Omni-Fedge

Federated Algorithm With A Bayesian Approach

K. Sai Anuroop
170030035, Final Year, Dept. of Computer Science and Engineering

Supervised by: **B. N. Bharath**, Dept. of Electrical Engineering

॥ सा विद्या या विमुक्तये ॥
भारतीय प्रौद्योगिकी संस्थान धारवाड
Indian Institute of Technology Dharwad

Indian Institute of Technology Dharwad

December 24, 2020

What is Federated Learning?

- Distributed machine learning architecture where edge-devices learn shared predictive model collaboratively

*https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

What is Federated Learning?

- Distributed machine learning architecture where edge-devices learn shared predictive model collaboratively

Figure: FL Architecture *

*https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

Where is FL being used?

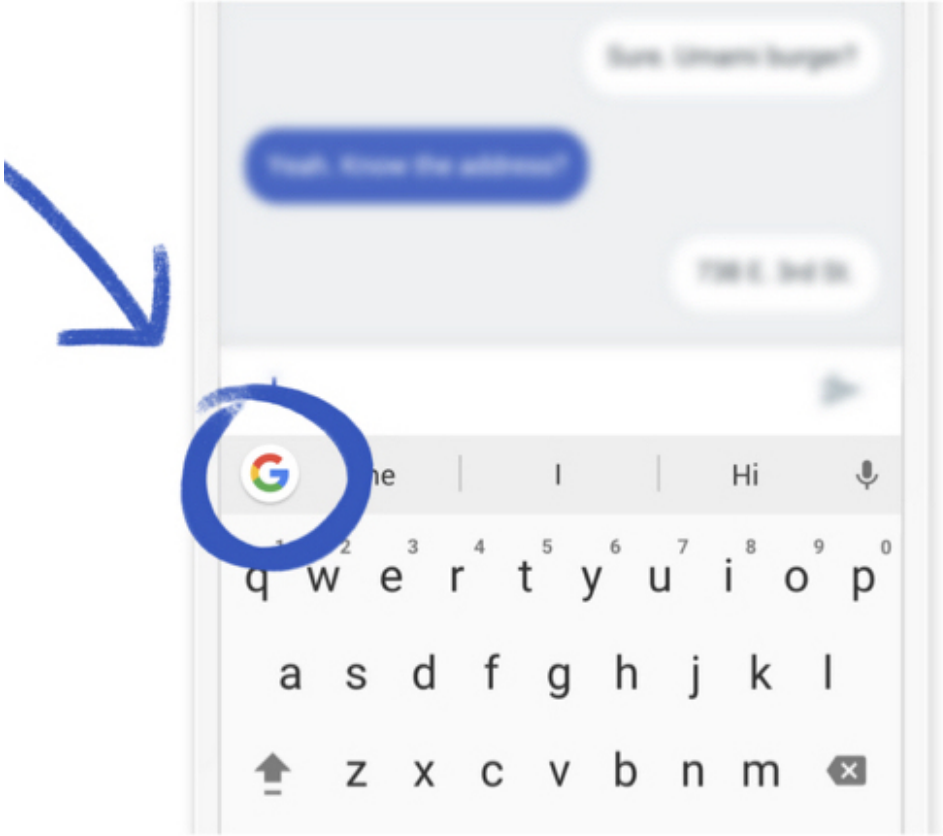


Figure: GBoard: Query prediction using FL †

† https://blog.google/products/search/gboard-now-on-android/

Where can FL be used?

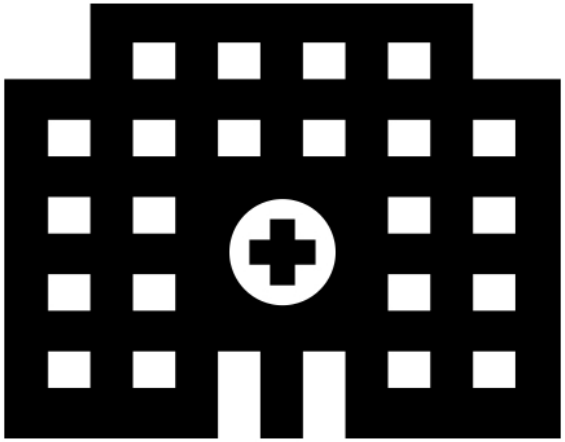


Figure: Private learning among hospitals [‡]

[‡] https://www.nature.com/articles/s41746-020-00323-1

Where can FL be used?

