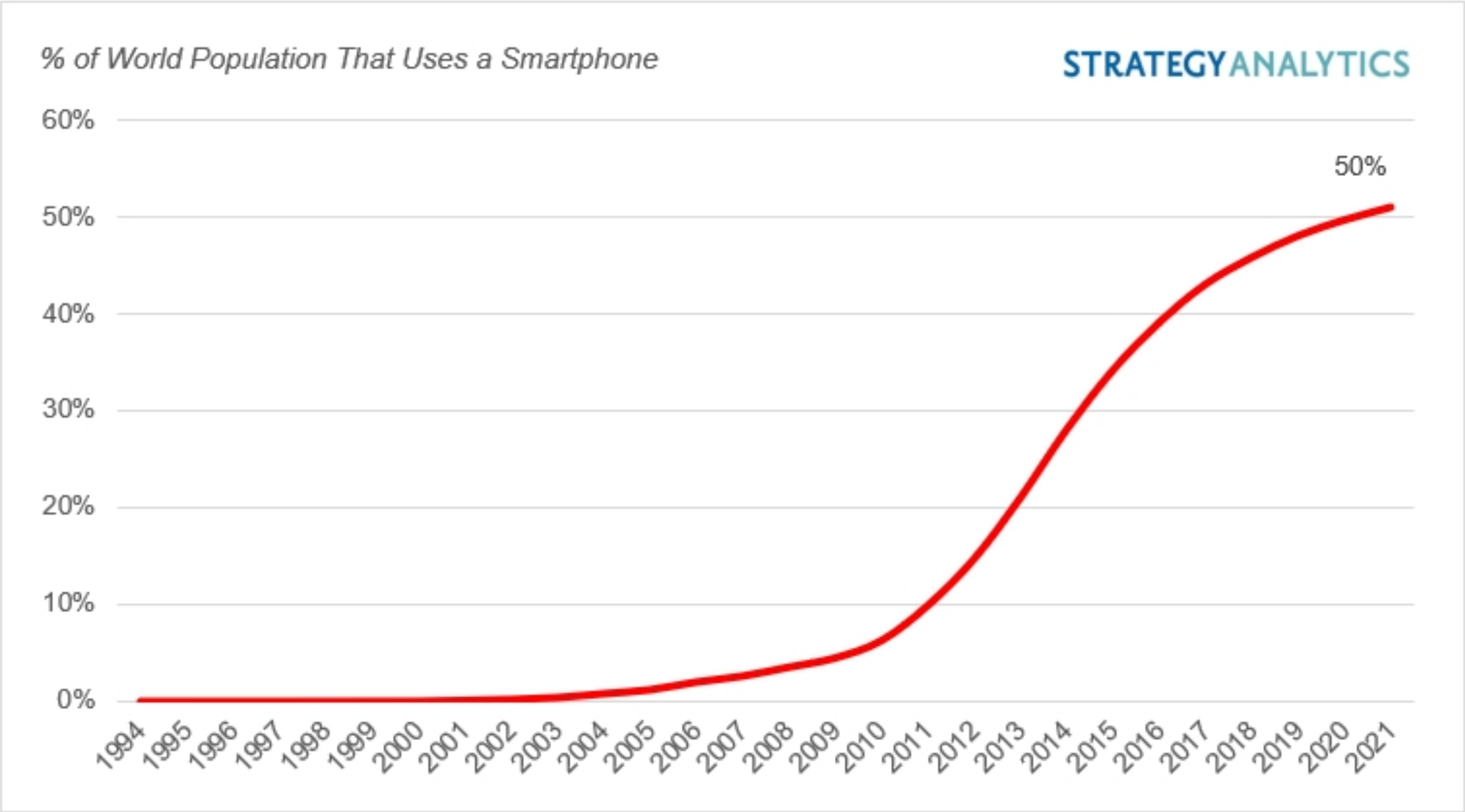
Federated Learning with Heterogeneous Users: A Superquantile Optimization Approach

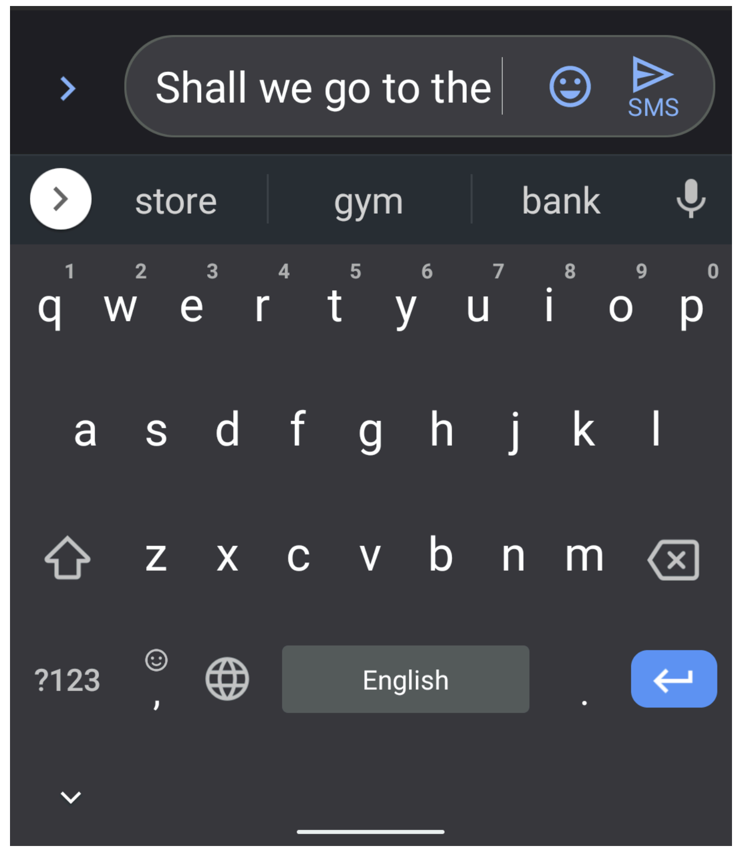
IEEE CISS 2021. Long version under review.

March 14 @ INFORMS

Krishna Pillutla,

Yassine Laguel, Jérôme Malick, Zaid Harchaoui





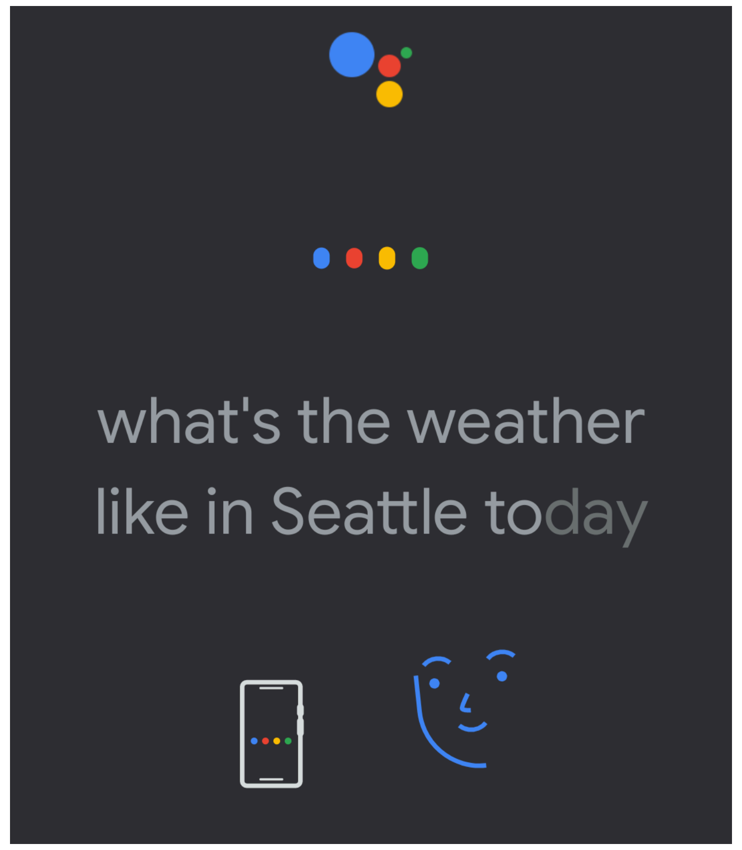
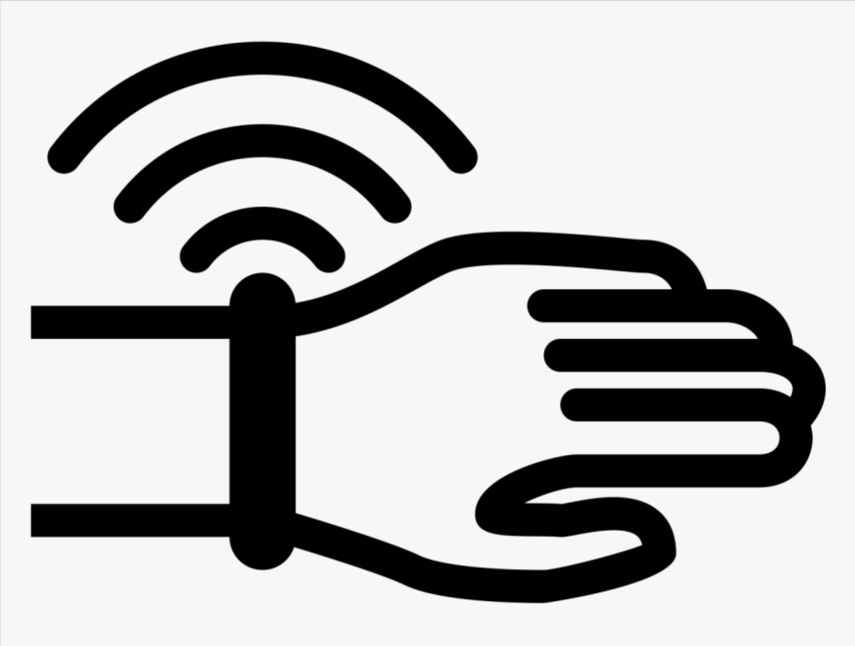
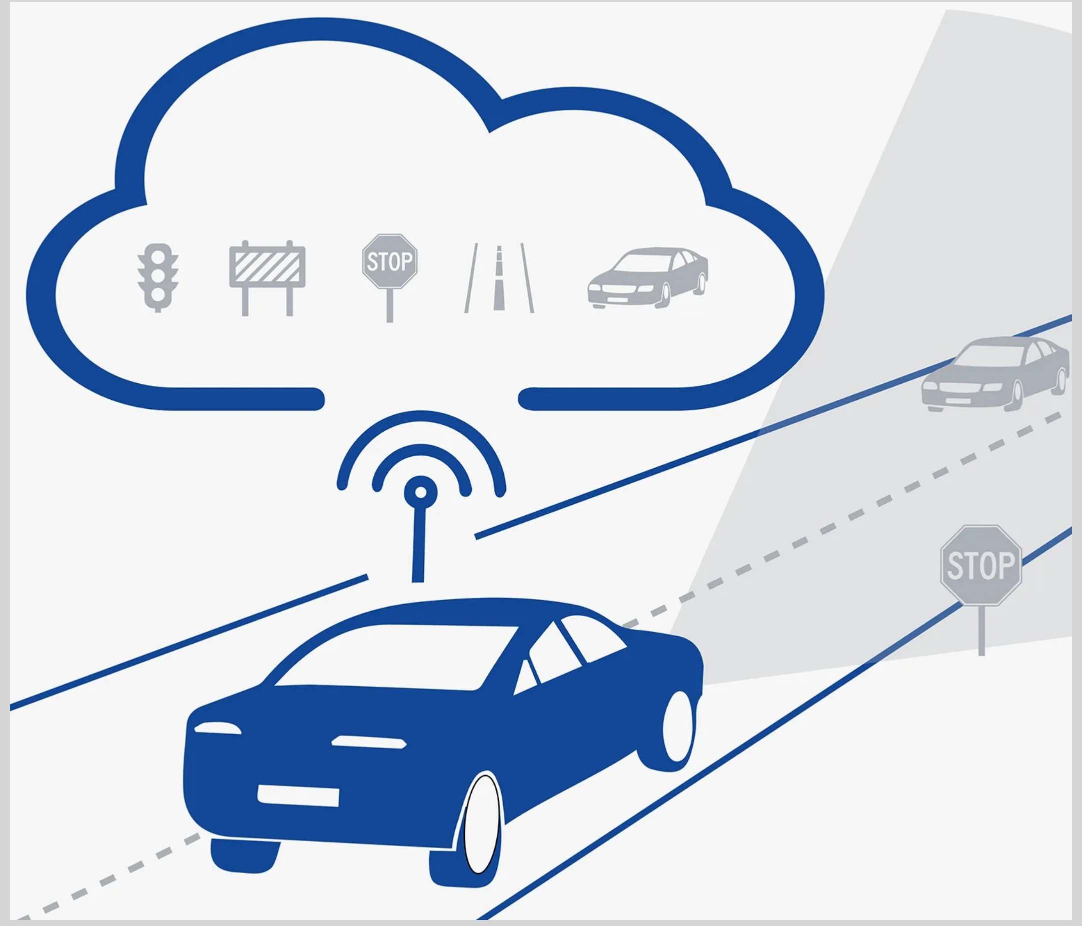
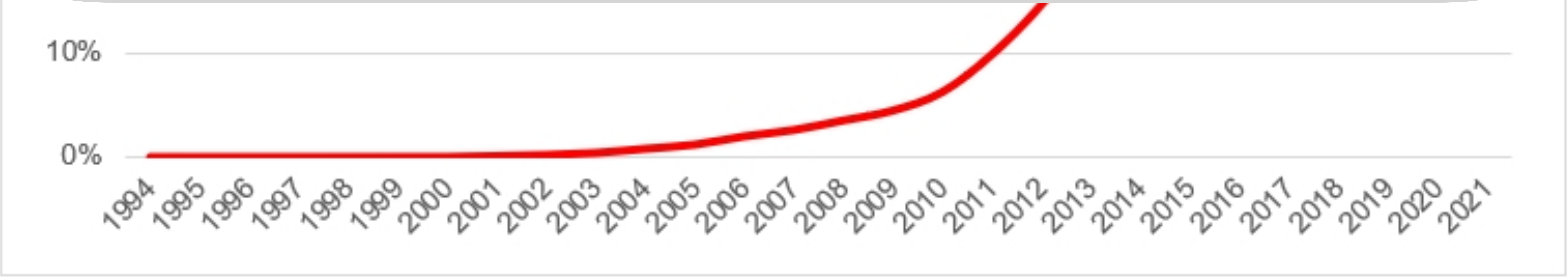
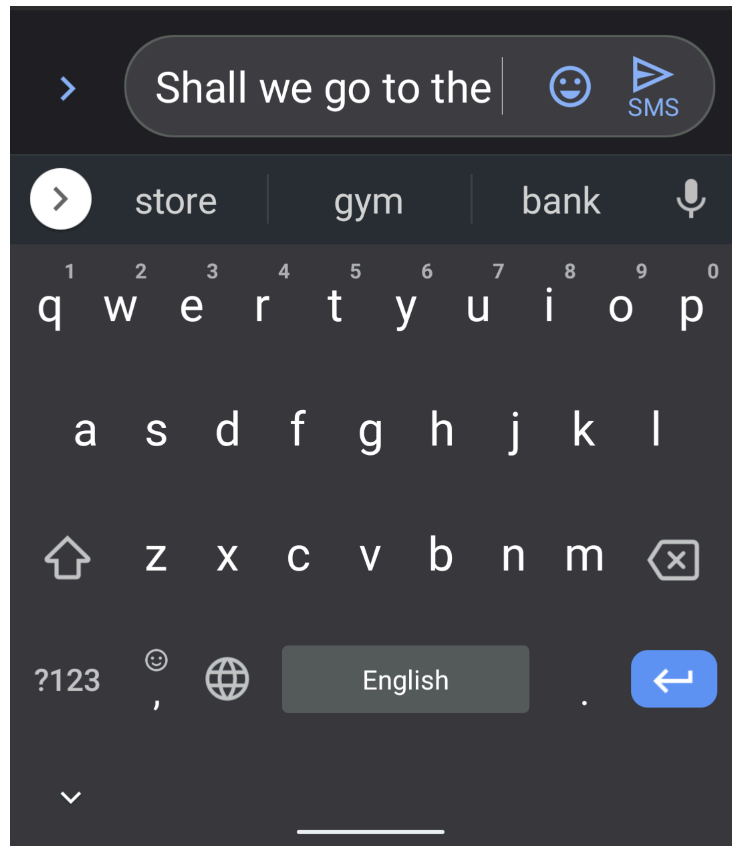


Image Credit: Business Wire









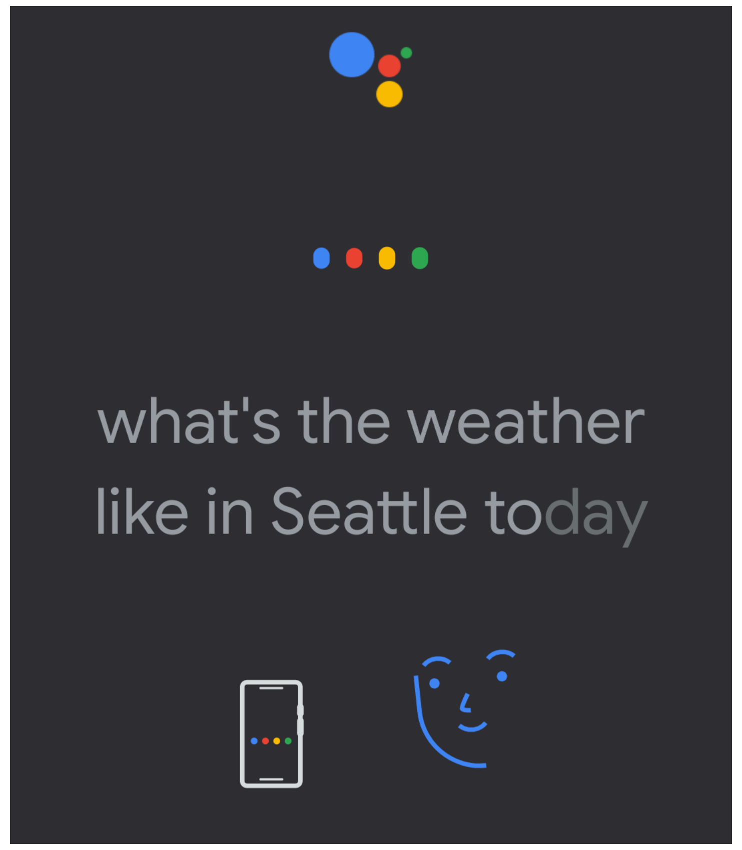
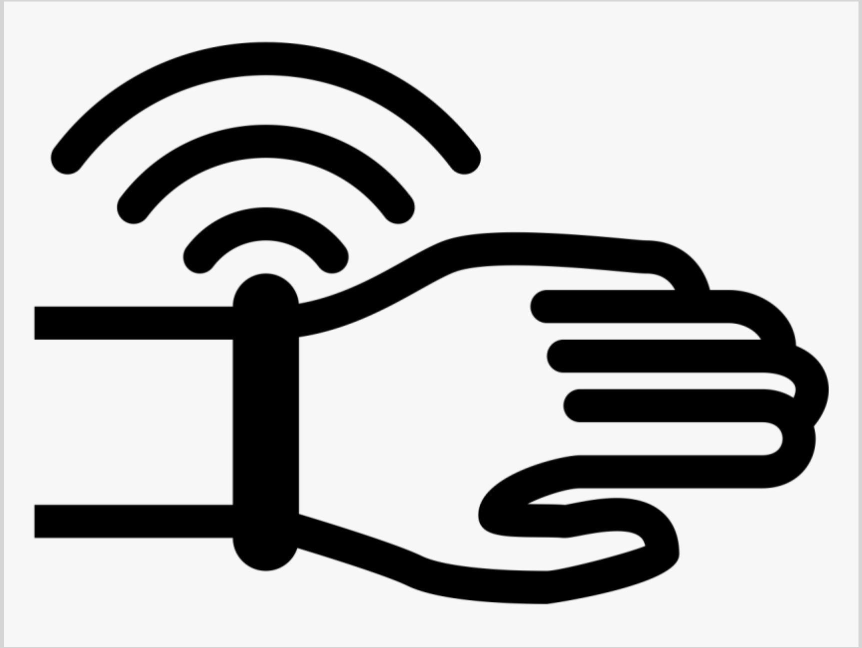
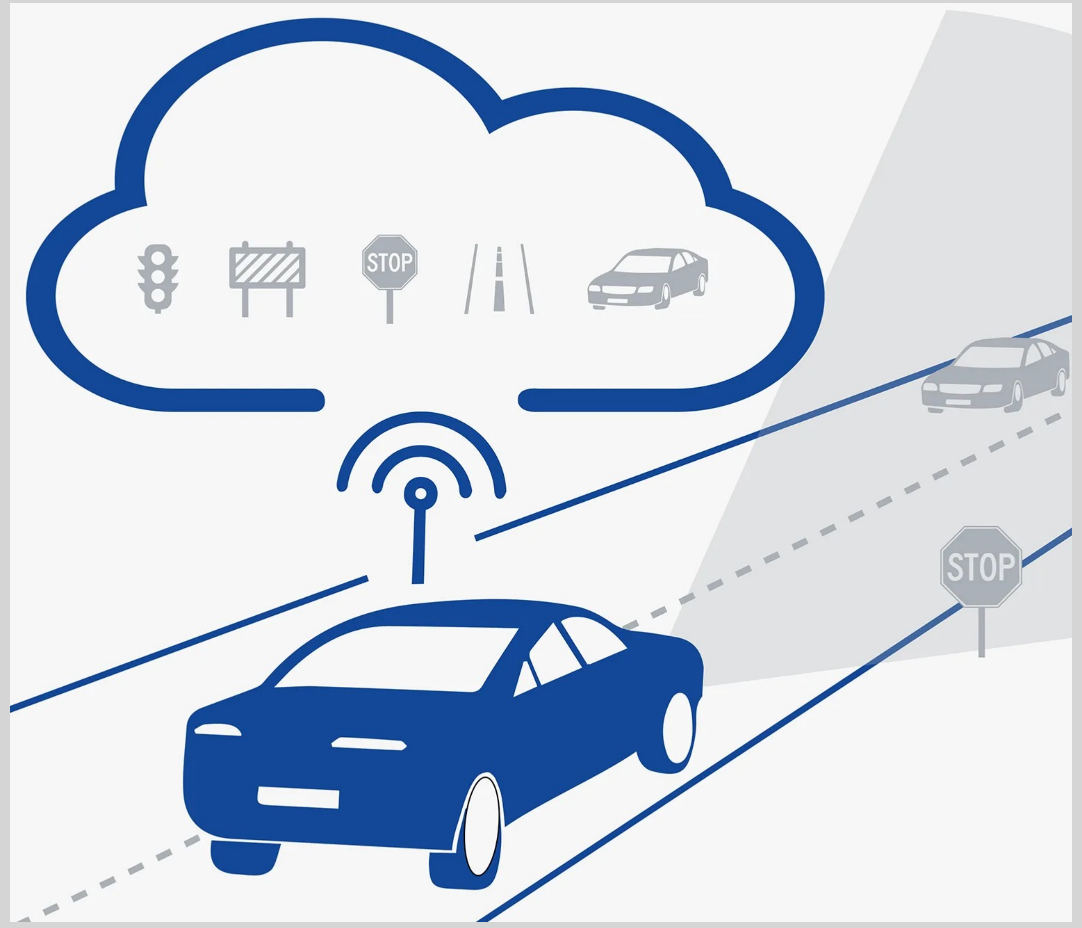
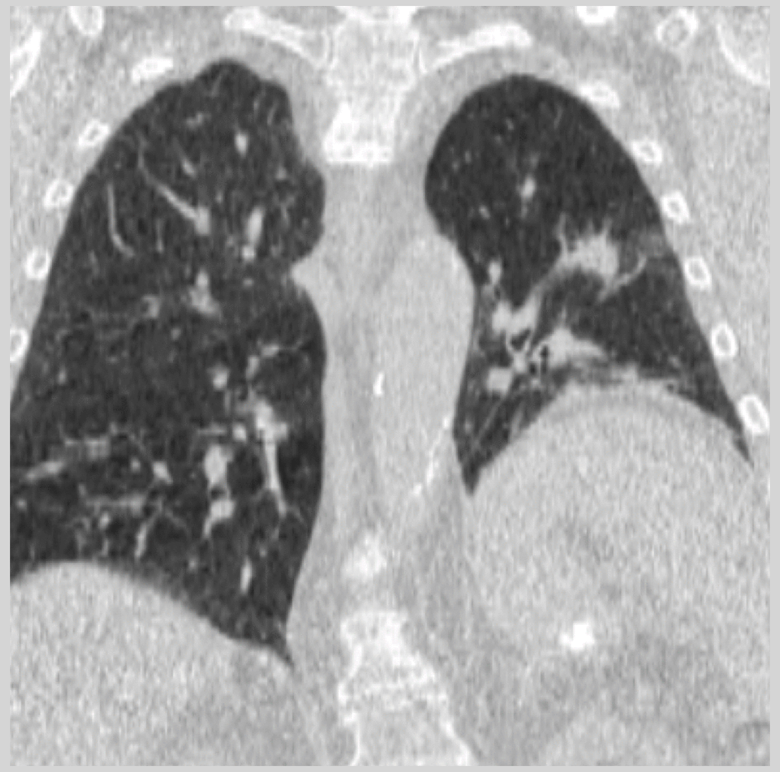