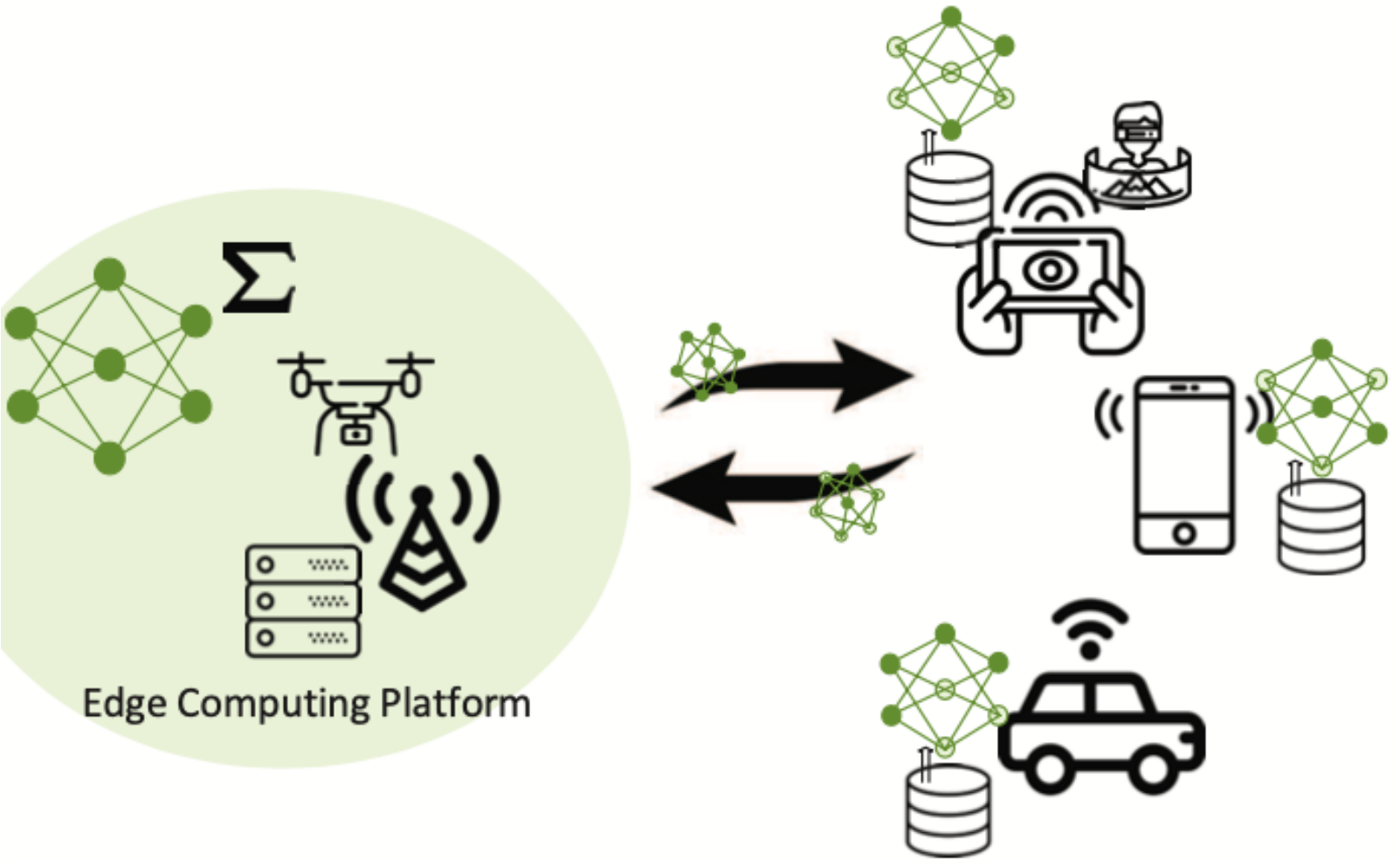


Figure: Self-driving cars in an autonomous vehicle network [§]

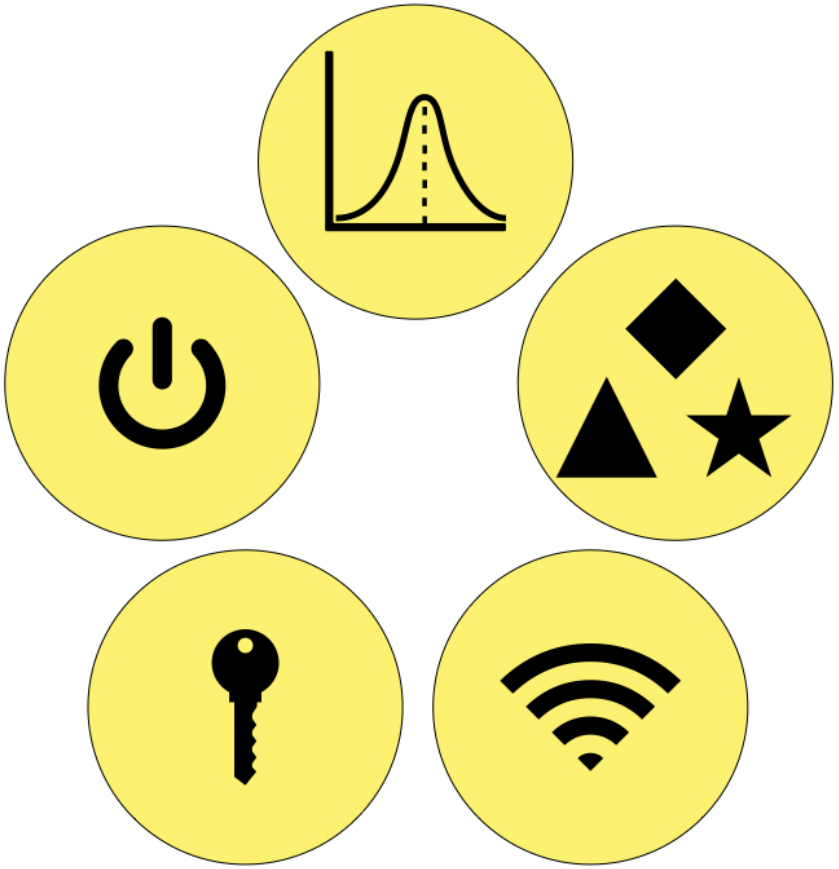
[§] https://ieeexplore.ieee.org/abstract/document/9141214

What are the fundamental challenges in FL?