Image Credit: Business Wire

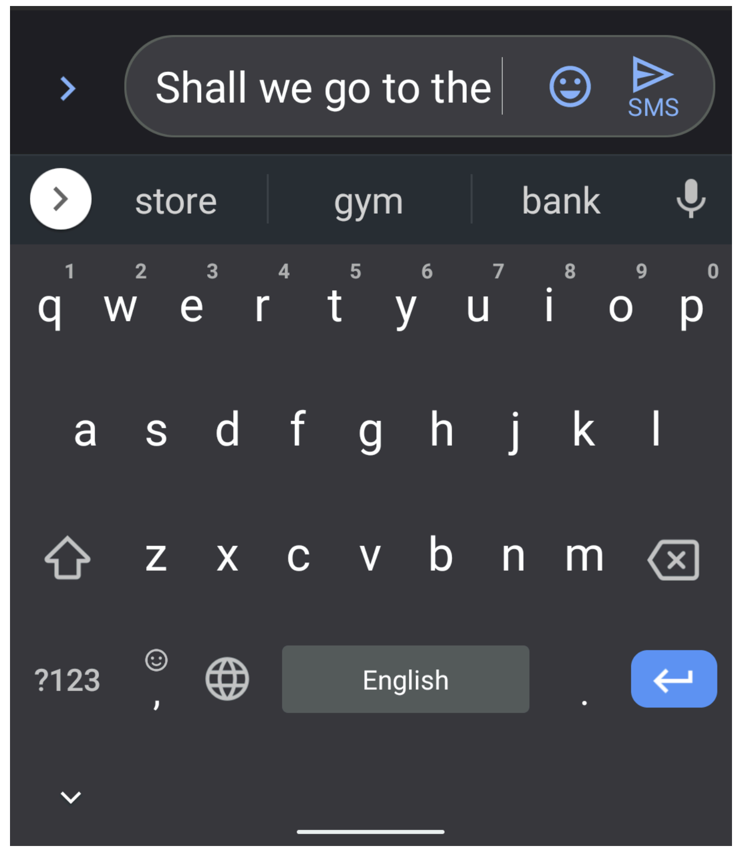


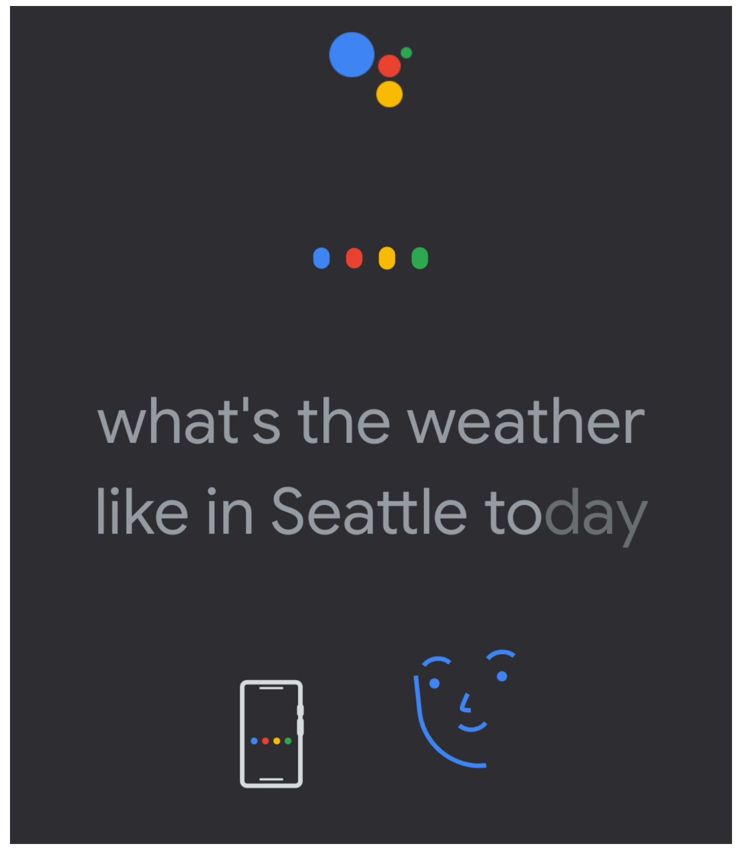

IOT
TV


STOP
STOP



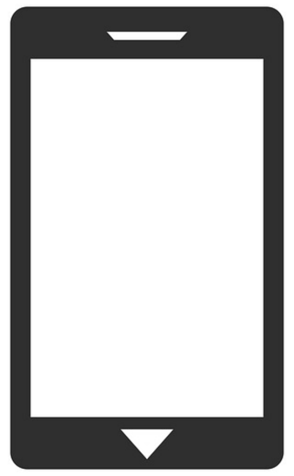
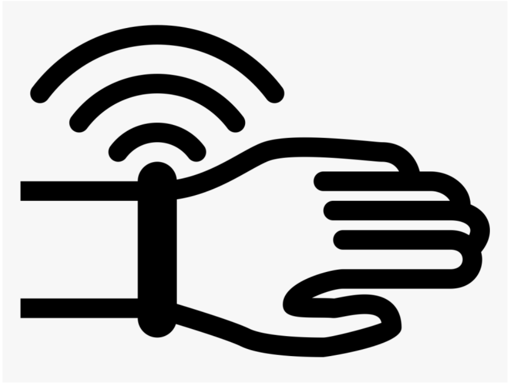

HOSPITAL

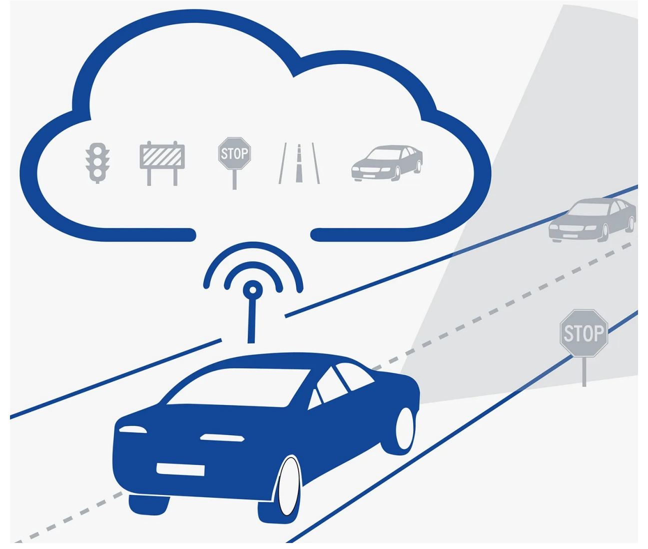

Shall we go to the
SMS
store
gym
bank
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
English

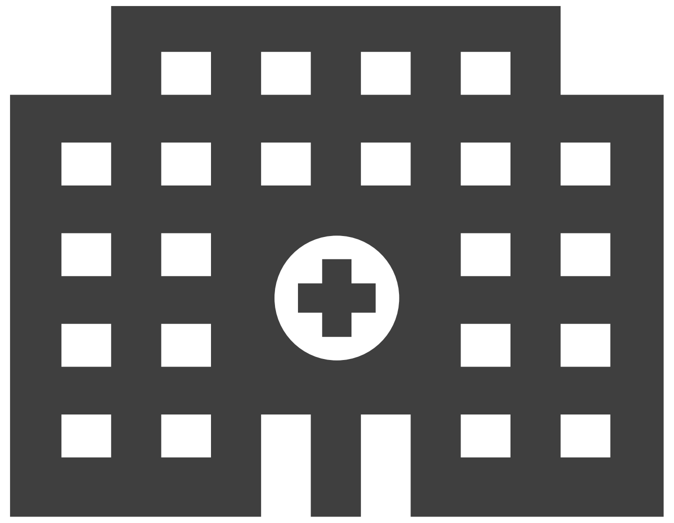

what's the weather
like in Seattle today

Federated Learning

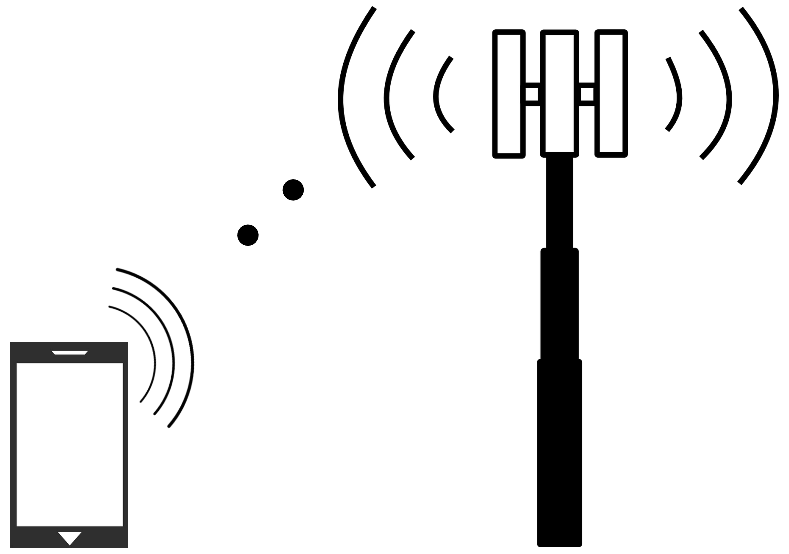
Machine learning has moved from the data centers to edge devices

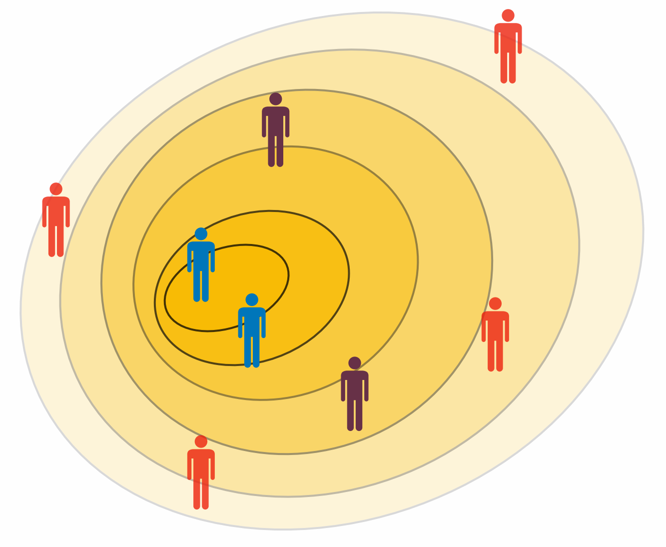



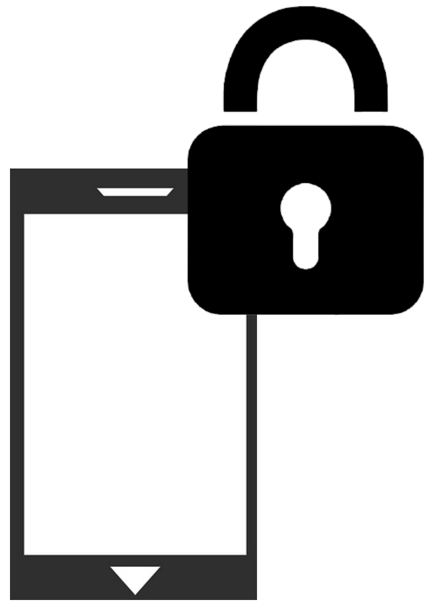
Challenges:



Communication efficiency



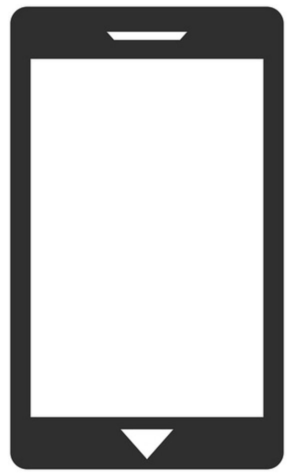
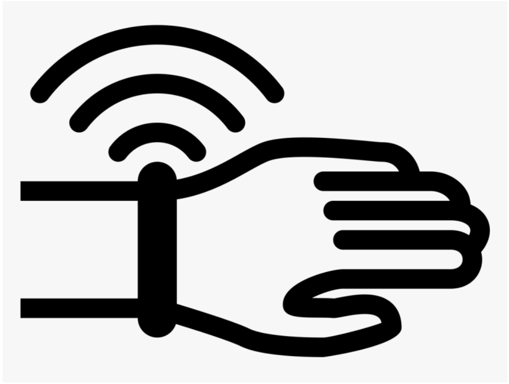
Statistical heterogeneity

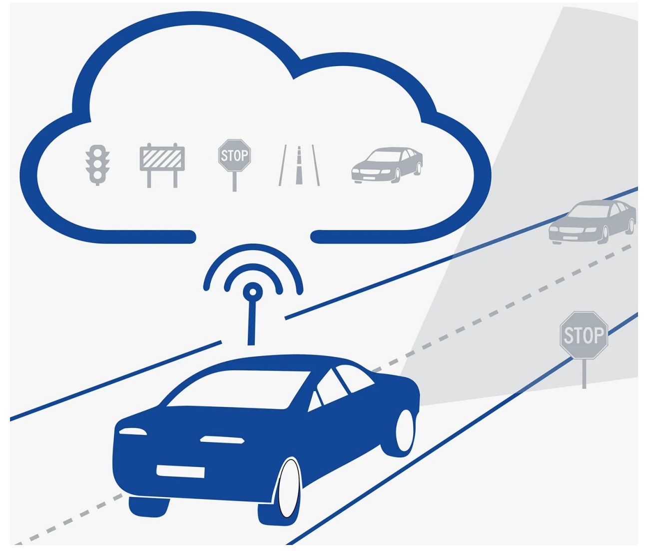


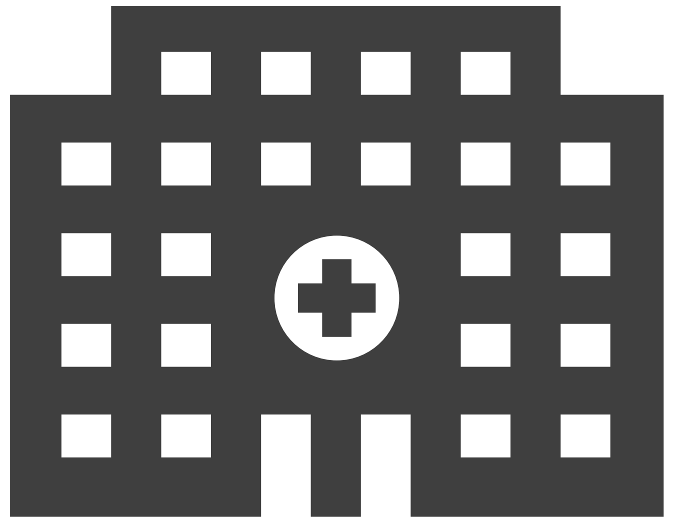
Privacy of user data

Federated Learning

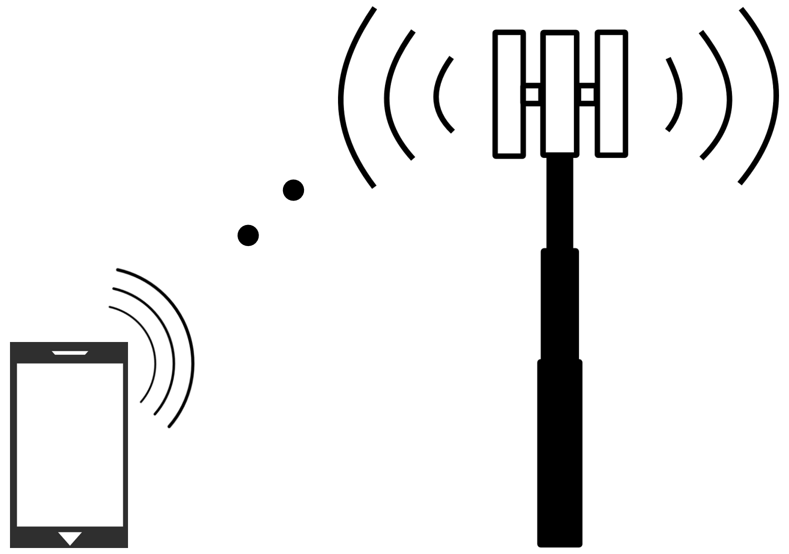
Machine learning has moved from the data centers to edge devices

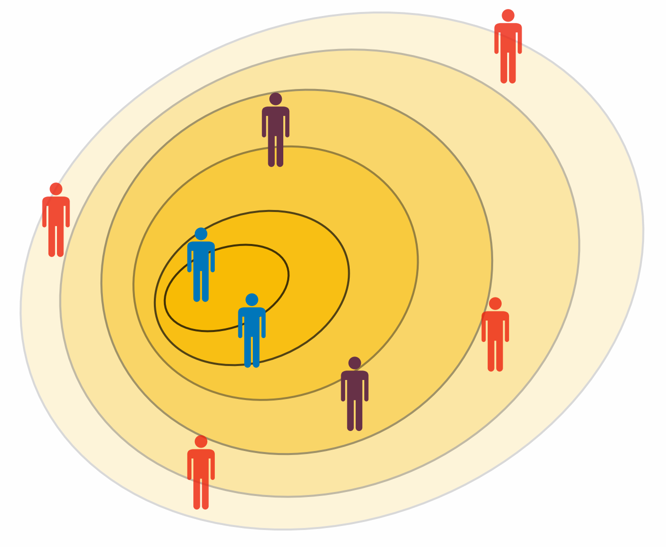





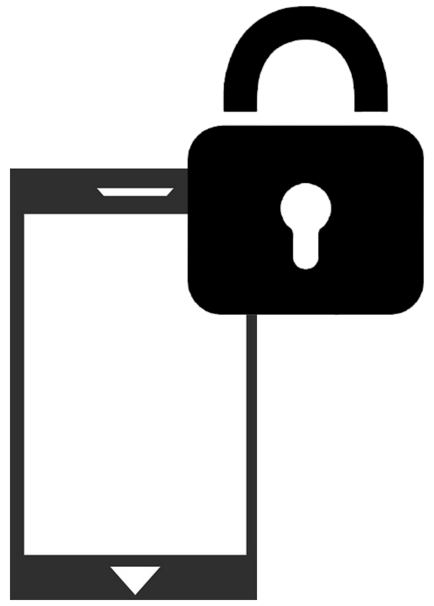
Challenges:



Communication efficiency



Statistical heterogeneity



Privacy of user data

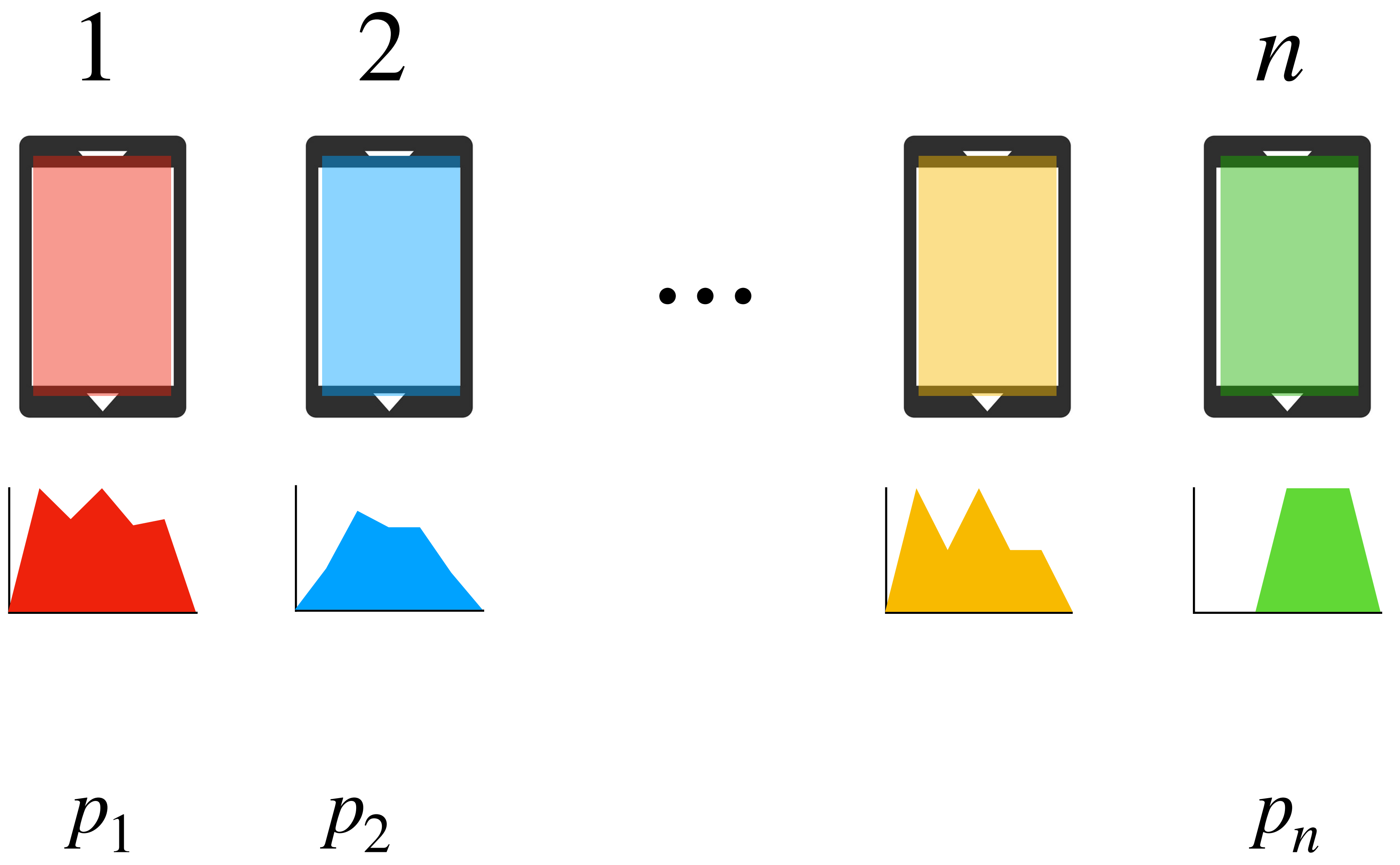
Outline

- **Background**
- Distributional Robustness with Simplicial-FL
- Algorithm & Convergence Guarantees
- Numerical Results

Usual Approach to Federated Learning

Clients

Data Distribution



Statistical heterogeneity

Objective

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} F_i(w) \quad \text{where} \quad F_i(w) = \mathbb{E}_{z \sim p_i}\left[f(w; z)\right]$$

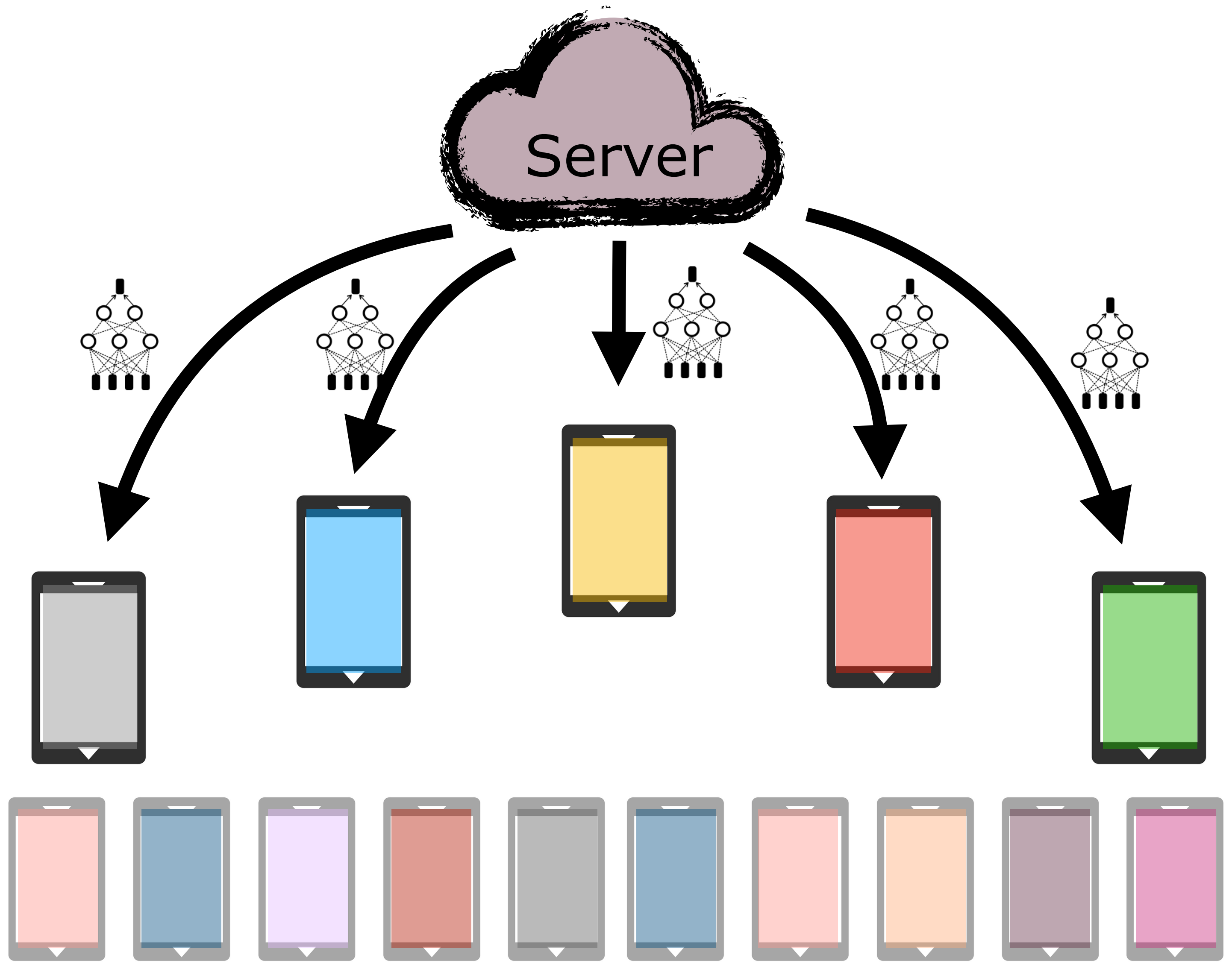
loss on client i

[McMahan et al. AISTATS (2017), Kairouz et al. (2021)]

Usual Approach to Federated Learning

The FedAvg Algorithm [McMahan et al. (2017)]:

Step 1 of 3: Server broadcasts global model to sampled clients



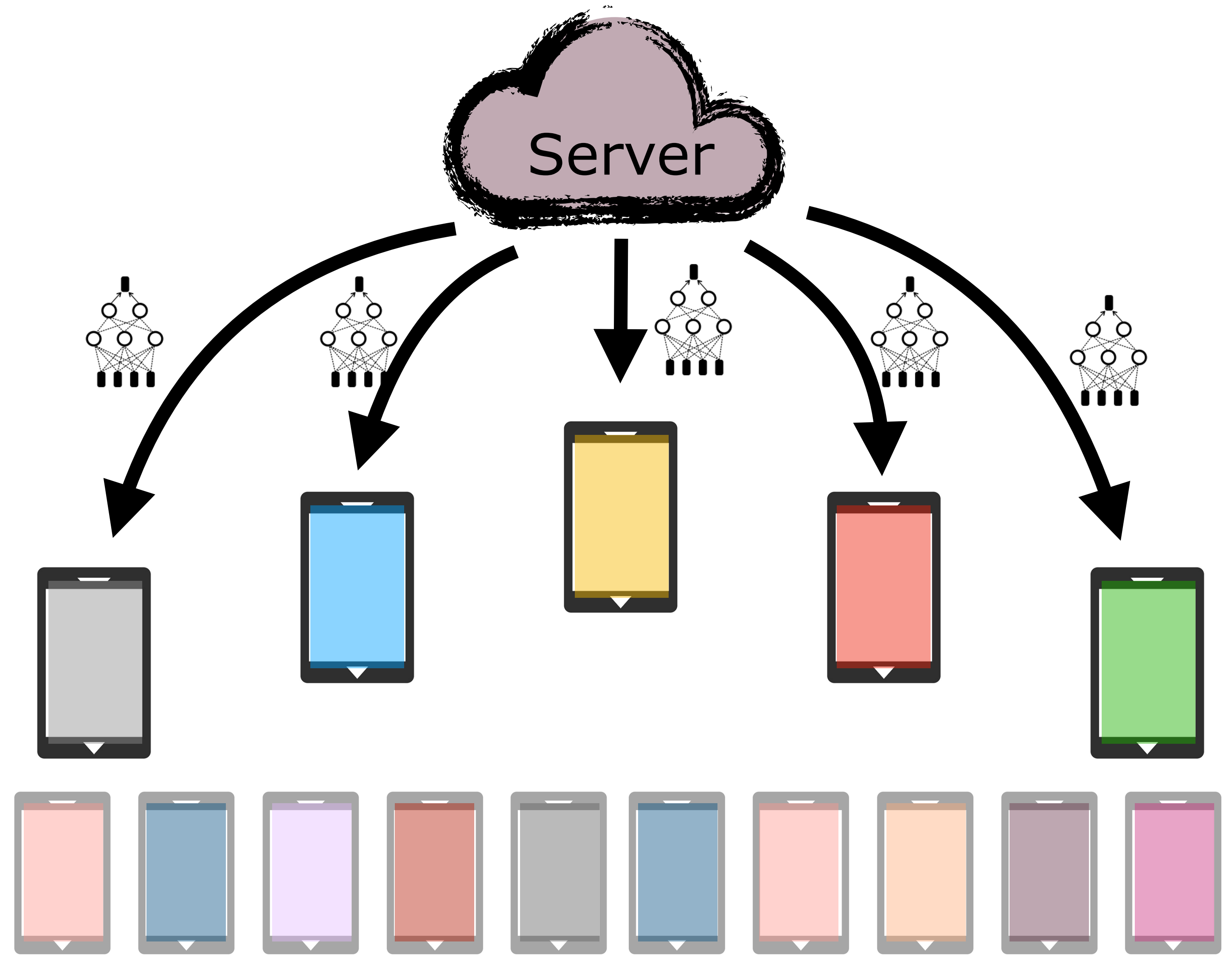
Parallel Gradient Distribution [Mangasarian. SICON (1995)]
Iterative Parameter Mixing [McDonald et al. ACL (2009)]

BMUF [Chen & Huo. ICASSP (2016)]
Local SGD [Stich. ICLR (2019)]

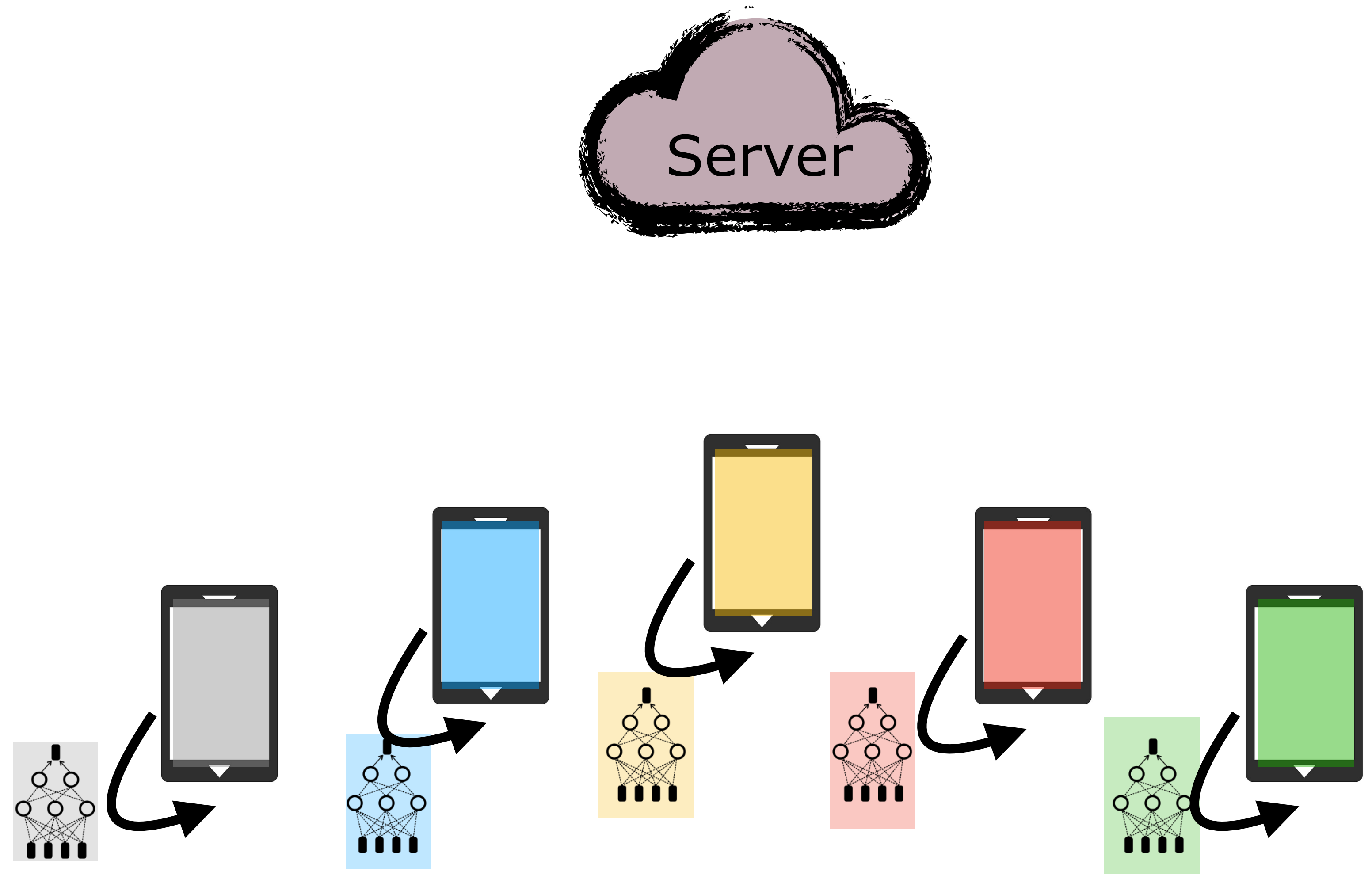
Usual Approach to Federated Learning

The FedAvg Algorithm [McMahan et al. (2017)]:

Step 1 of 3: Server broadcasts global model to sampled clients



Step 2 of 3: Clients perform some local SGD steps on their local data



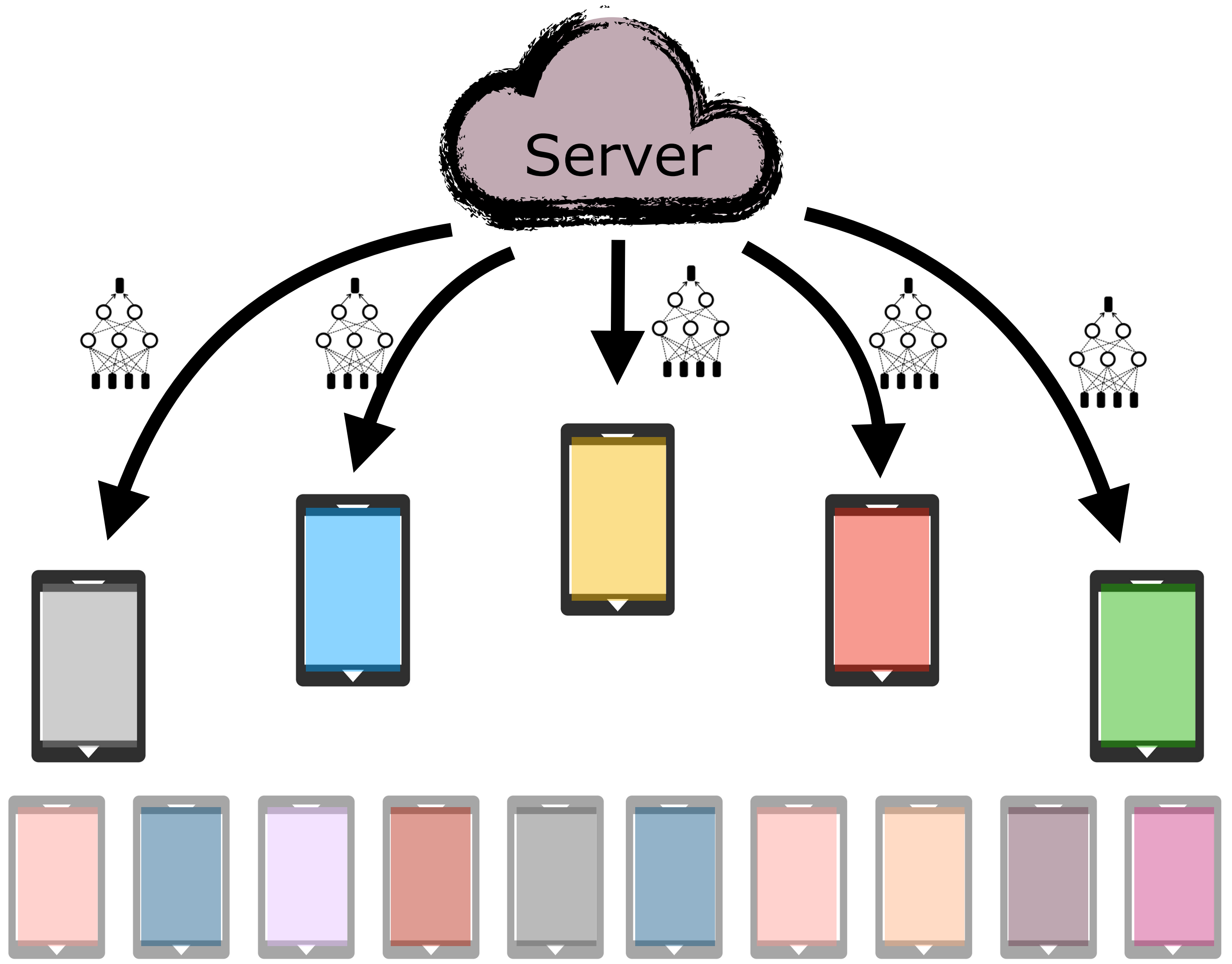
Parallel Gradient Distribution [Mangasarian. SICON (1995)]
Iterative Parameter Mixing [McDonald et al. ACL (2009)]

BMUF [Chen & Huo. ICASSP (2016)]
Local SGD [Stich. ICLR (2019)]

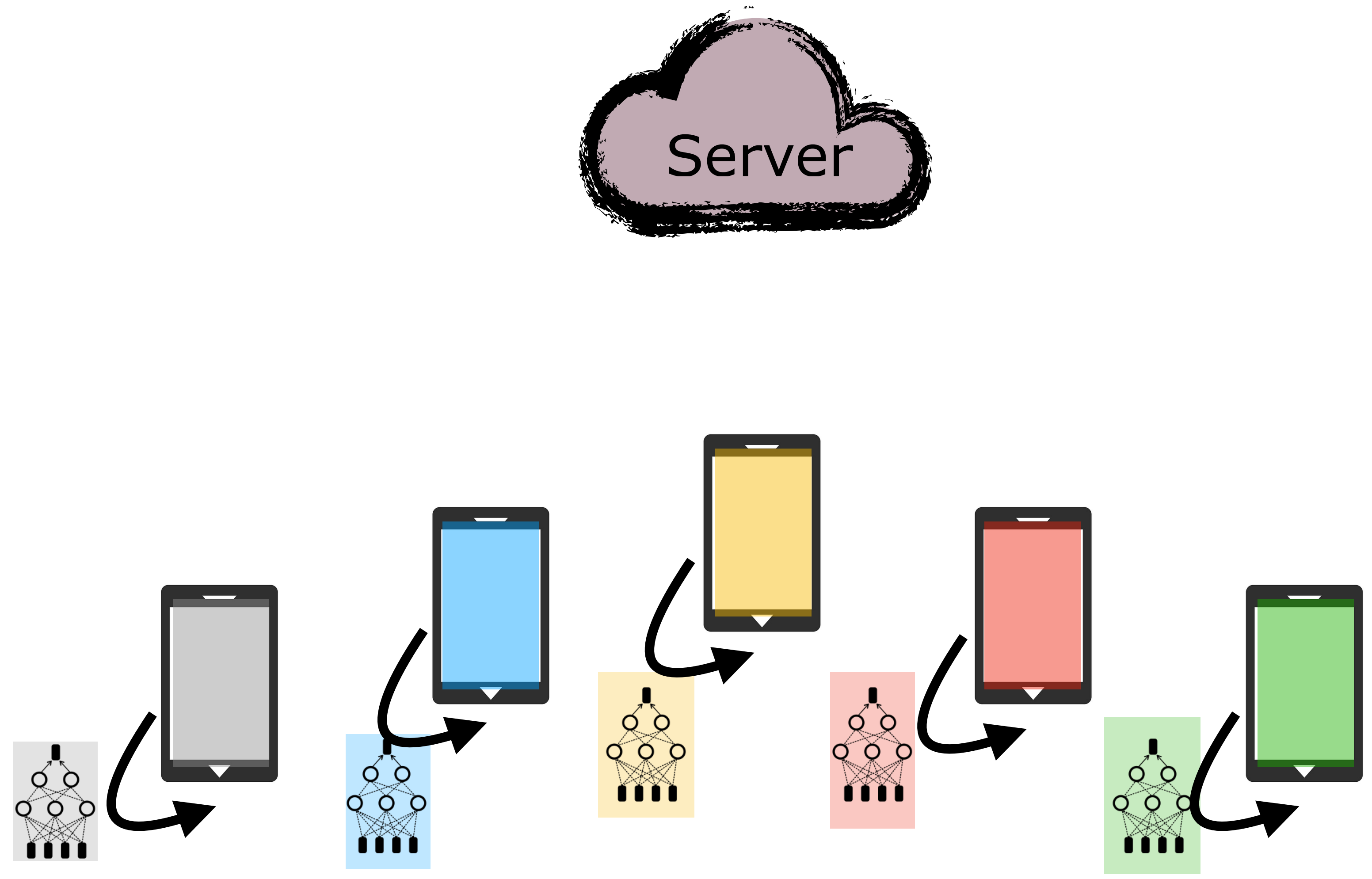
Usual Approach to Federated Learning

The FedAvg Algorithm [McMahan et al. (2017)]:

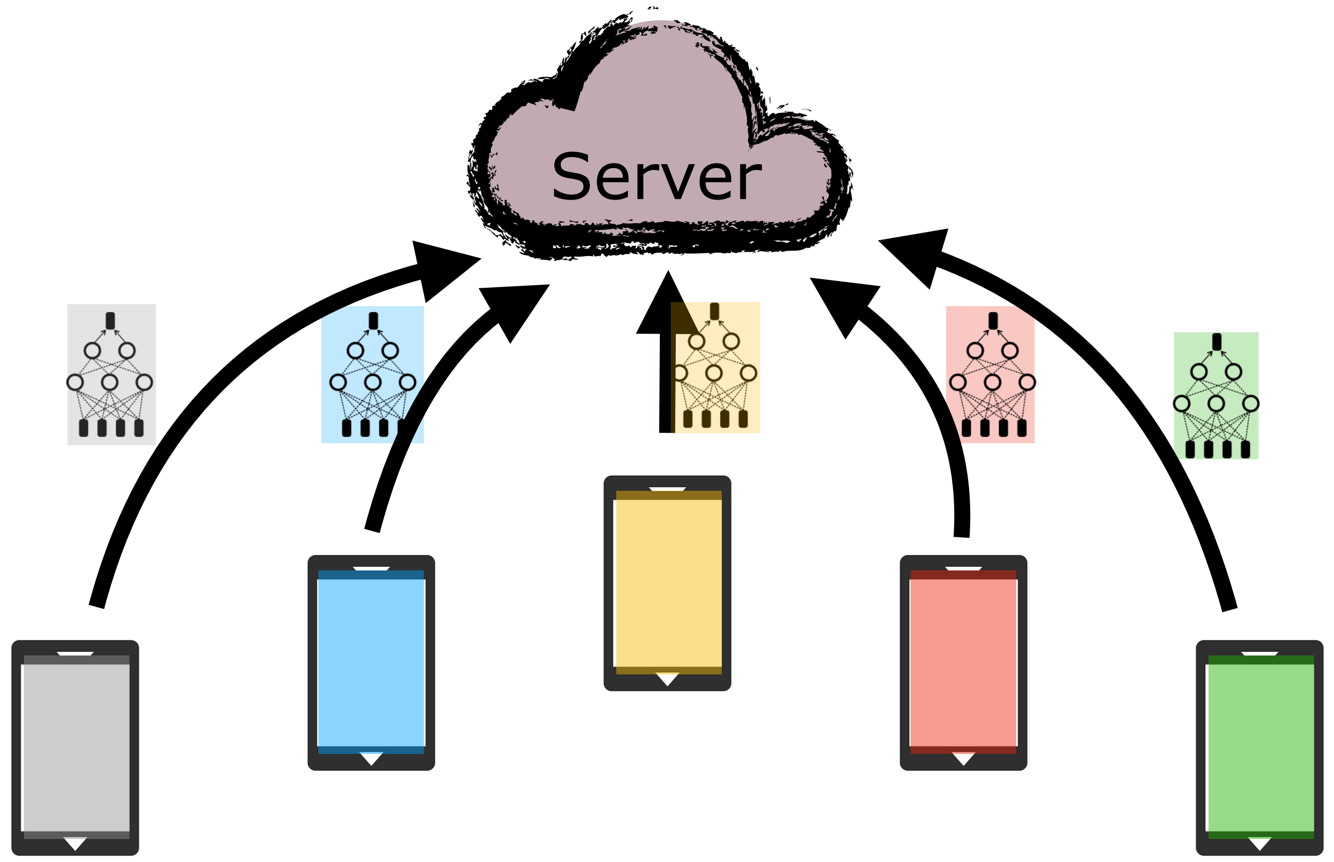
Step 1 of 3: Server broadcasts global model to sampled clients



Step 2 of 3: Clients perform some local SGD steps on their local data



Step 3 of 3: Aggregate client updates securely



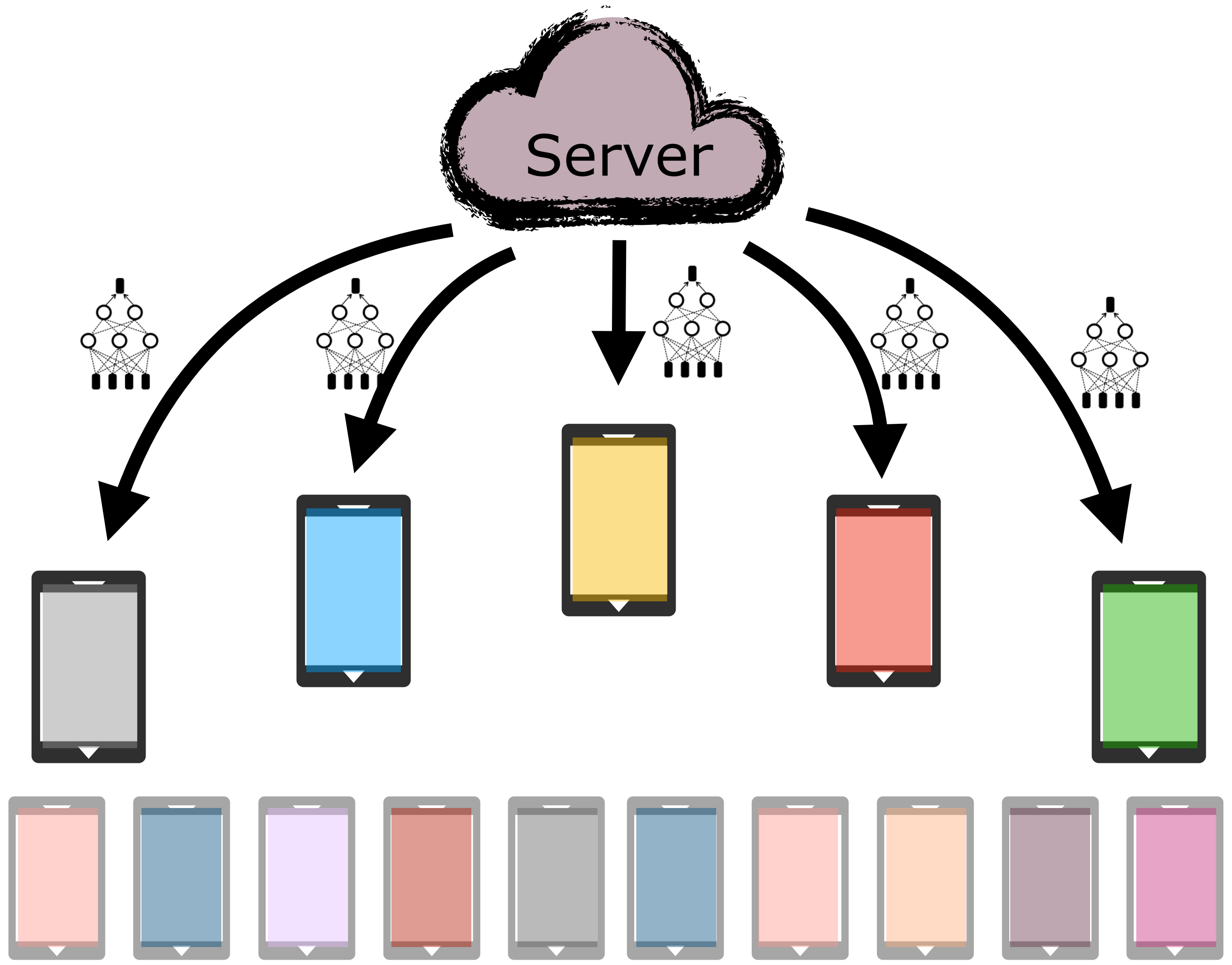
Parallel Gradient Distribution [Mangasarian. SICON (1995)]
Iterative Parameter Mixing [McDonald et al. ACL (2009)]

BMUF [Chen & Huo. ICASSP (2016)]
Local SGD [Stich. ICLR (2019)]

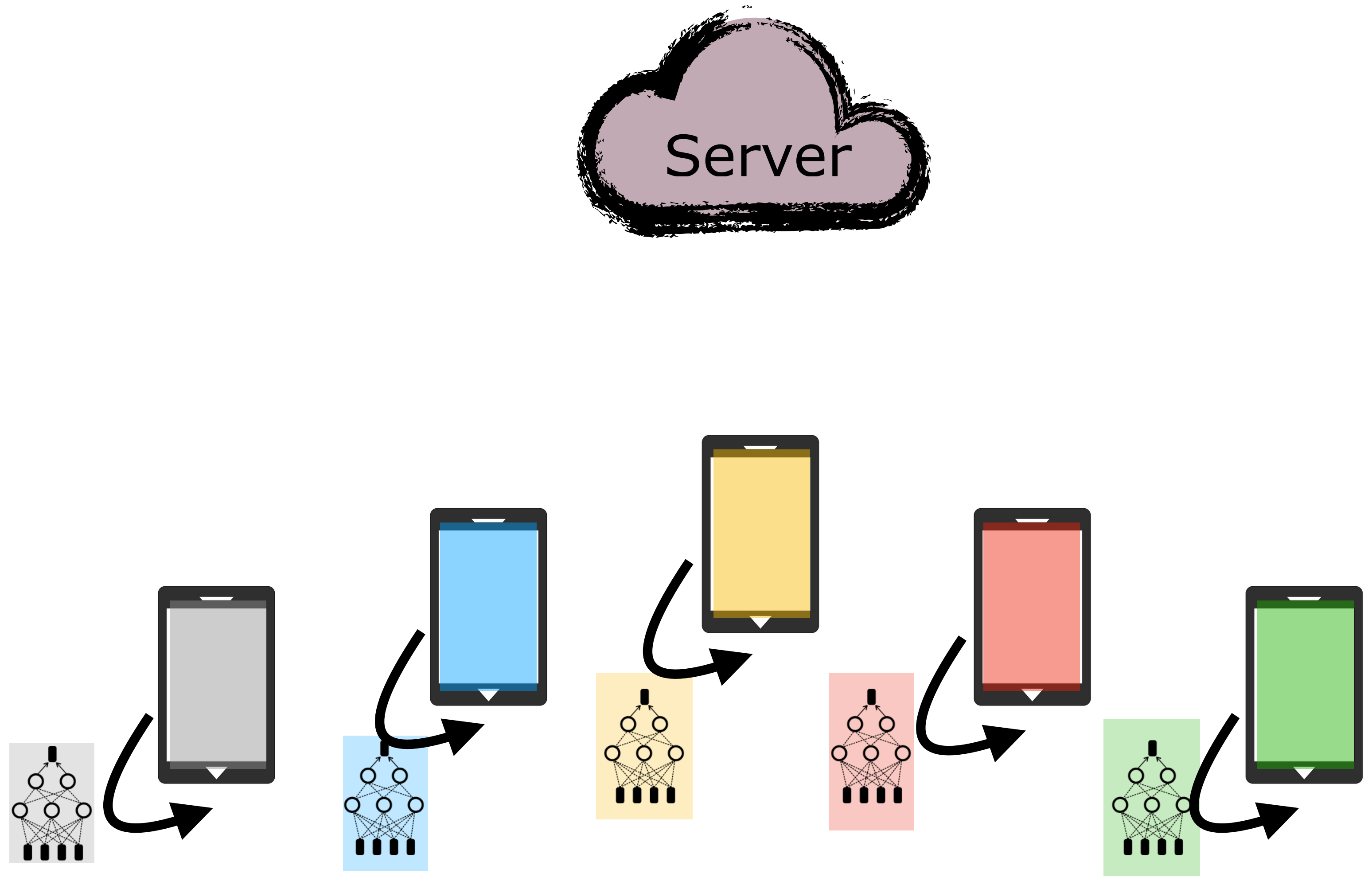
Usual Approach to Federated Learning

The FedAvg Algorithm [McMahan et al. (2017)]:

Step 1 of 3: Server broadcasts global model to sampled clients

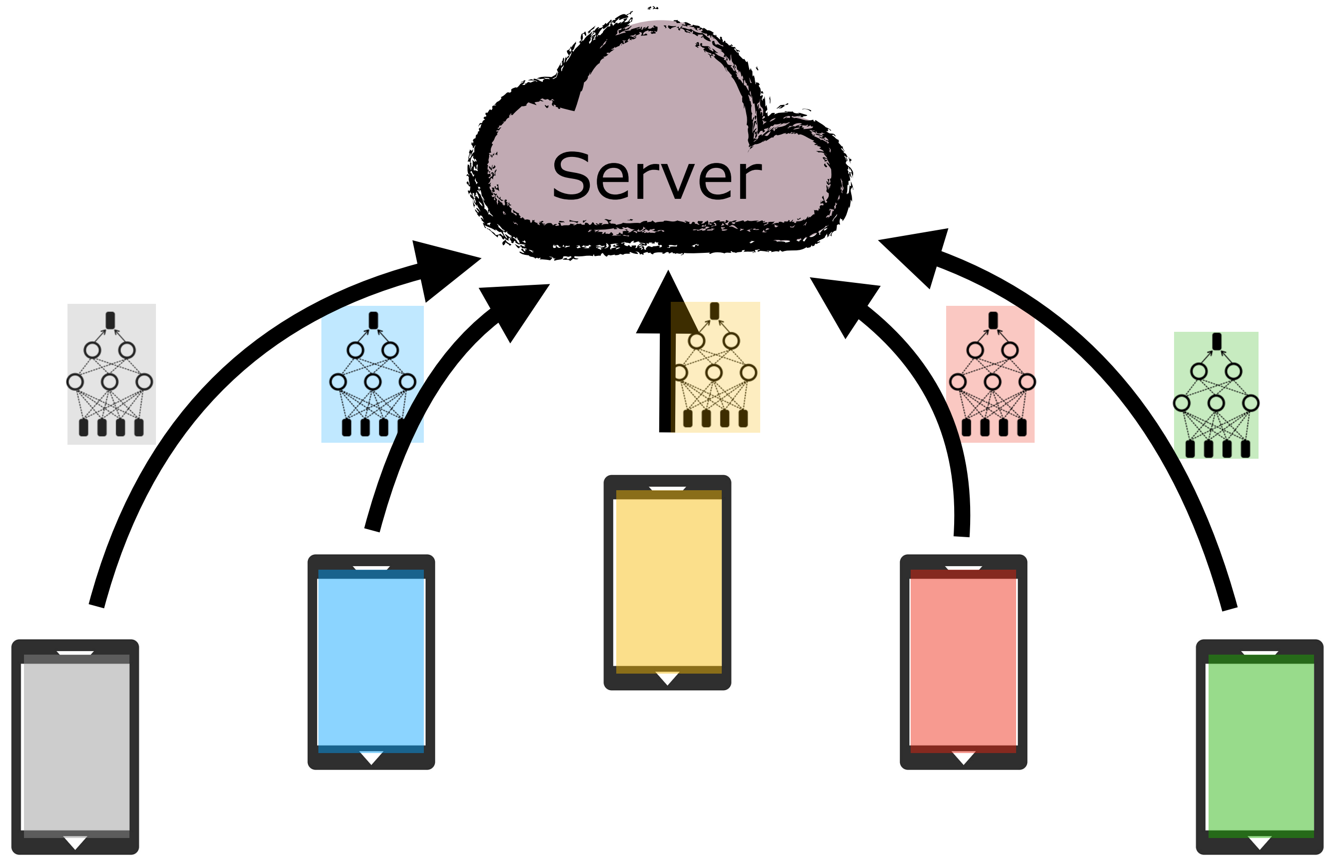


Step 2 of 3: Clients perform some local SGD steps on their local data



Communication | Privacy

Step 3 of 3: Aggregate client updates securely

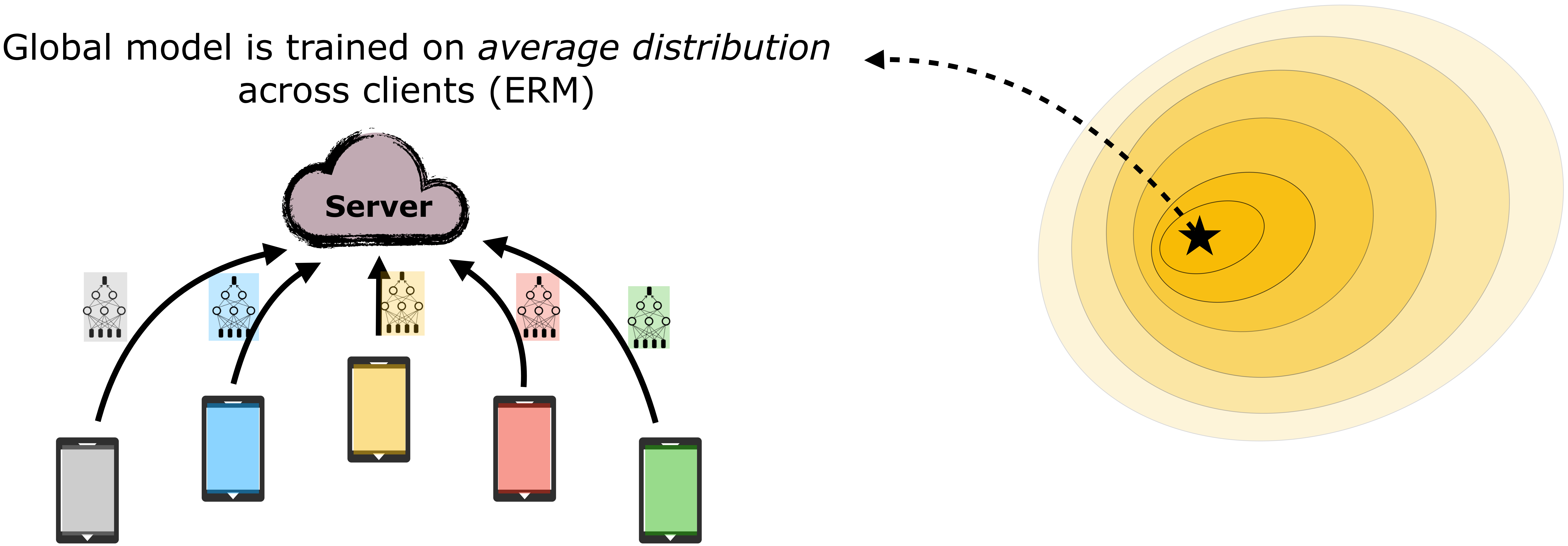


Parallel Gradient Distribution [Mangasarian. SICON (1995)]
Iterative Parameter Mixing [McDonald et al. ACL (2009)]

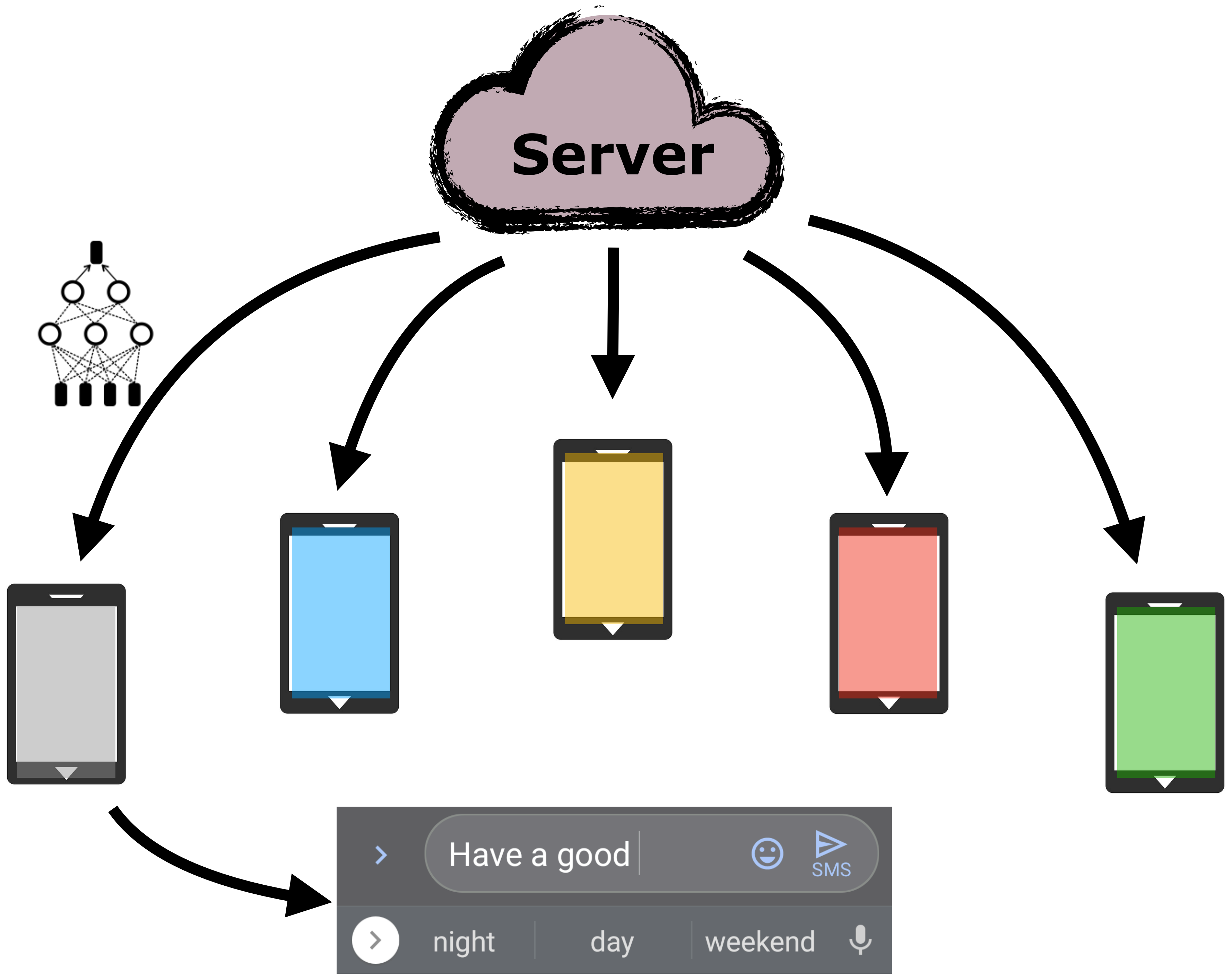
BMUF [Chen & Huo. ICASSP (2016)]
Local SGD [Stich. ICLR (2019)]