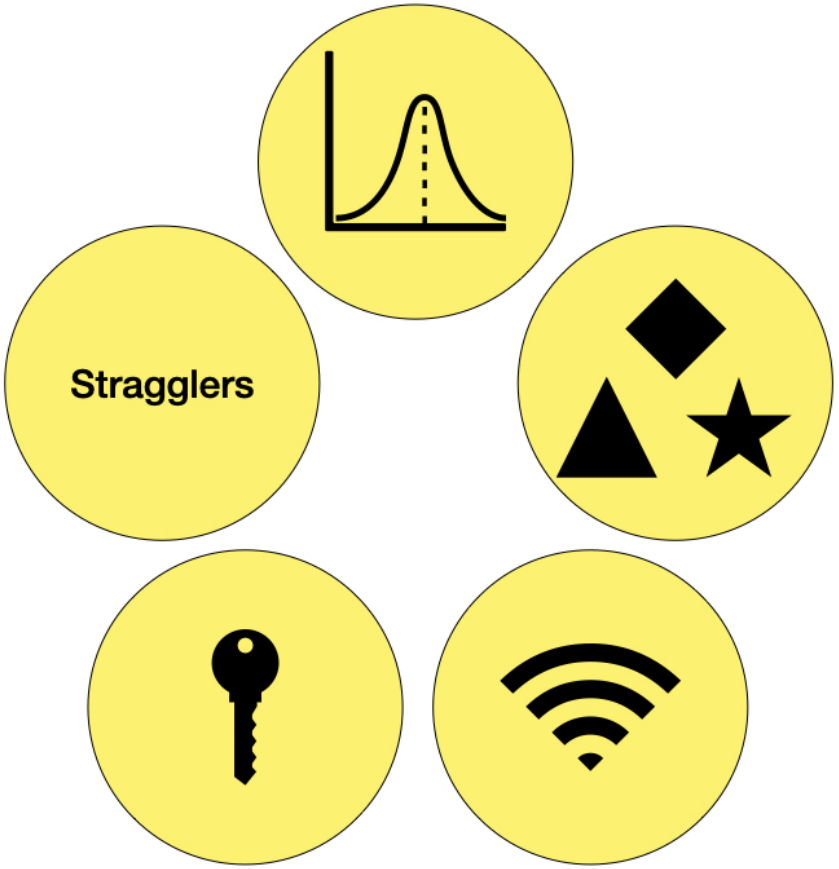
- Implementing FL in practice imposes several challenges

What are the fundamental challenges in FL?

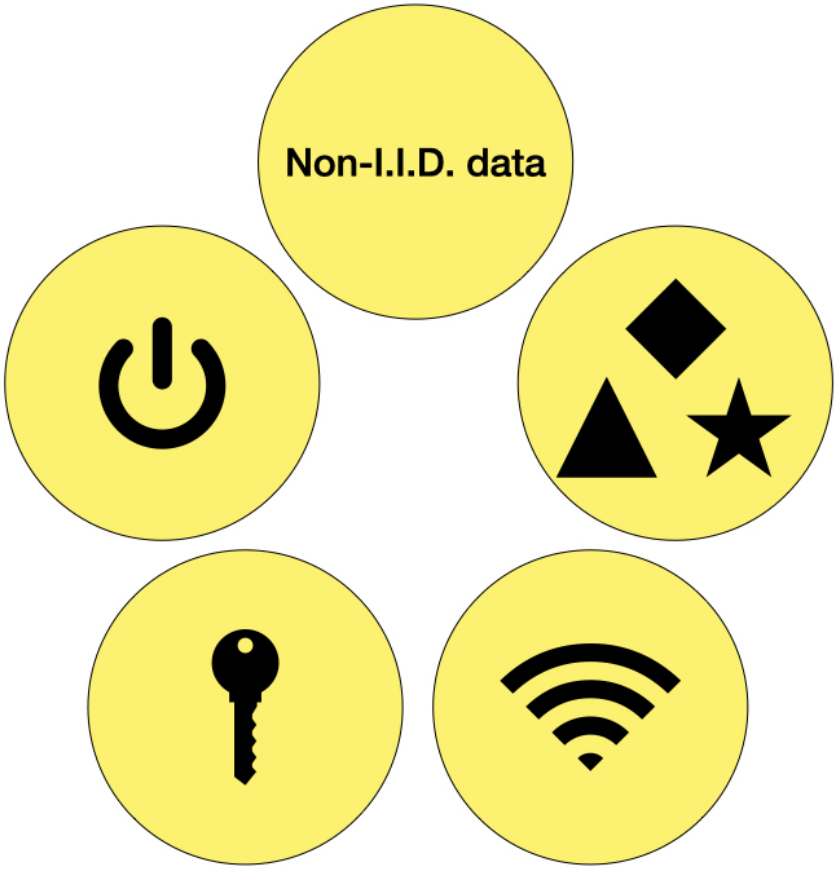
- Implementing FL in practice imposes several challenges