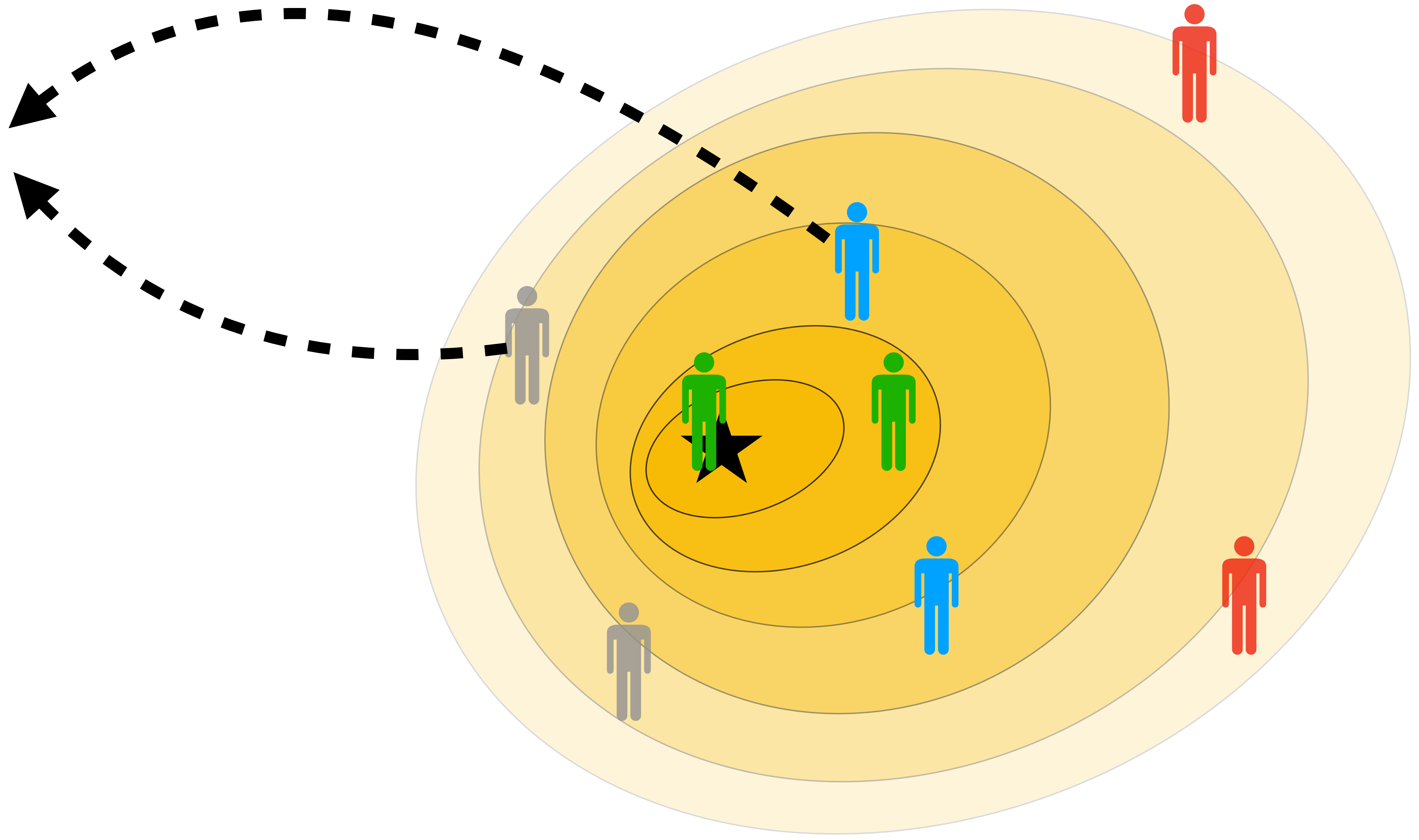
Outline

- Background
- **Distributional Robustness with Simplicial-FL**
- Algorithm & Convergence Guarantees
- Numerical Results

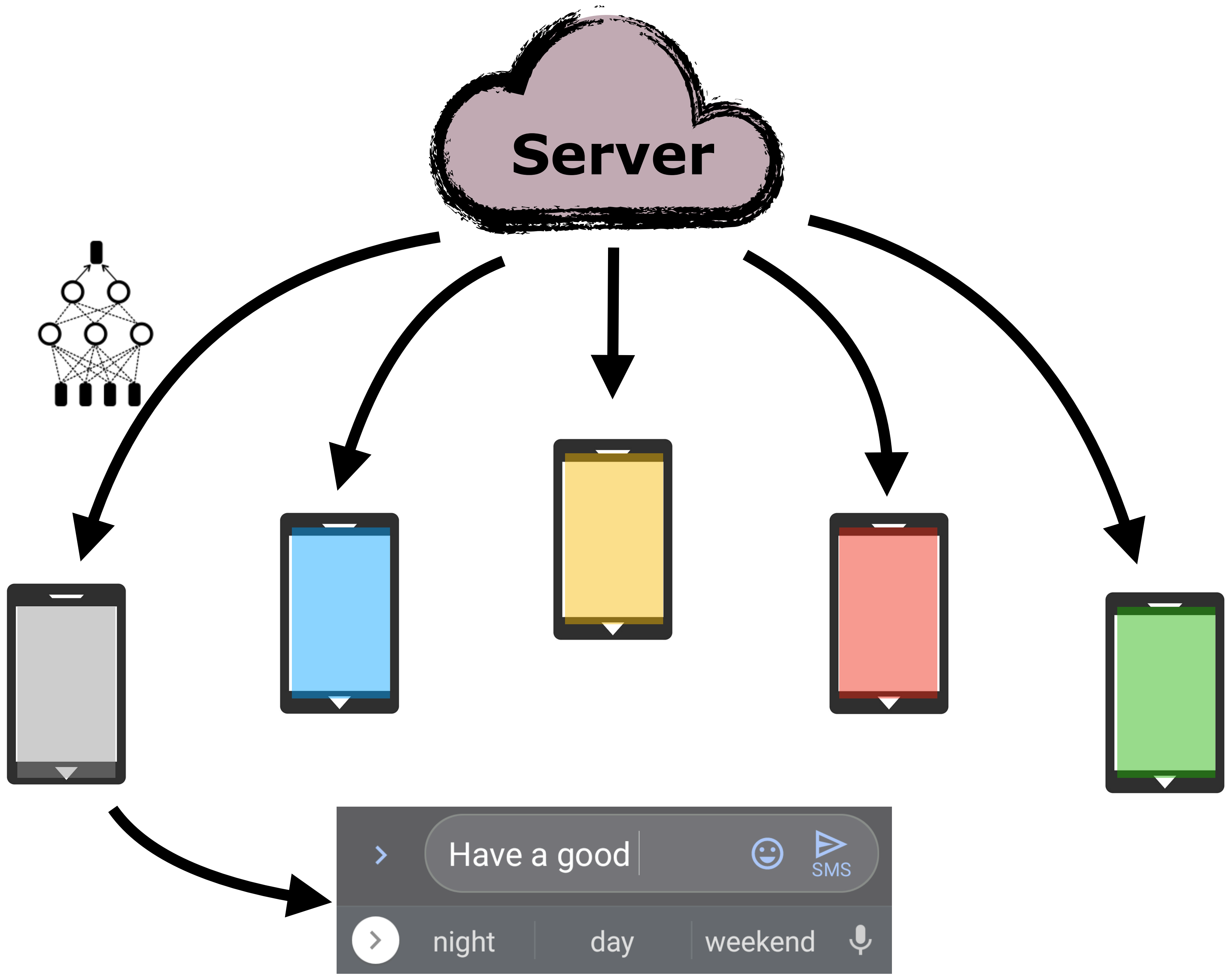

Global model is trained on *average distribution* across clients (ERM)
Server

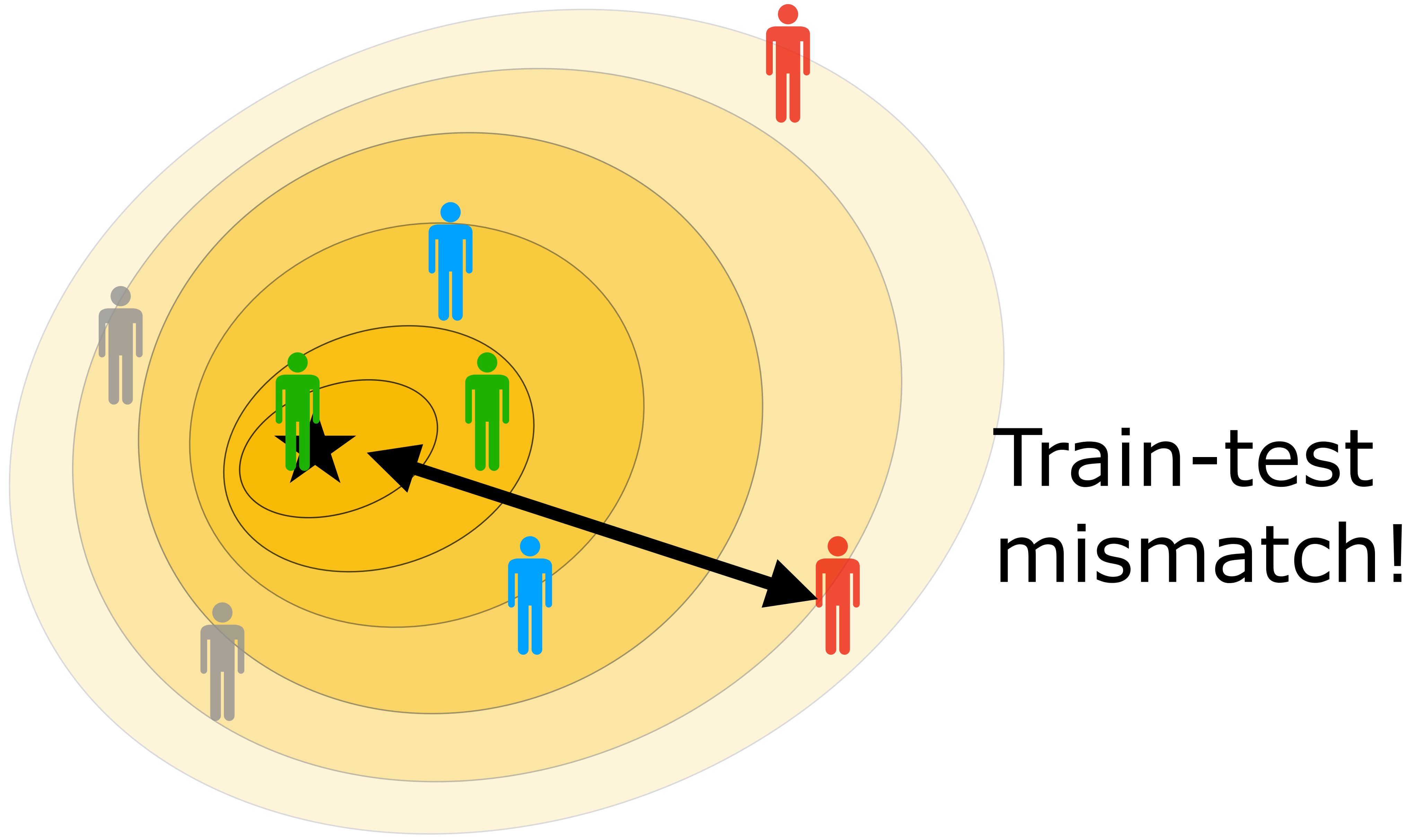
Global model is deployed on *individual* clients



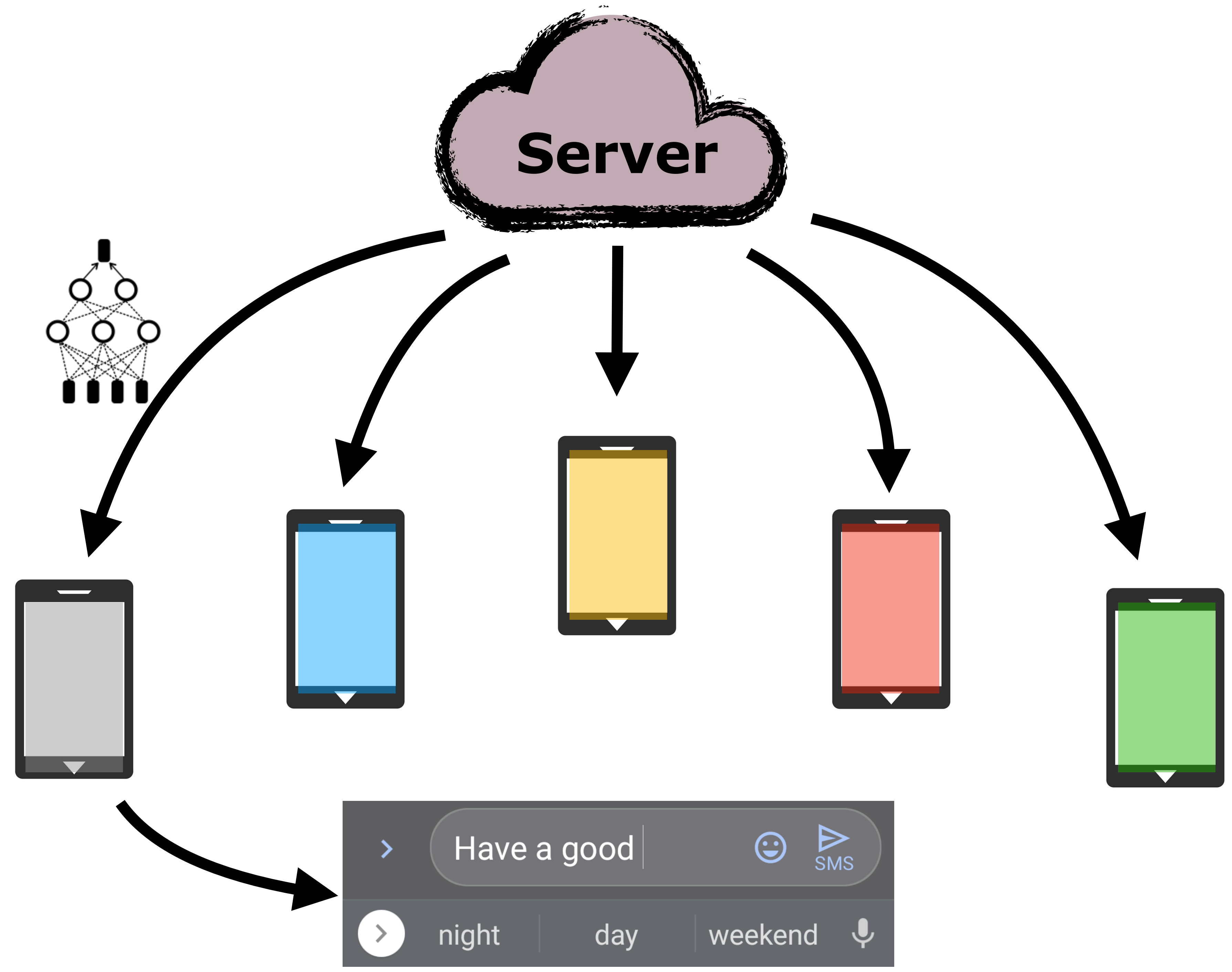


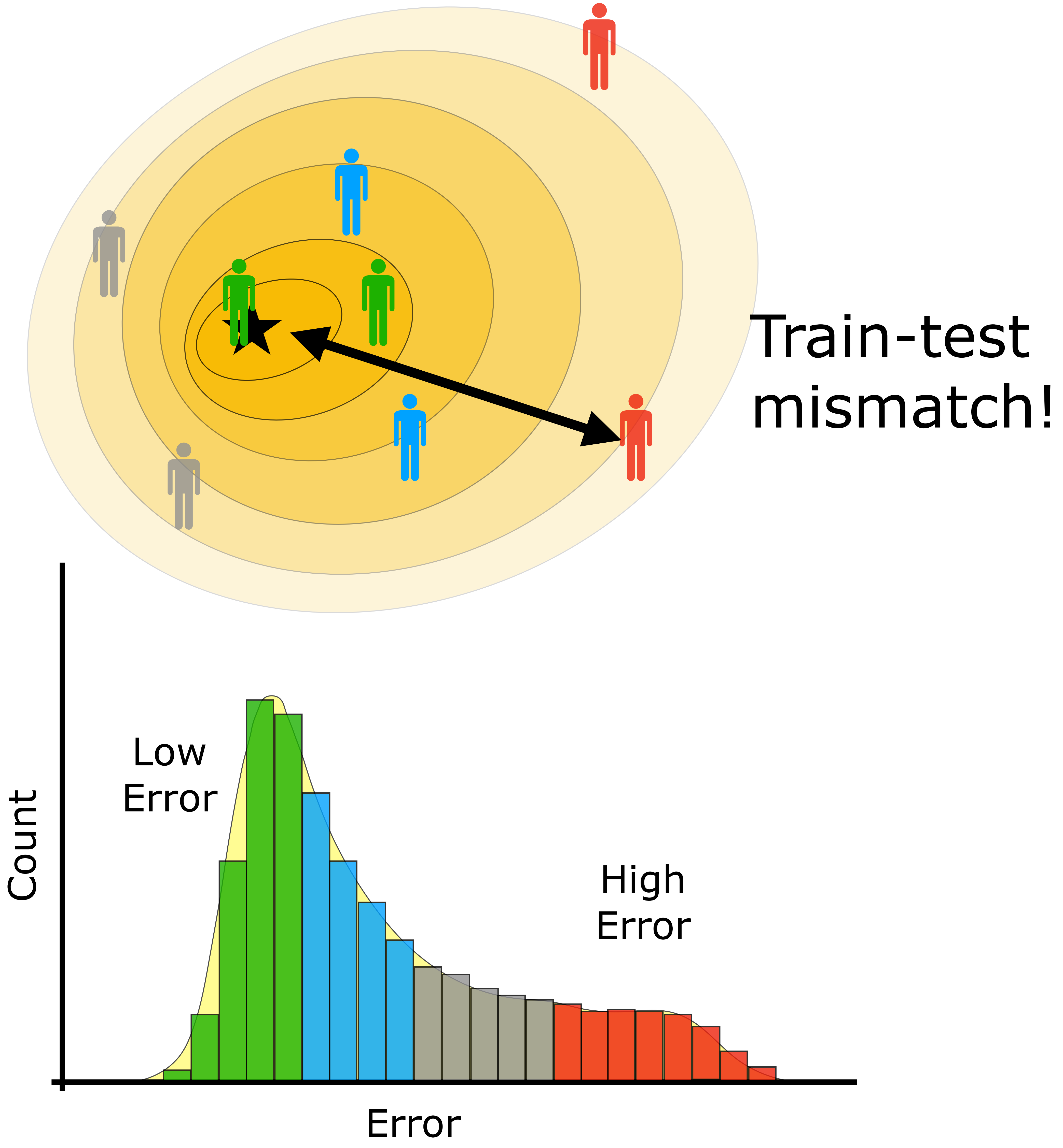
Global model is deployed on *individual* clients





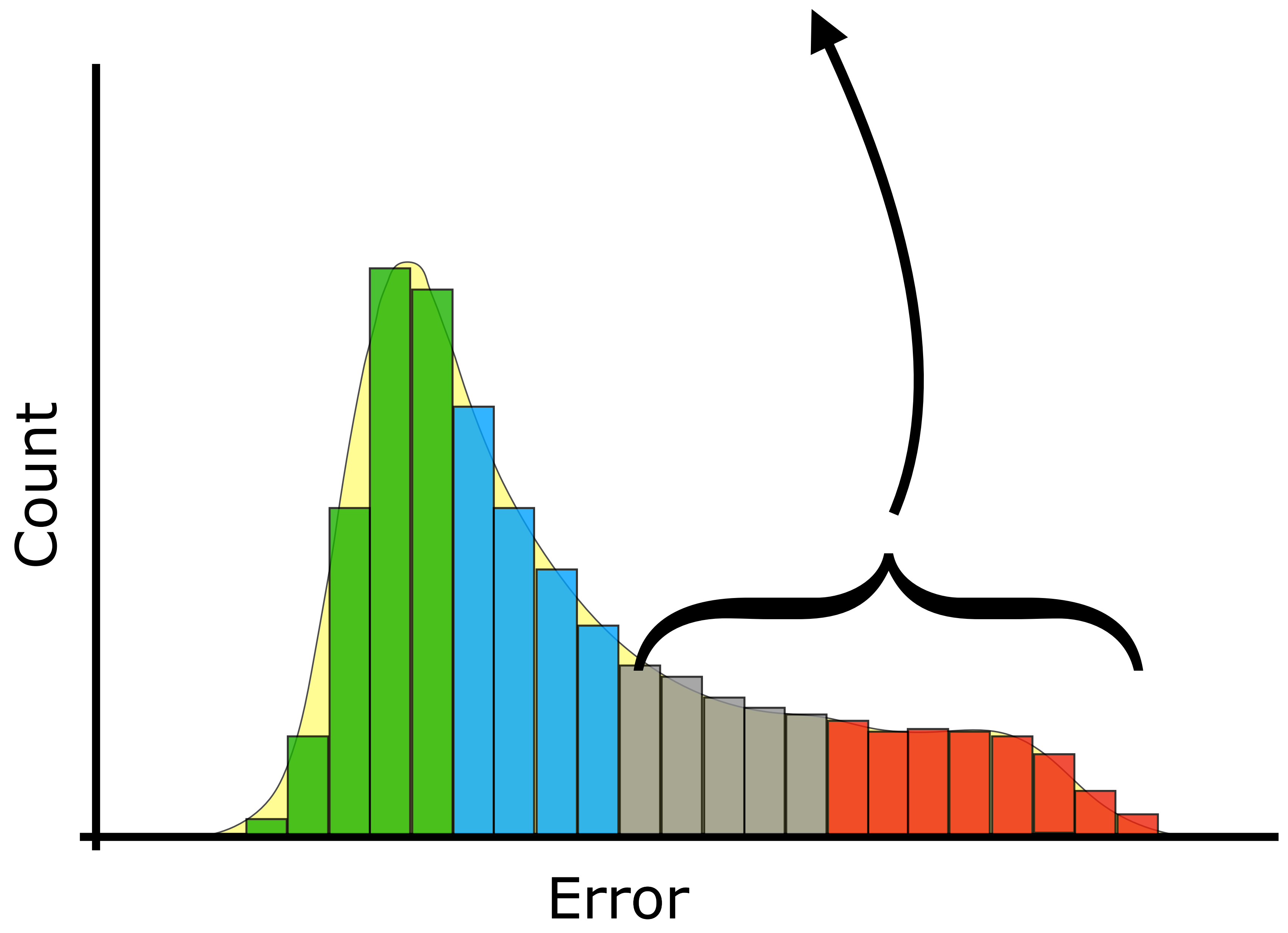
Global model is deployed on *individual* clients





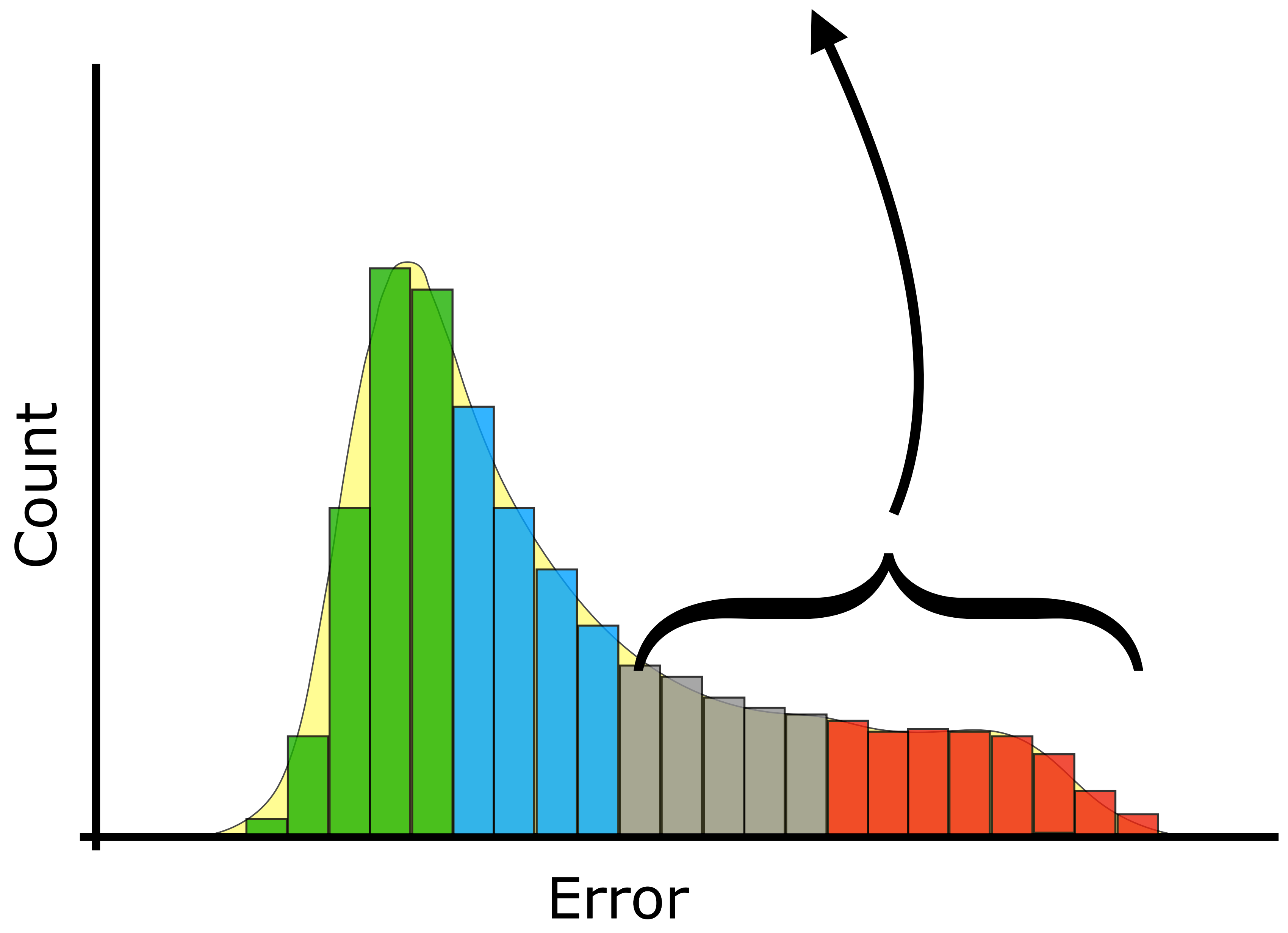
Simplicial-FL

Our Approach: minimize the tail error directly!