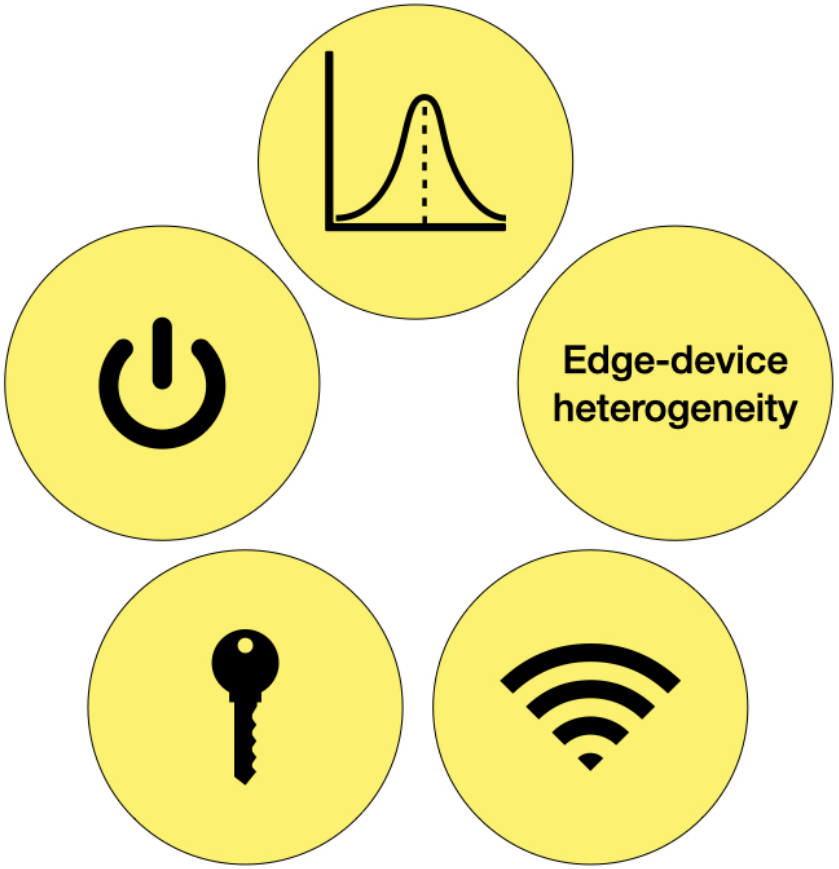
What are the fundamental challenges in FL?



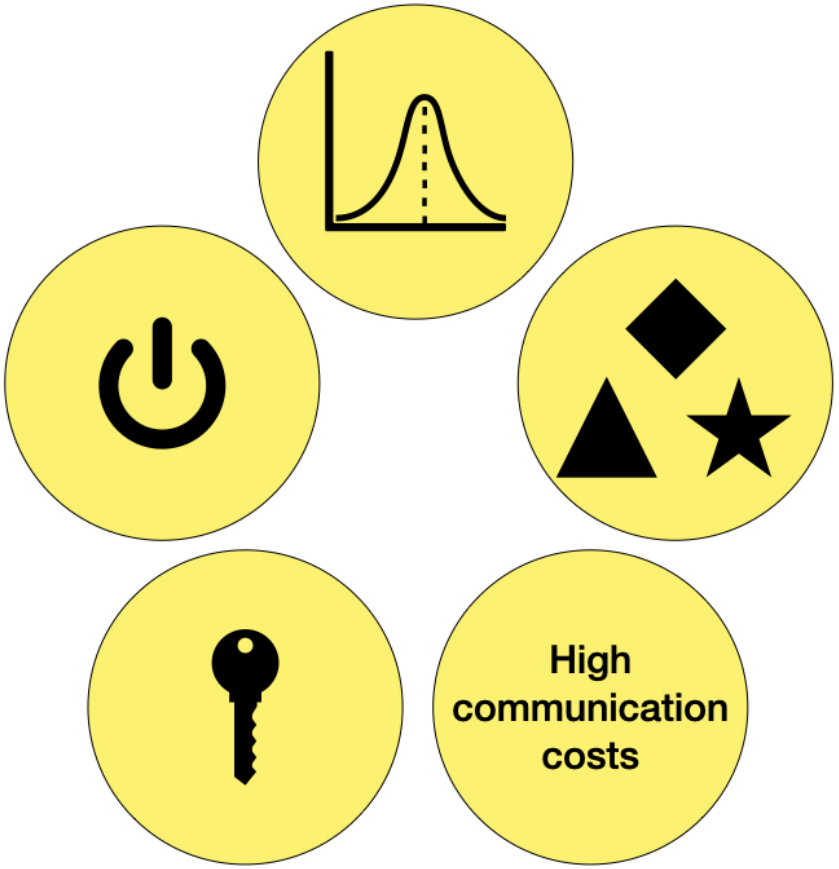
What are the fundamental challenges in FL?



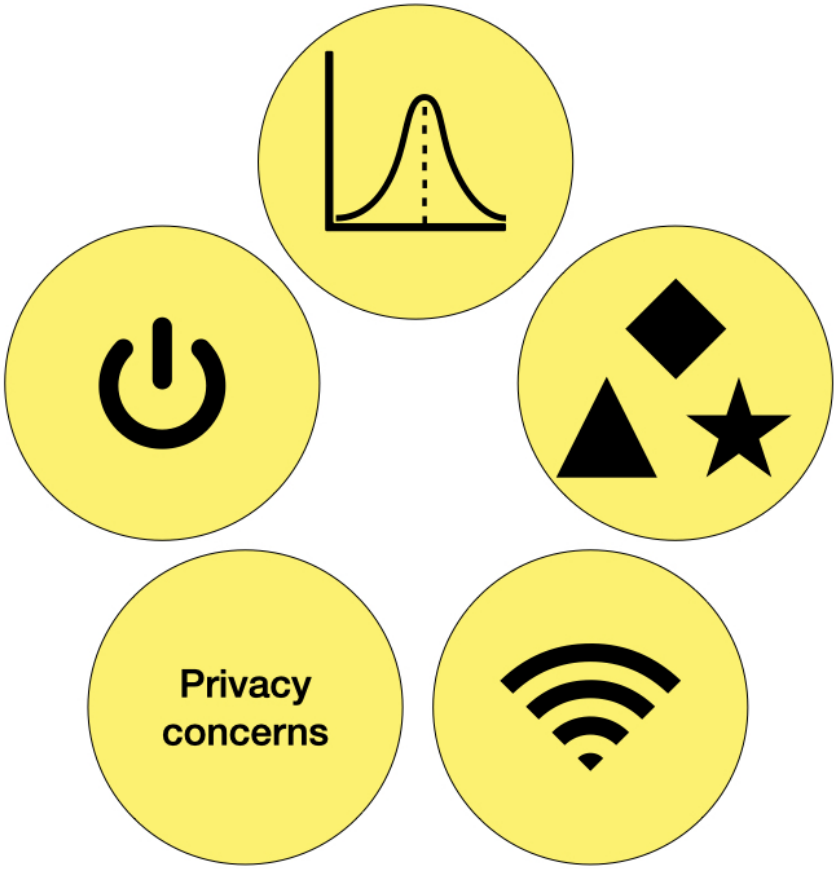
What are the fundamental challenges in FL?



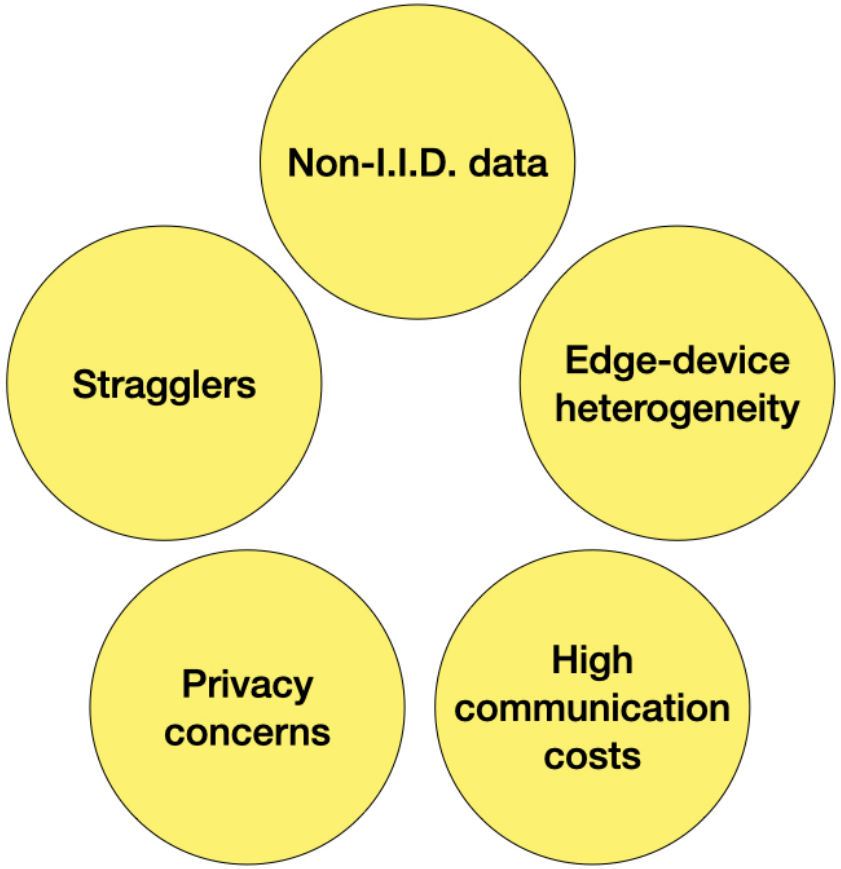
What are the fundamental challenges in FL?



What are the fundamental challenges in FL?



What are the fundamental challenges in FL?



Recalling what is a neural network?