Simplicial-FL

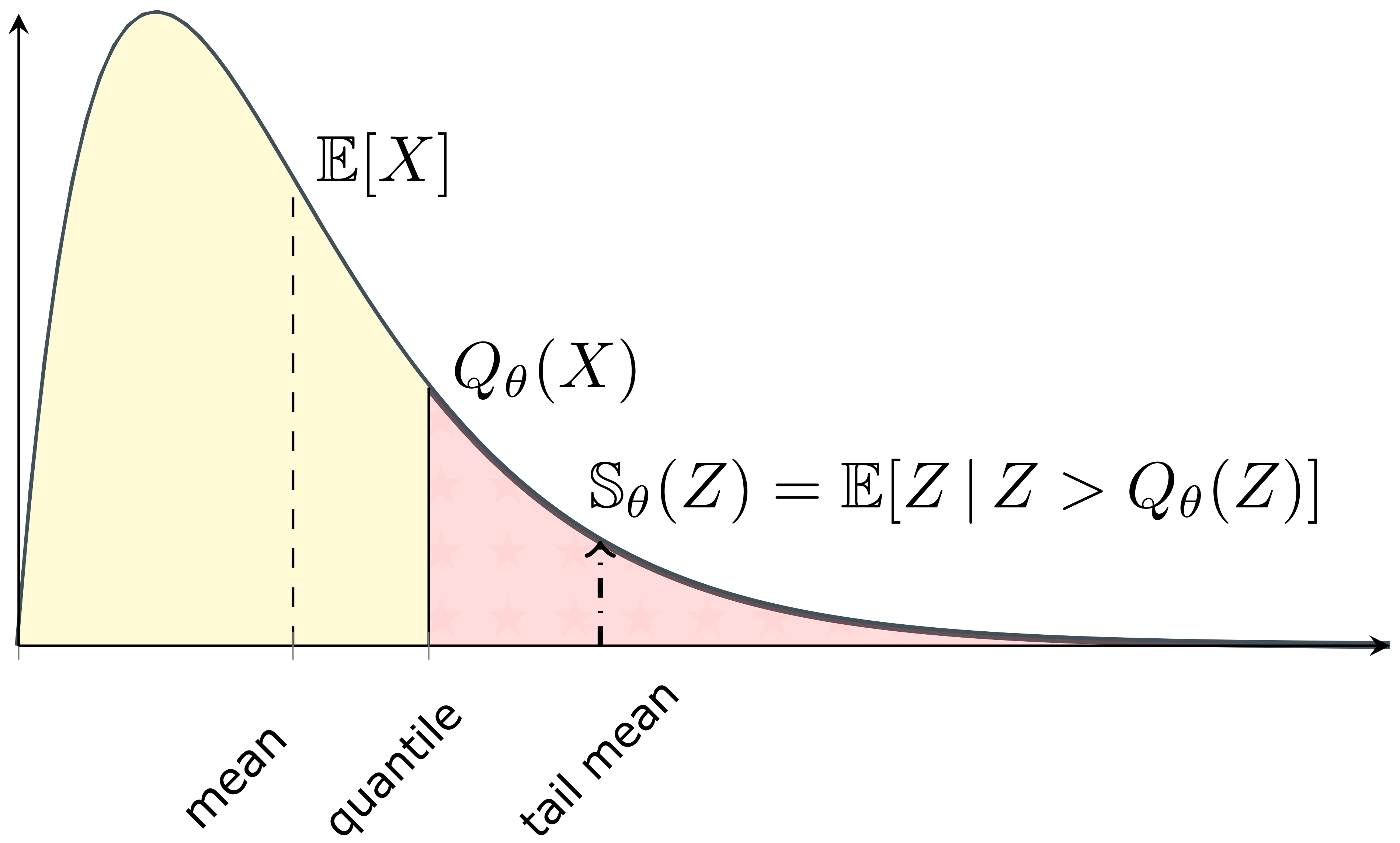
Our Approach: minimize the tail error directly!



Simplicial-FL Objective:

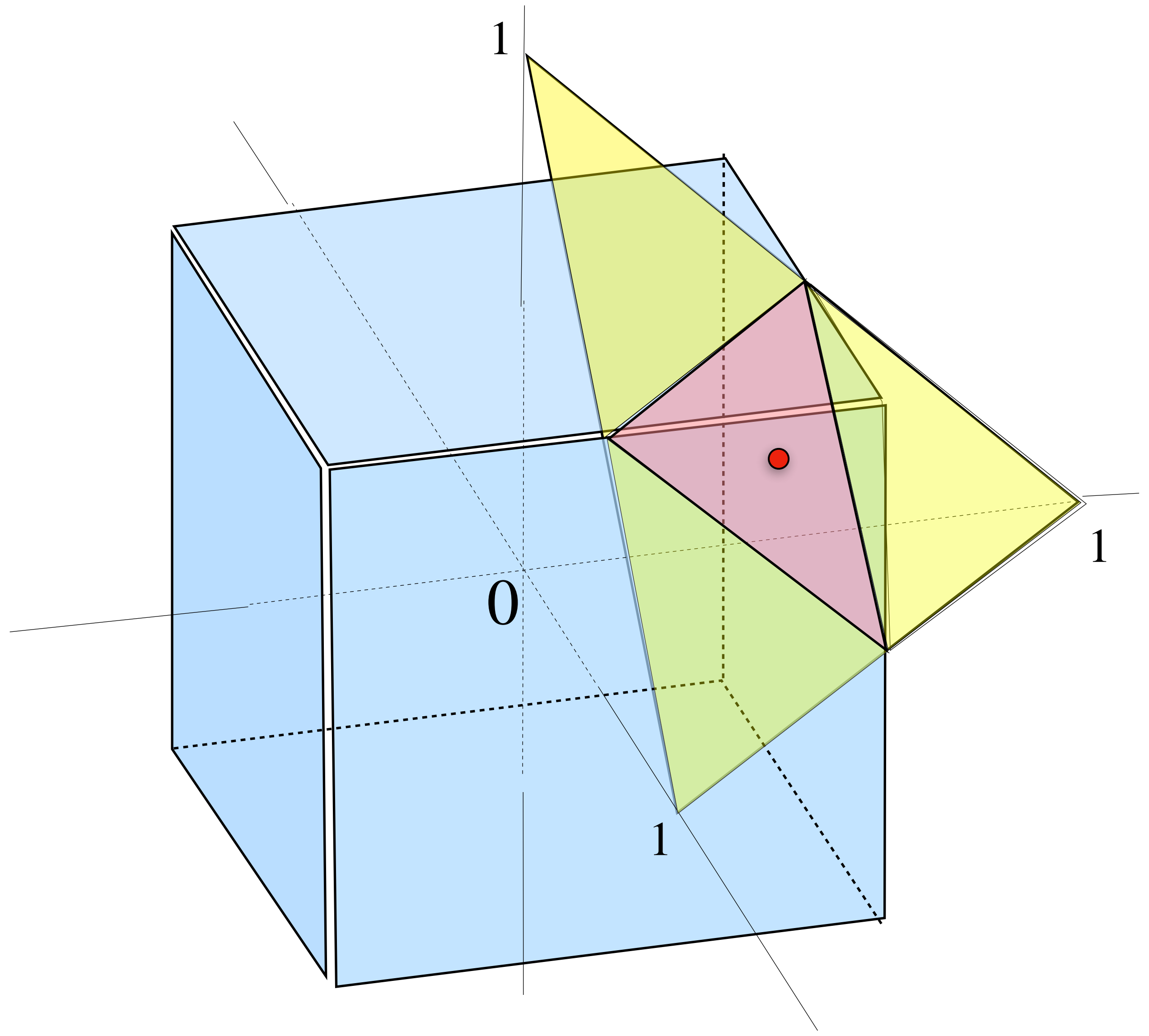
$$\min_{w} \mathbb{S}_\theta\Big(\big(F_1(w), \cdots, F_n(w)\big) \Big)$$

Superquantile | Conditional Value at Risk



[Rockafellar & Uryasev (2002)]

Distributional robustness



Dual expression [Ben-Tal & Teboulle (1987), Föllmer & Schied (2002)]

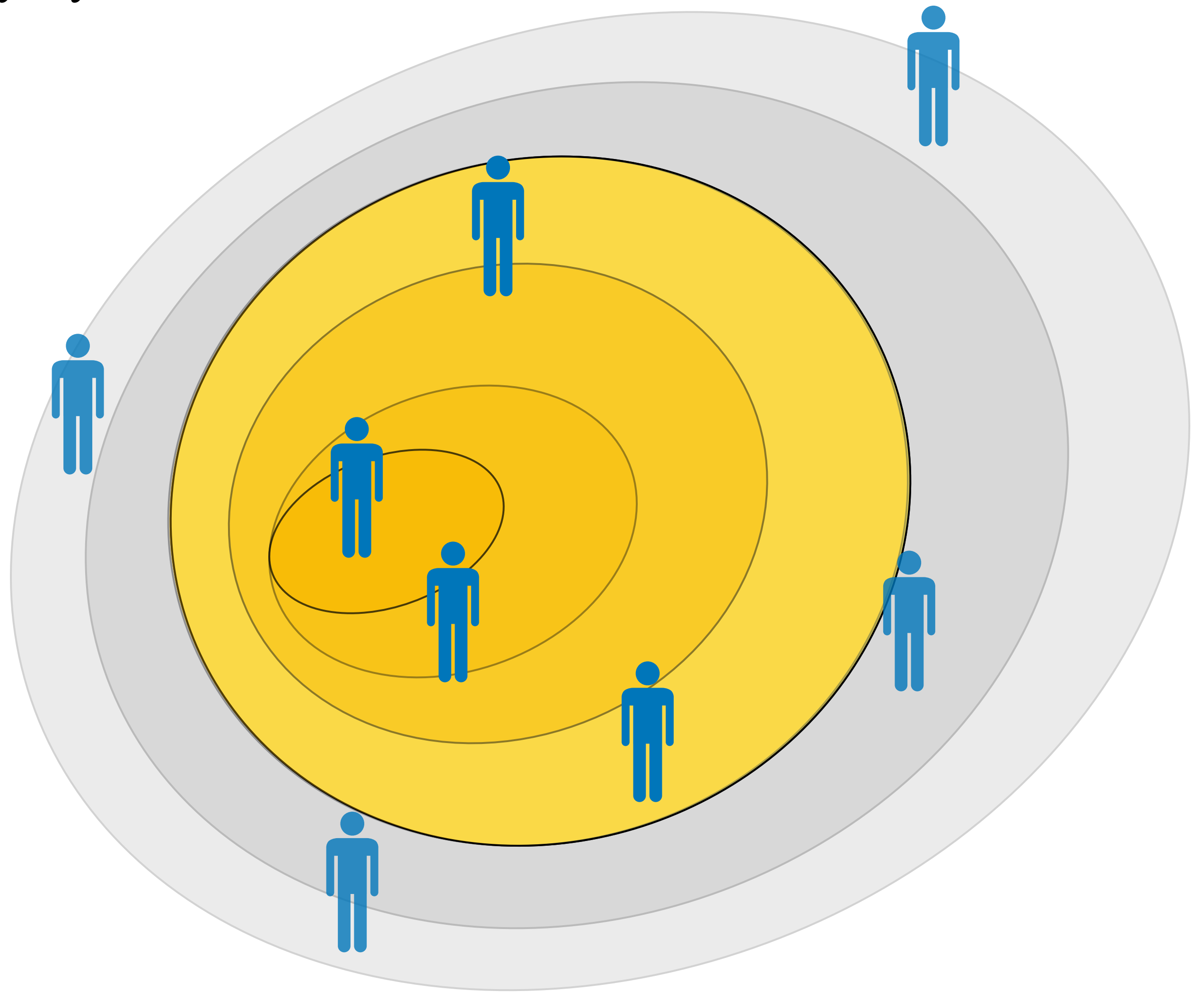
$$\mathbb{S}_\theta(x_1, \cdots, x_n) = \max \left\{ \sum_i \pi_i x_i \; : \; \pi_i \geq 0, \; \sum_i \pi_1 = 1, \; \pi_i \leq p_i/\theta \right\}$$

Assuming a new test client with mixture distribution $p_\pi = \sum_i \pi_i p_i$,

Simplicial-FL objective is equivalent to:

$$\min_{w} \max_{\pi : \pi_i \leq (n\theta_i)^{-1}} \mathbb{E}_{z \sim p_\pi} \left[f(w; z) \right]$$

Worst-case over a family of distributions



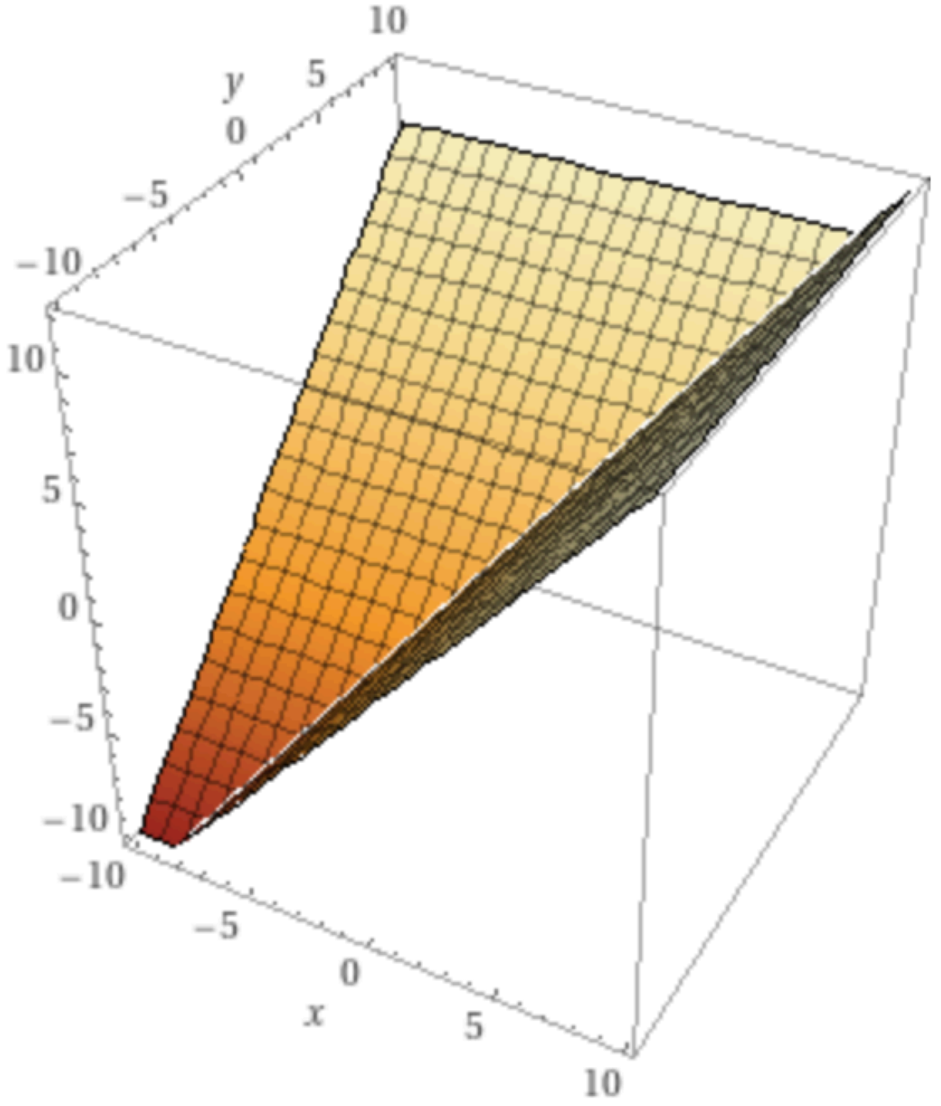
Outline

- Background
- Distributional Robustness with Simplicial-FL
- **Algorithm & Convergence Guarantees**
- Numerical Results

Optimization

Simplicial-FL Objective:

$$F_\theta(w) = \mathbb{S}_\theta\Big(\big(F_1(w), \cdots, F_n(w)\big) \Big)$$



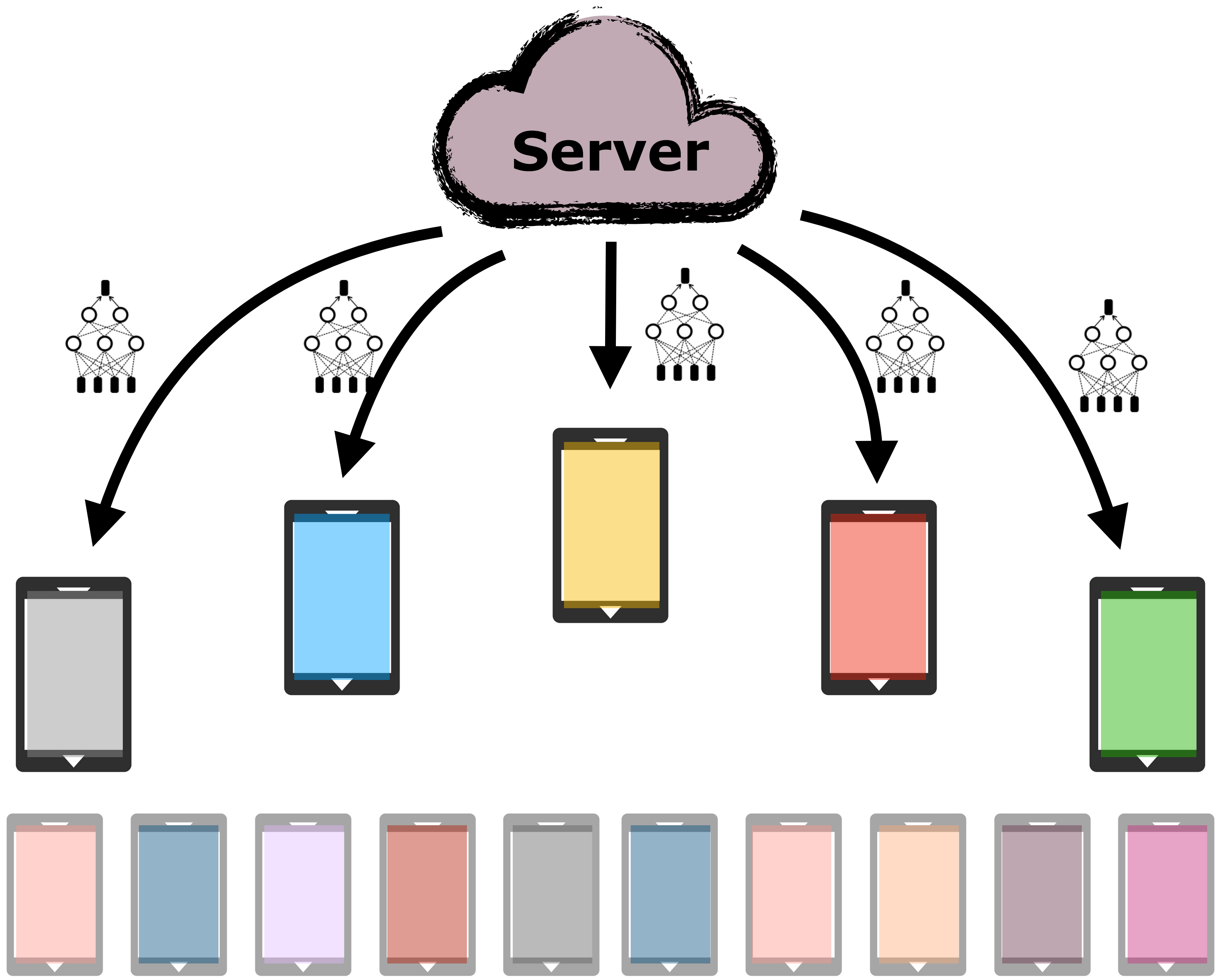
Challenges:

- Superquantile is nonsmooth
- Superquantile is nonlinear (unbiased stochastic gradients not possible)

ERM Algorithm (FedAvg):

$$\min_{w} \quad \frac{1}{n}\sum_{i=1}^{n} F_i(w)$$

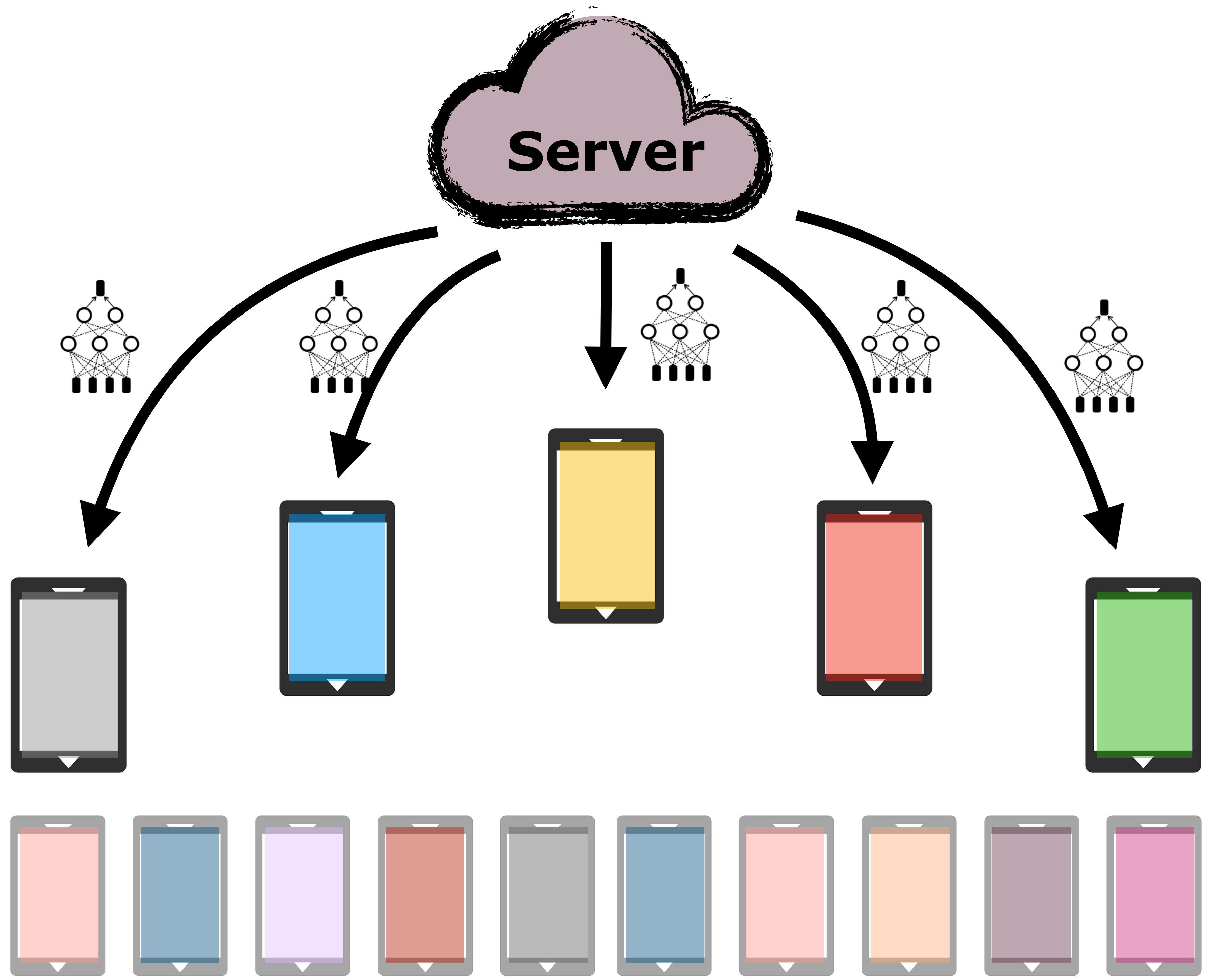
Step 1 of 3: Server samples m clients and broadcasts global model



Simplicial-FL Algorithm:

$$\min_{w} \mathbb{S}_\theta\Big(\big(F_1(w), \cdots, F_n(w)\big) \Big)$$

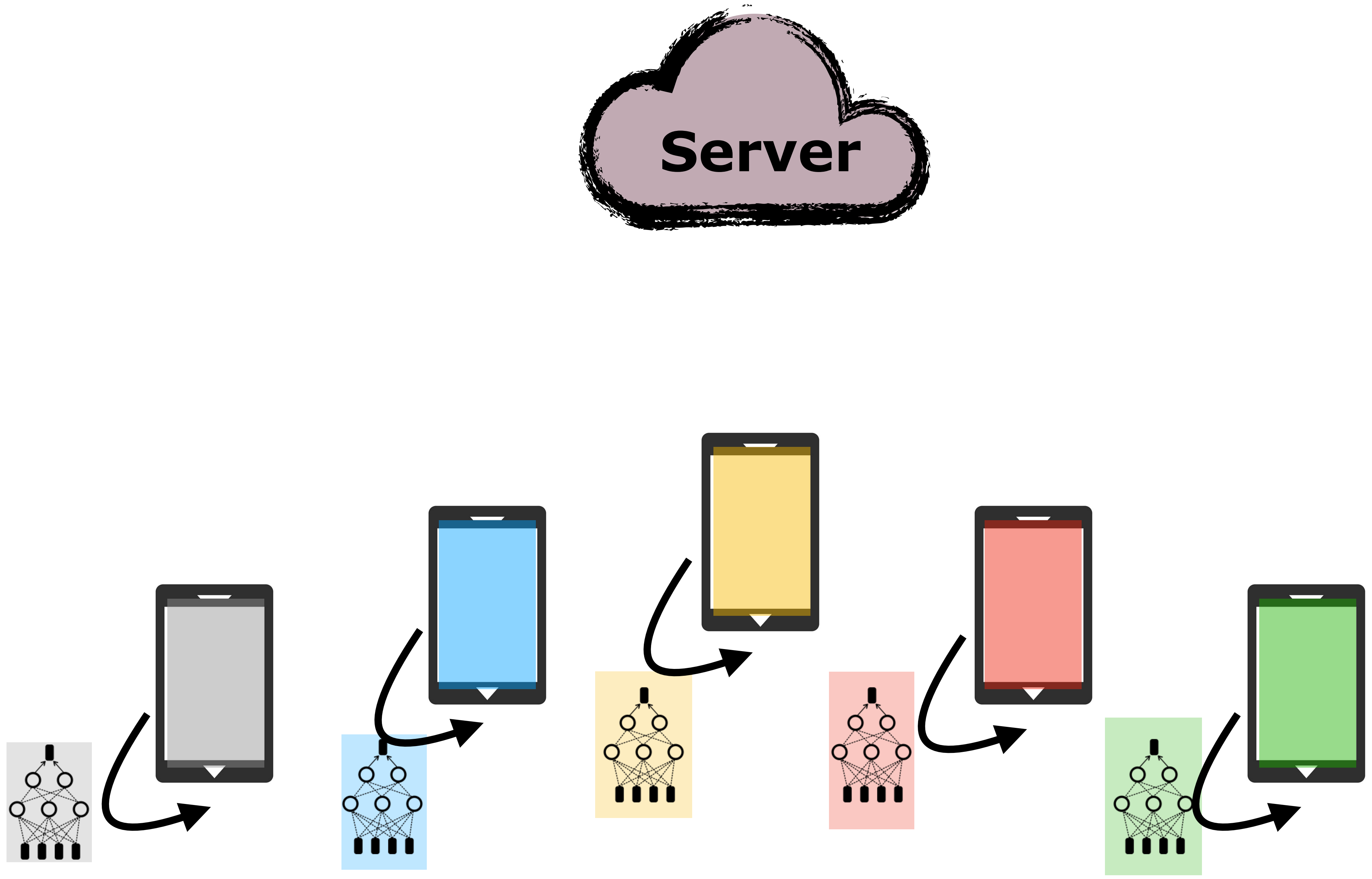
Step 1 of 3: Server samples m clients and broadcasts global model



ERM Algorithm (FedAvg):

$$\min_{w} \quad \frac{1}{n}\sum_{i=1}^{n} F_i(w)$$

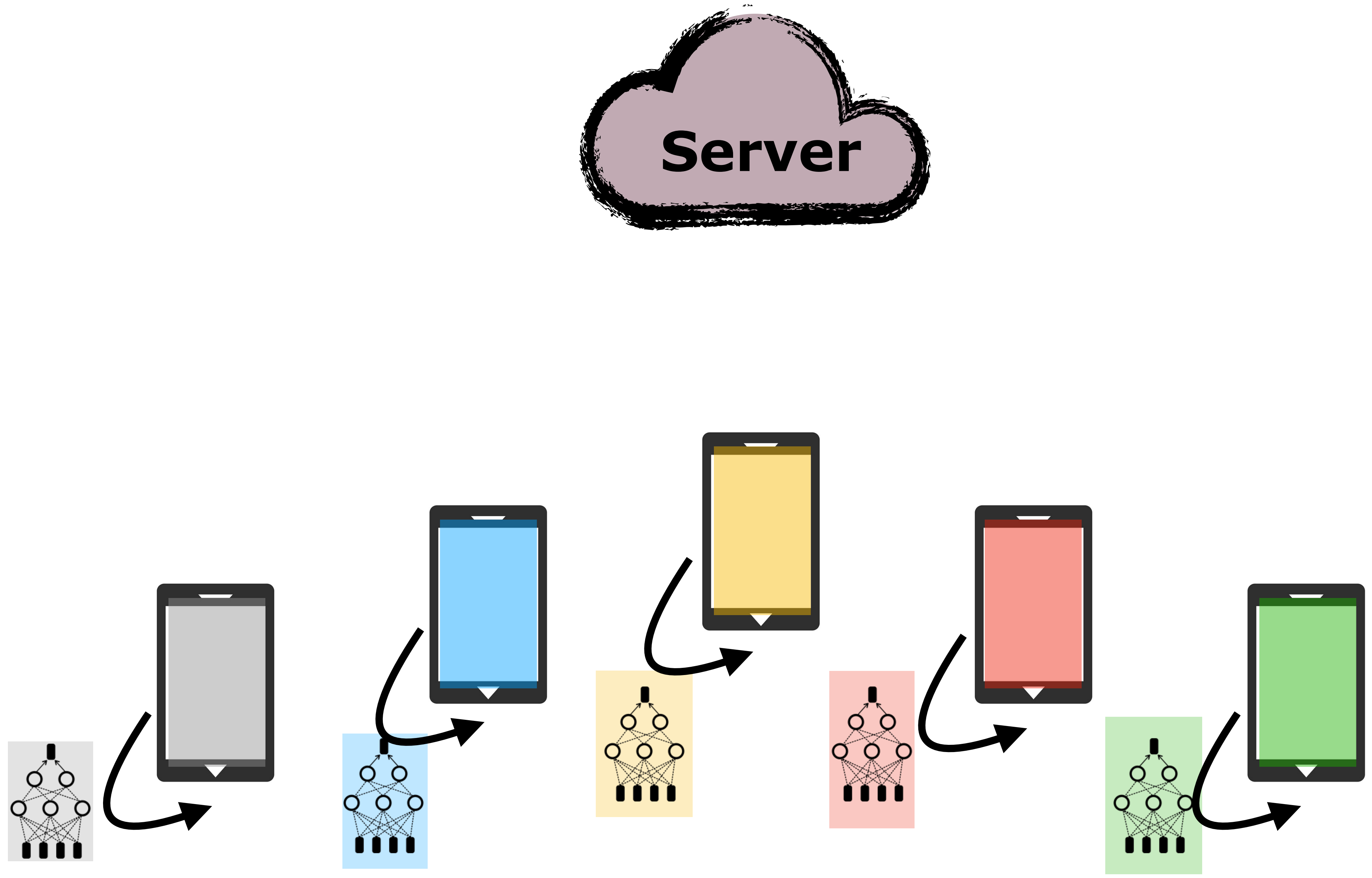
Step 2 of 3: Clients perform τ local SGD steps on their local data



Simplicial-FL Algorithm:

$$\min_{w} \mathbb{S}_{\theta}\Big(\big(F_1(w), \cdots, F_n(w)\big) \Big)$$

Step 2 of 3: Clients perform τ local SGD steps on their local data



ERM Algorithm (FedAvg):

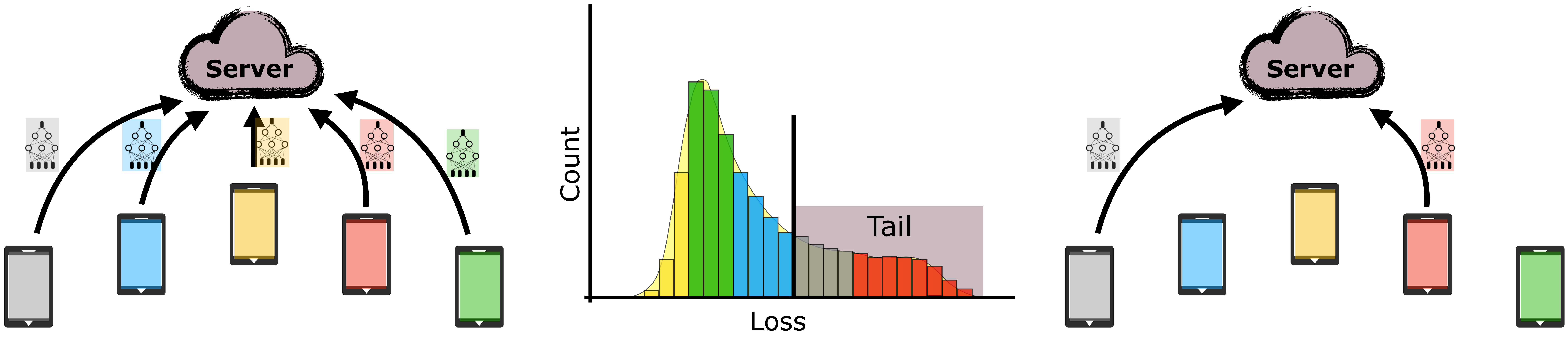
$$\min_{w} \quad \frac{1}{n}\sum_{i=1}^{n} F_i(w)$$

Step 3 of 3: Aggregate updates contributed by ***all clients***

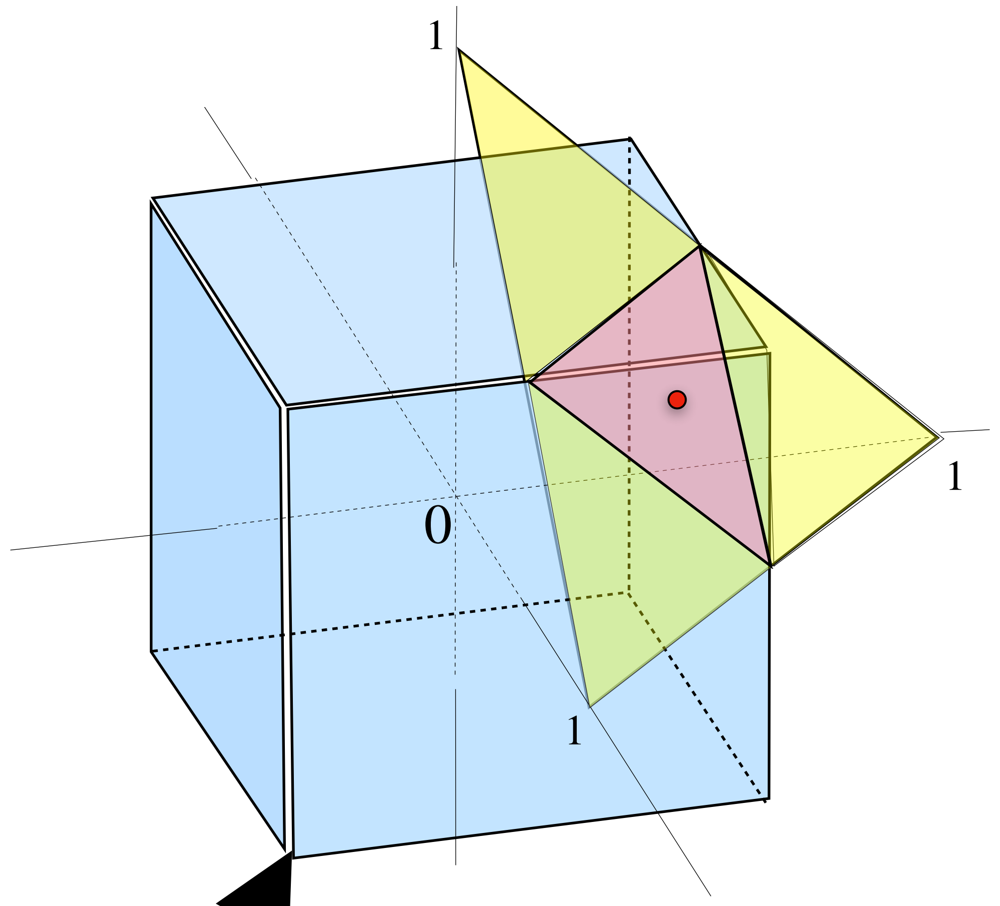
Simplicial-FL Algorithm:

$$\min_{w} \mathbb{S}_\theta\Big(\big(F_1(w), \cdots, F_n(w)\big) \Big)$$

Step 3 of 3: Aggregate updates contributed by ***tail clients*** *only*



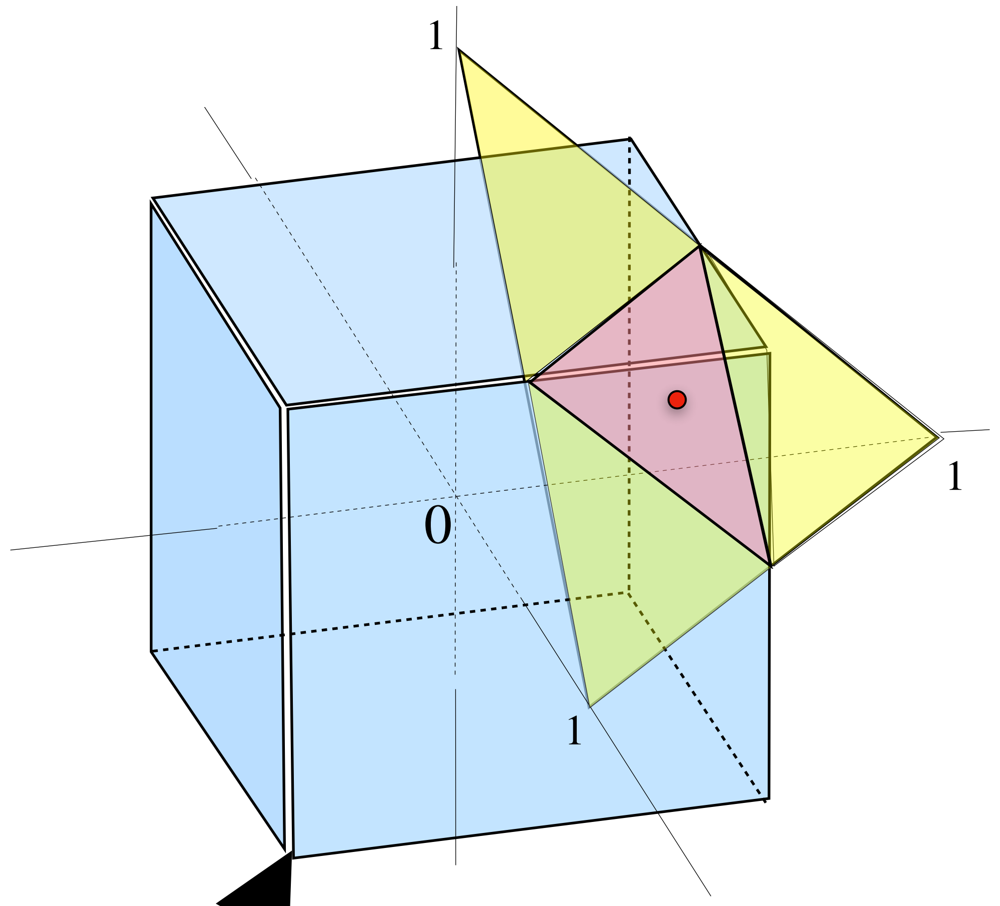
Convergence (Non-convex)



Nonsmooth: Subdifferential from the chain rule

$$\partial F_\theta(w) \ni \sum_{i=1}^{n} \pi_i^\star \nabla F_i(w) \quad \text{where} \quad \pi^\star \in \arg\max_{\pi \in \mathcal{P}_\theta} \sum_i \pi_i F_i(w)$$

Convergence (Non-convex)

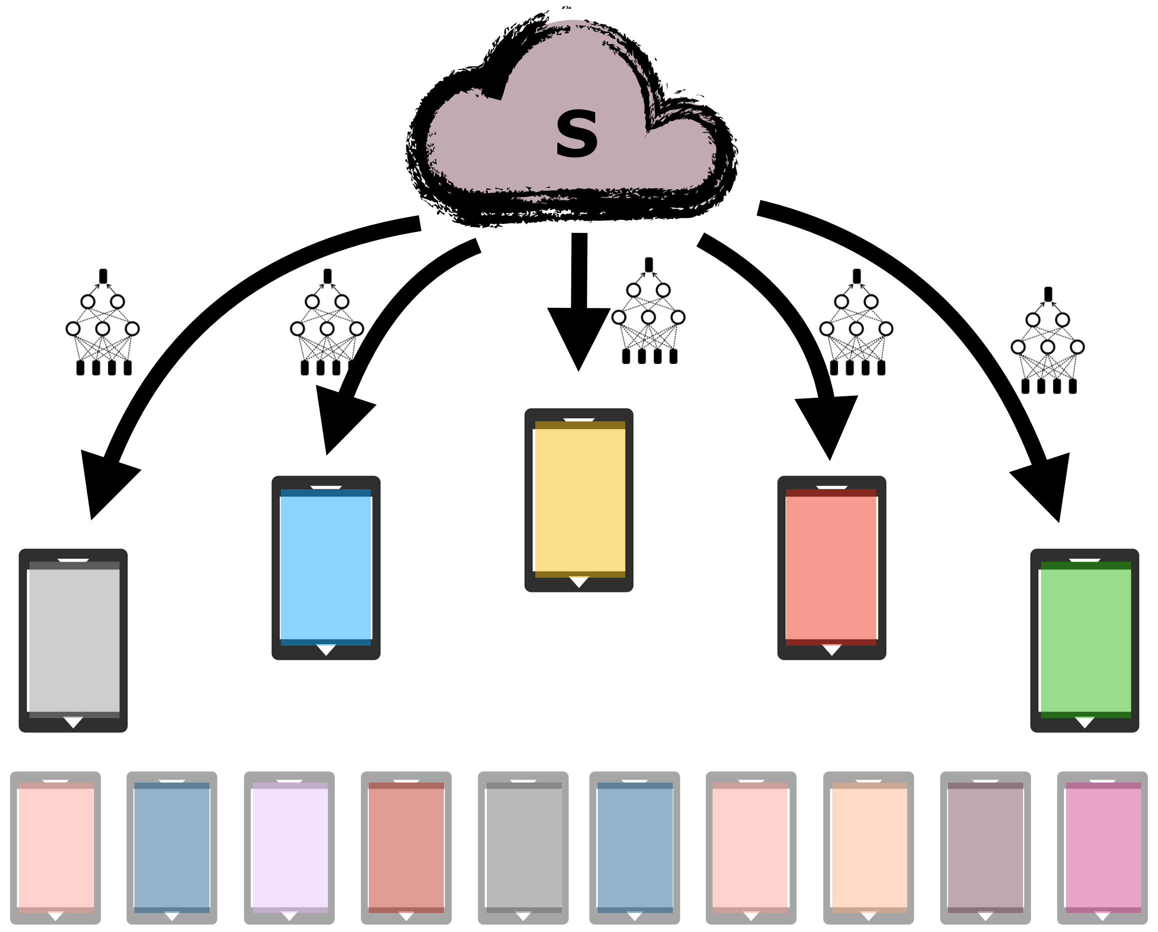


Nonsmooth: Subdifferential from the chain rule

$$\partial F_\theta(w) \ni \sum_{i=1}^{n} \pi_i^\star \nabla F_i(w) \quad \text{where} \quad \pi^\star \in \arg\max_{\pi \in \mathscr{P}_\theta} \sum_i \pi_i F_i(w)$$

Nonlinear: We optimize a surrogate

$$\overline{F}_\theta(w) = \mathbb{E}_{S\,:\,|S|=m}\left[\mathbb{S}_\theta\Big(\big(F_i(w) \,:\, i \in S\big)\Big)\right]$$



Theorem [*P.*, Laguel, Malick, Harchaoui]

Suppose each F_i is L-smooth and G-Lipschitz.