$z_{11} \longrightarrow$

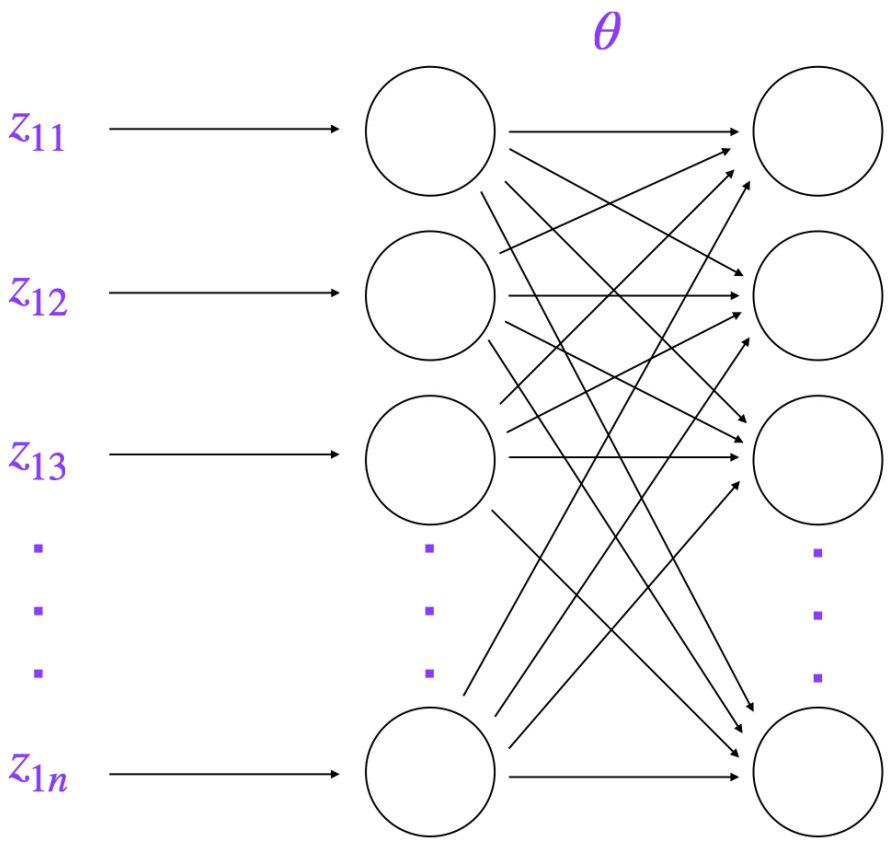
$z_{12} \longrightarrow$

$z_{13} \longrightarrow$

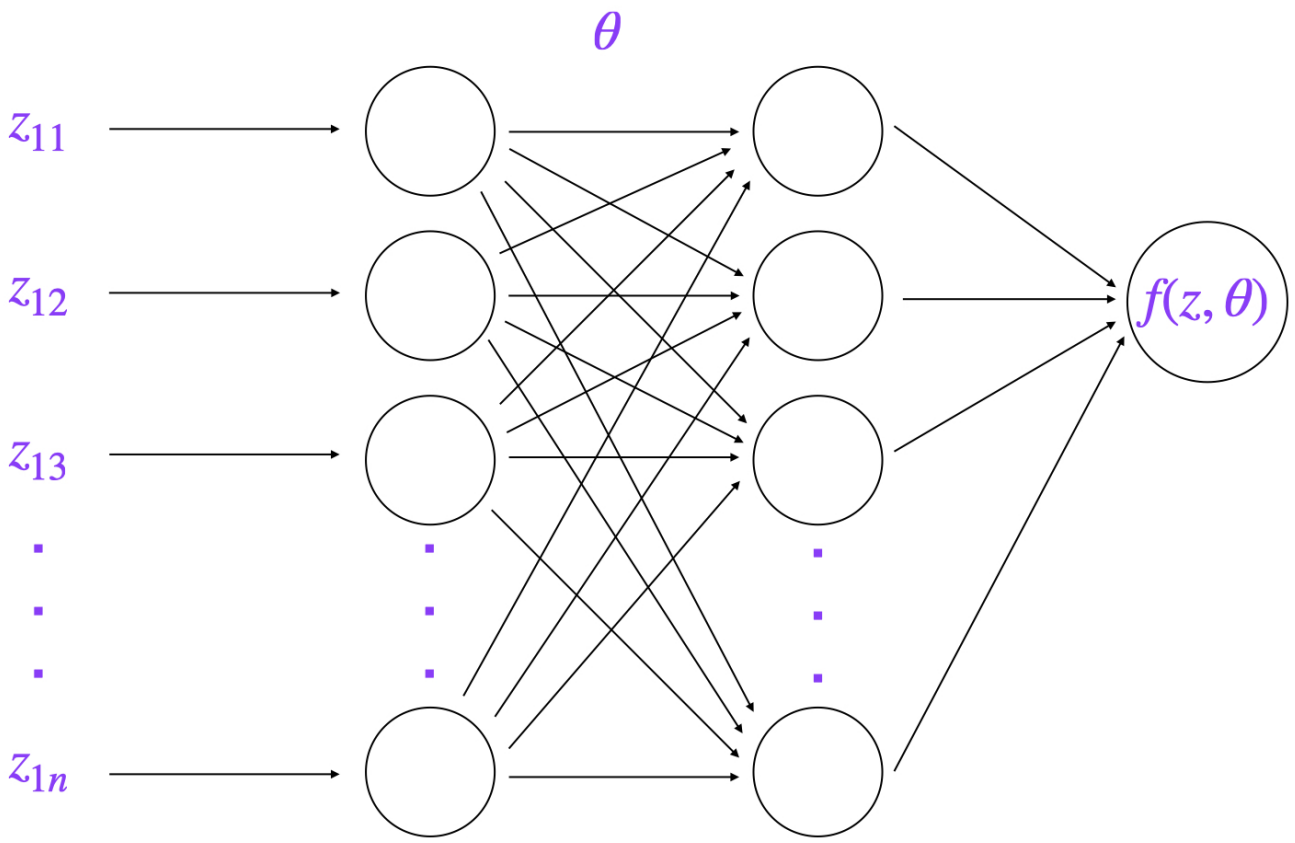
$\vdots$

$z_{1n} \longrightarrow$

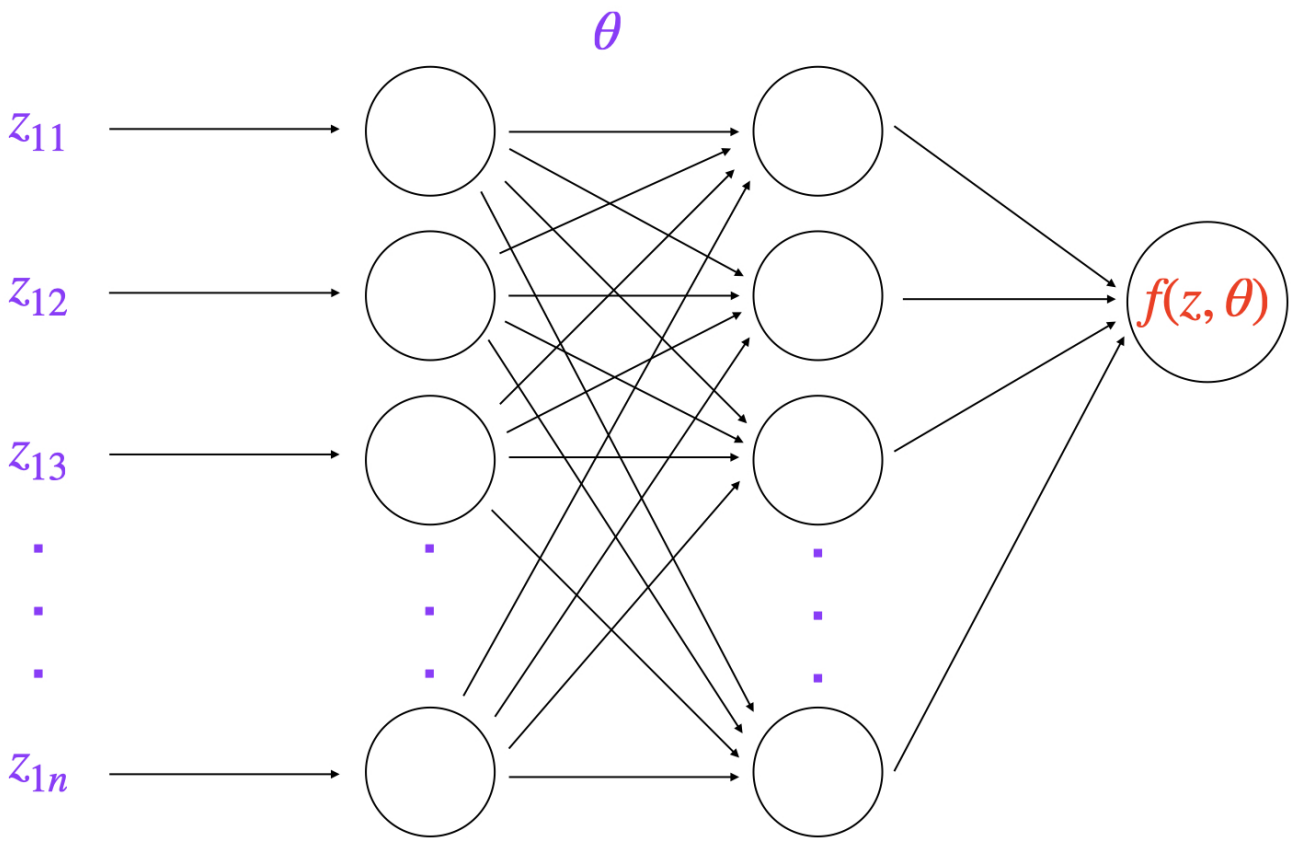
Recalling what is a neural network?



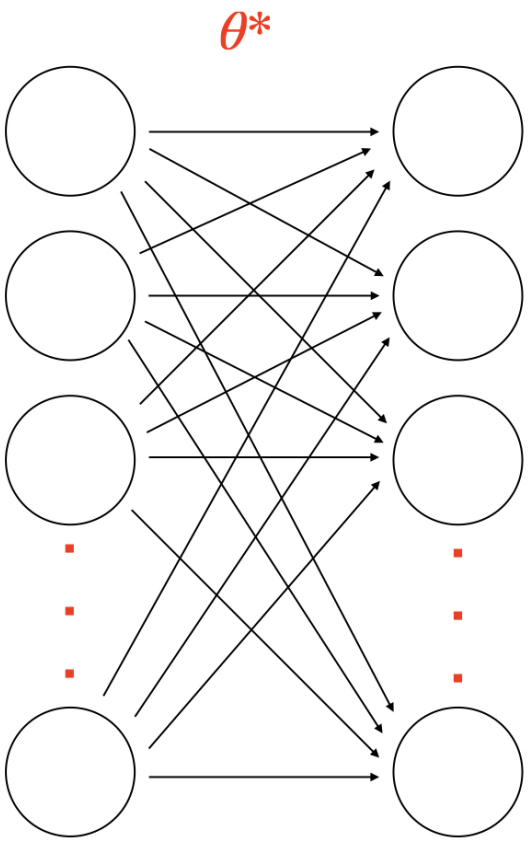
Recalling what is a neural network?