Then, Simplicial-FL satisfies the convergence guarantee:

$$\mathbb{E}\left\|\Phi_\theta^{2L}(w_t)\right\|^2 \leq \sqrt{\frac{\Delta_0 L G^2}{t}} + (1-\tau)^{1/3}\left(\frac{\Delta_0 L G}{t}\right)^{2/3} + \frac{\Delta_0 L}{t}$$

t: #comm. rounds
τ: #local update steps
Δ_0: initial error

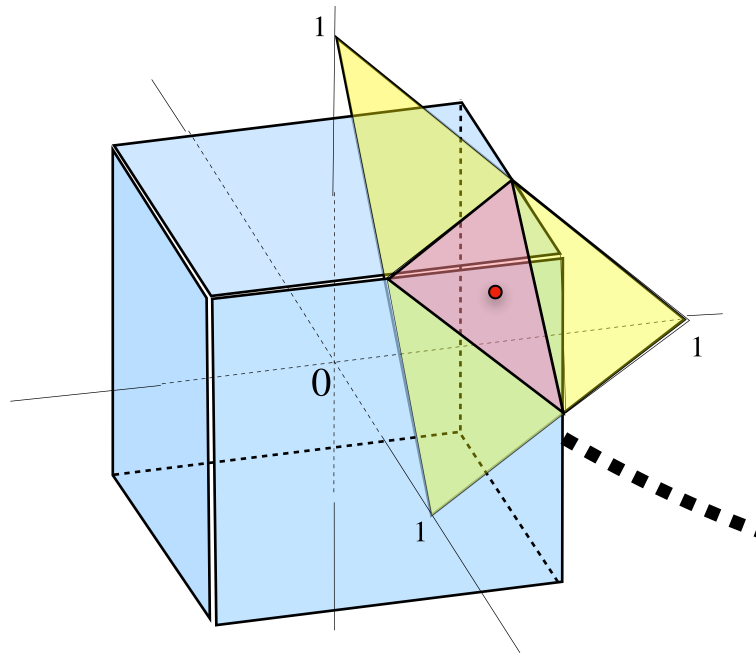
$$\Phi_\theta^\mu(w) = \inf_y \left\{ \overline{F}_\theta(y) + \frac{\mu}{2}\|y - w\|^2 \right\} \quad \longleftarrow$$

Moreau envelope of $\overline{F}_\theta$ | well defined for $\mu > L$

Convergence (strongly convex)

Nonsmooth: Consider the smoothing

strongly convex
neg. entropy



$$F_\theta^\nu(w) = \max_{\pi \in \mathscr{P}_\theta} \left\{ \sum_i \pi_i F_i(w) - \nu \sum_i \pi_i \log \pi_i \right\}$$

$$\nabla F_\theta(w) = \sum_{i=1}^{n} [\pi_\nu]_i^\star \nabla F_i(w)$$ where

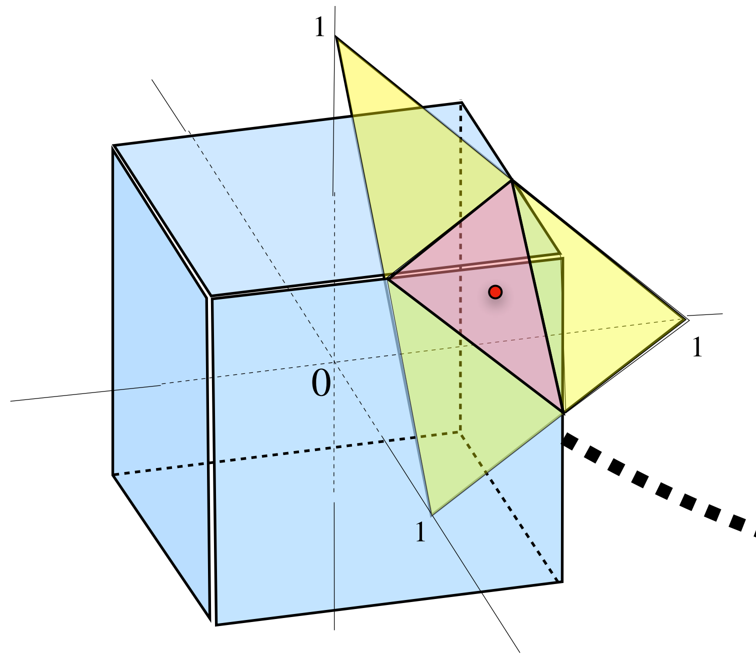
$$\pi_\nu^\star = \arg\max_{\pi \in \mathscr{P}_\theta} \left\{ \sum_i \pi_i F_i(w) - \nu \sum_i \pi_i \log \pi_i \right\}$$

Infimal convolution smoothing [Nesterov. Math. Prog. (2005), Beck & Teboulle. SIOPT (2012)]

Convergence (strongly convex)

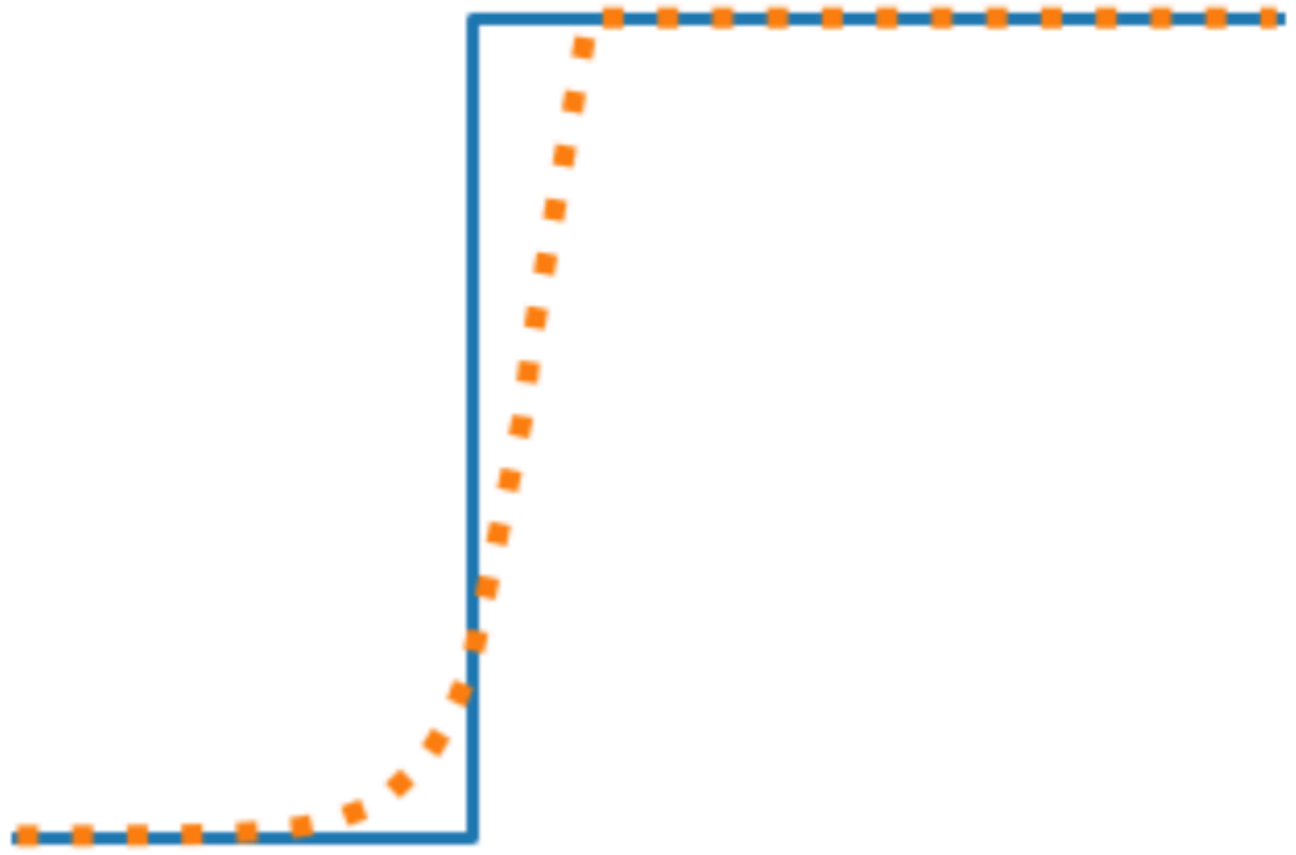
Nonsmooth: Consider the smoothing

strongly convex
neg. entropy



$$F_\theta^\nu(w) = \max_{\pi \in \mathscr{P}_\theta} \left\{ \sum_i \pi_i F_i(w) - \nu \sum_i \pi_i \log \pi_i \right\}$$

$$\nabla F_\theta(w) = \sum_{i=1}^{n} [\pi_\nu]_i^\star \, \nabla F_i(w)$$ where



$$\pi_\nu^\star = \arg\max_{\pi \in \mathscr{P}_\theta} \left\{ \sum_i \pi_i F_i(w) - \nu \sum_i \pi_i \log \pi_i \right\}$$

$$\pi^\star = \arg\max_{\pi \in \mathscr{P}_\theta} \left\{ \sum_i \pi_i F_i(w) \right\}$$

Infimal convolution smoothing [Nesterov. Math. Prog. (2005), Beck & Teboulle. SIOPT (2012)]

Theorem [*P.*, Laguel, Malick, Harchaoui]

Suppose each F_i is L-smooth and G-Lipschitz, and add a regularization $\frac{\lambda}{2}\|w\|^2$.

Then, Simplicial-FL satisfies the convergence guarantee:

$$\mathbb{E}\left[\overline{F}_\theta(w_t) - \overline{F}_\theta^\star\right] \le \lambda \Delta_0 \exp\left(-\frac{t}{\sqrt{2\kappa^3}}\right) + \frac{G^2}{\lambda T} + \frac{G^2\kappa^2}{\lambda T^2}$$

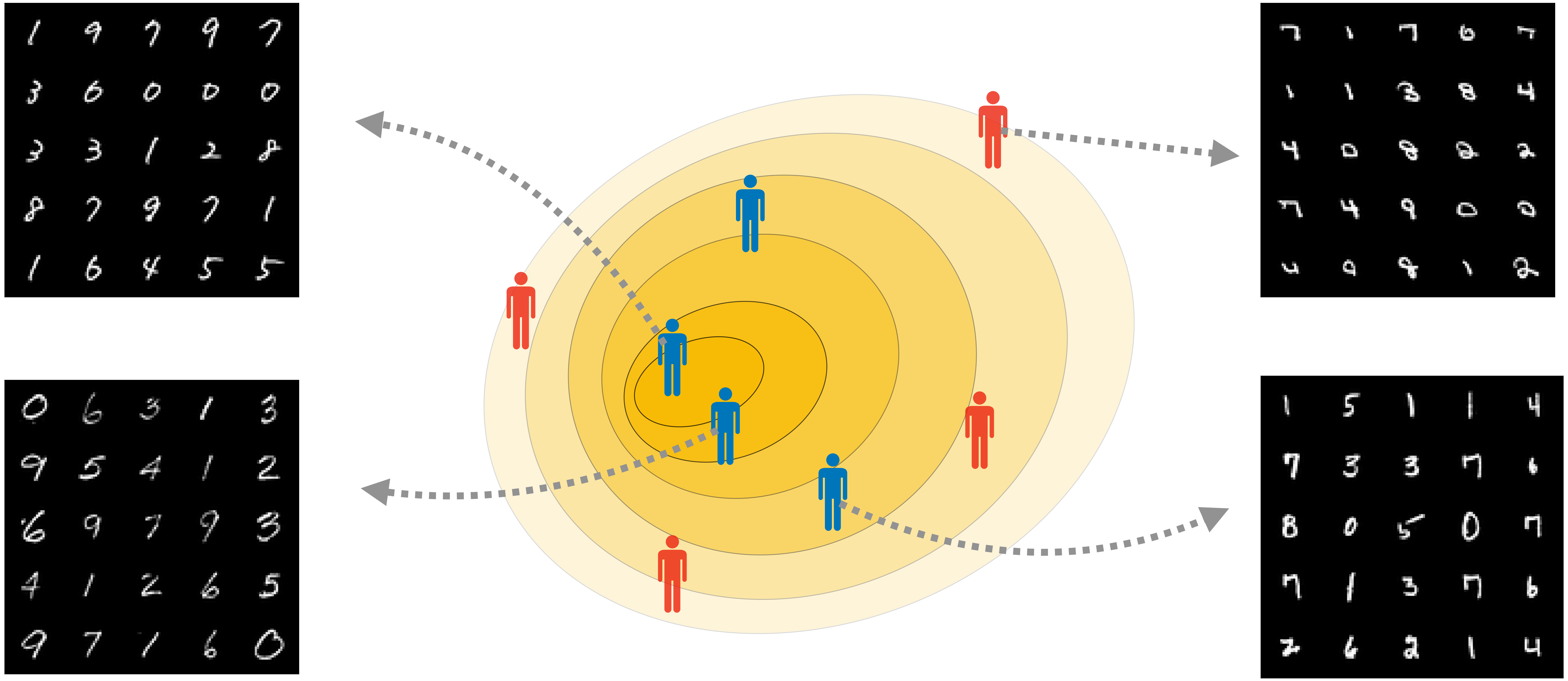
t: #comm. rounds
τ: #local update steps
Δ_0: initial error

$\kappa = L/\lambda$ is the condition number

Outline

- Background
- Distributional Robustness with Simplicial-FL
- Algorithm & Convergence Guarantees
- **Numerical Results**

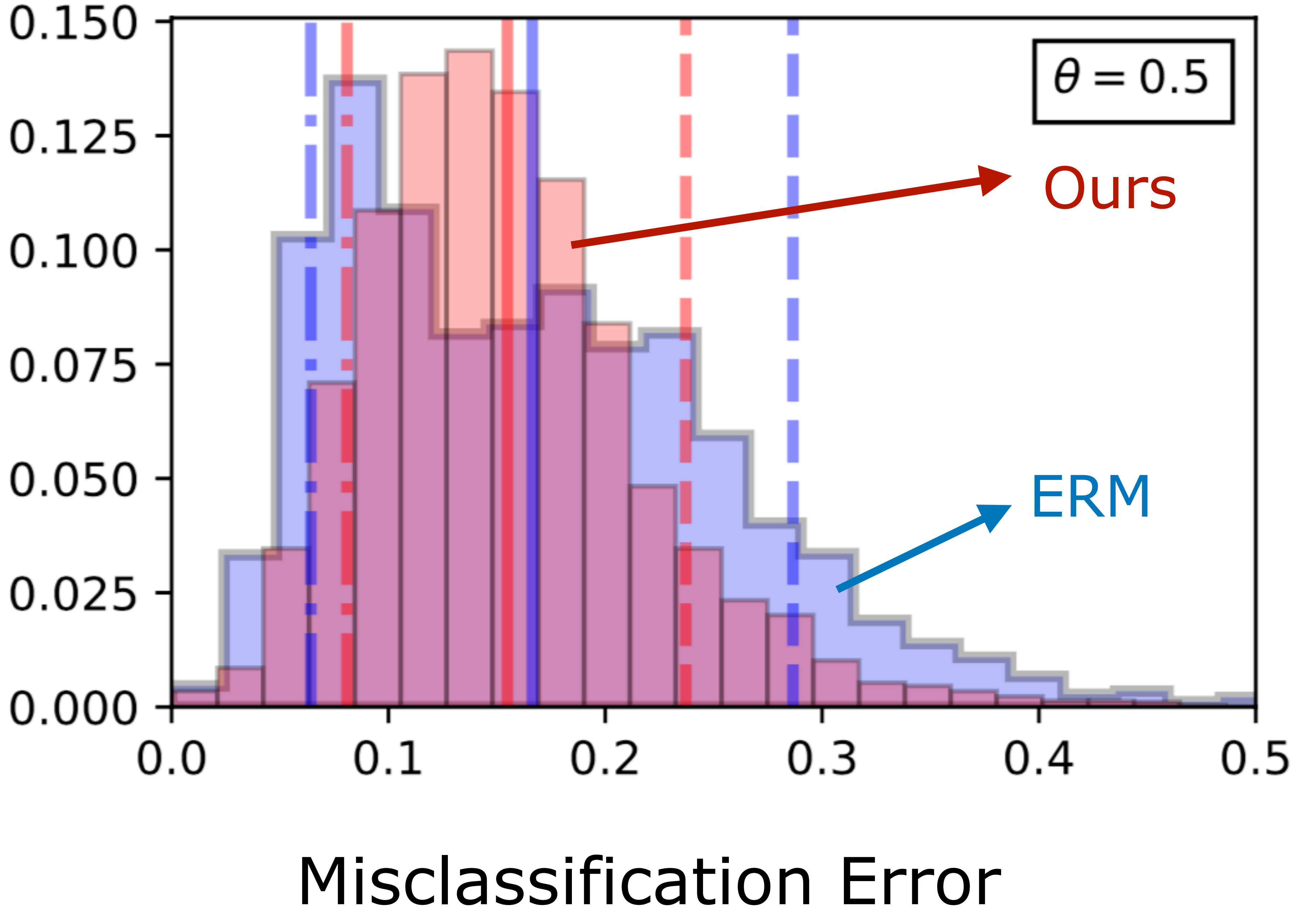
Experiments on EMNIST



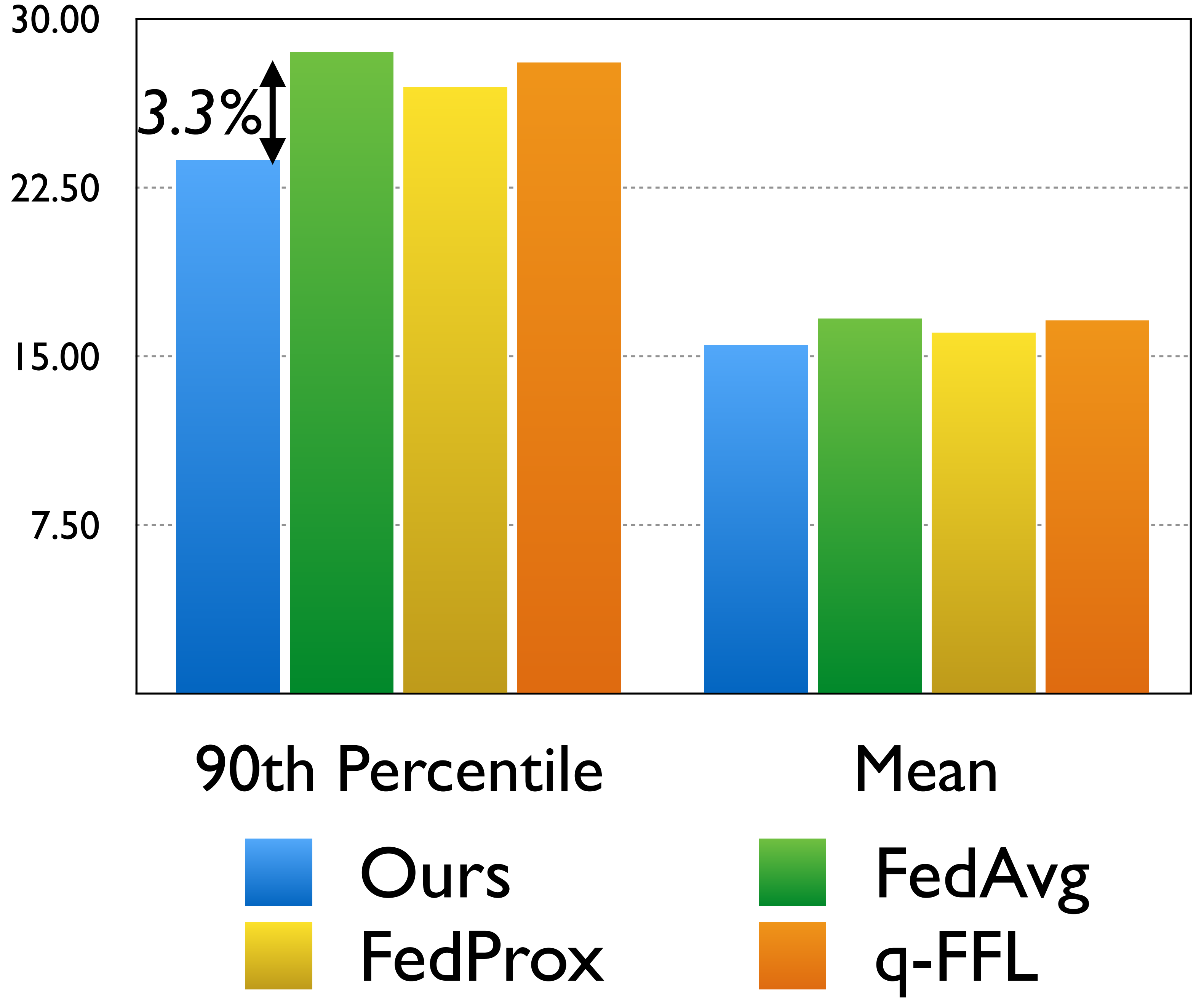
Objective
Misclassification Error
Mean
90th percentile
Mean Value
90th percentile
Train loss
Test misc. error
Communication Rounds
FedAvg
Simplicial-FL
$\theta = 0.8$
$\theta = 0.5$
$\theta = 0.1$

Experiments on EMNIST

Histogram of errors



Misclassif. Error



- Simplicial-FL has the smallest 90th percentile error
- Simplicial-FL is competitive on the mean error

Distributionally robust learning in PyTorch

```python
import torch.nn.functional as F
from sqwash import reduce_superquantile

for x, y in dataloader:
    y_hat = model(x)
    batch_losses = F.cross_entropy(y_hat, y, reduction='none')  # must set `reduction='none'`
    loss = reduce_superquantile(batch_losses, superquantile_tail_fraction=0.5)  # Additional line
    loss.backward()  # Proceed as usual from here
    ...
```

Install: **pip install sqwash**

Documentation: krishnap25.github.io/sqwash/

SCAN ME

Papers

Federated Learning with Heterogeneous Devices: A Superquantile Optimization Approach.
Krishna Pillutla*, Yassine Laguel*, Jérôme Malick, Zaid Harchaoui.
Under Review (arXiv 2112.09429)

A Superquantile Approach to Federated Learning with Heterogeneous Devices.
Yassine Laguel*, Krishna Pillutla*, Jérôme Malick, Zaid Harchaoui.
IEEE CISS (2021).

Superquantiles at Work : Machine Learning Applications and Efficient (Sub)gradient Computation.
Yassine Laguel, Krishna Pillutla, Jérôme Malick, Zaid Harchaoui.
Set-Valued and Variational Analysis (2021).

Code for experiments: https://github.com/krishnap25/simplicial-fl