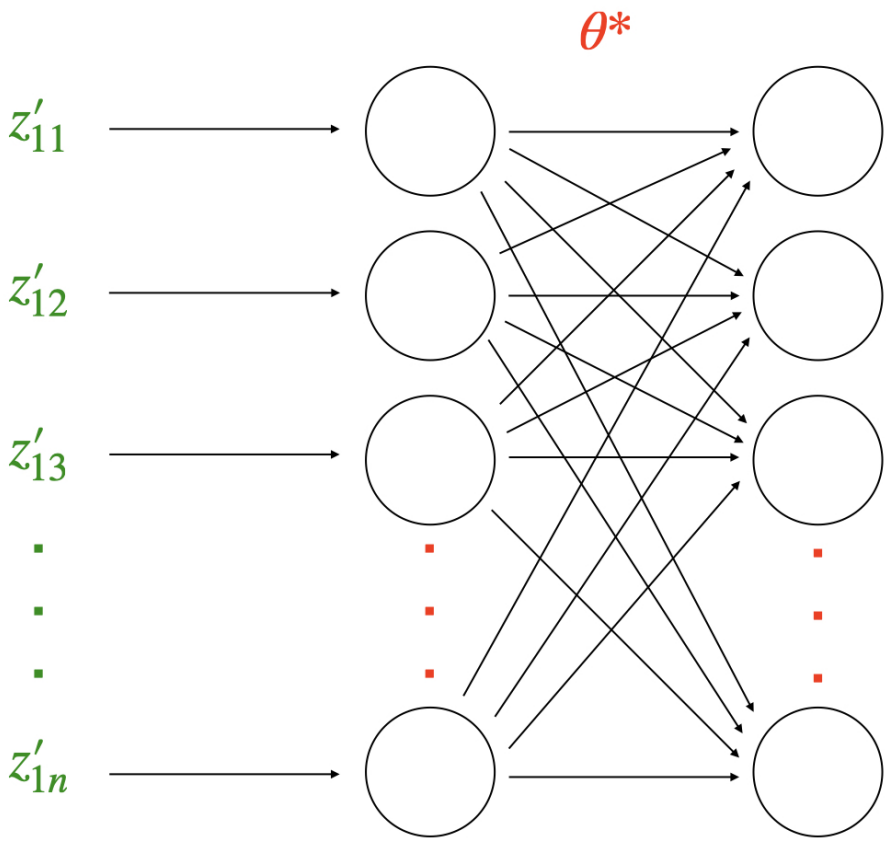
Recalling what is a neural network?



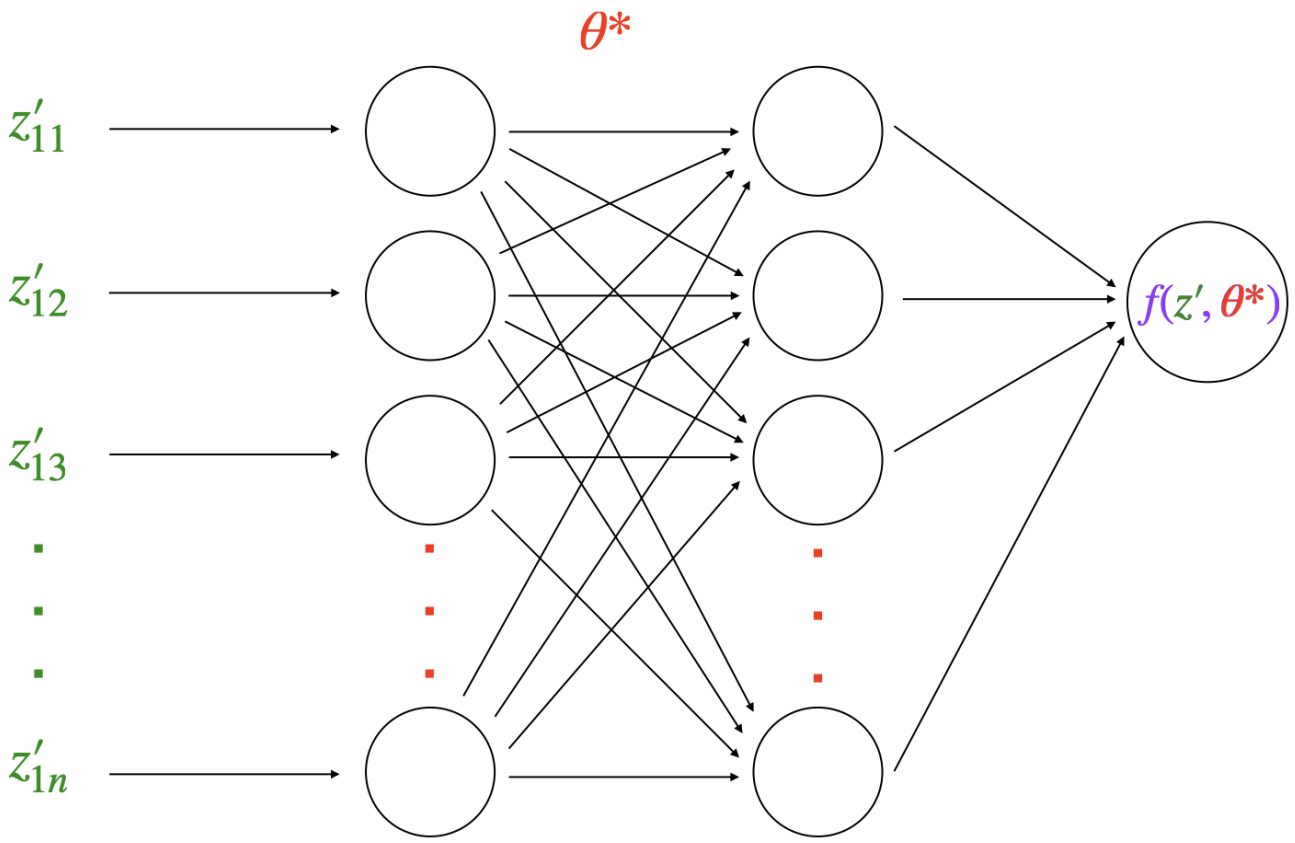
Recalling what is a neural network?



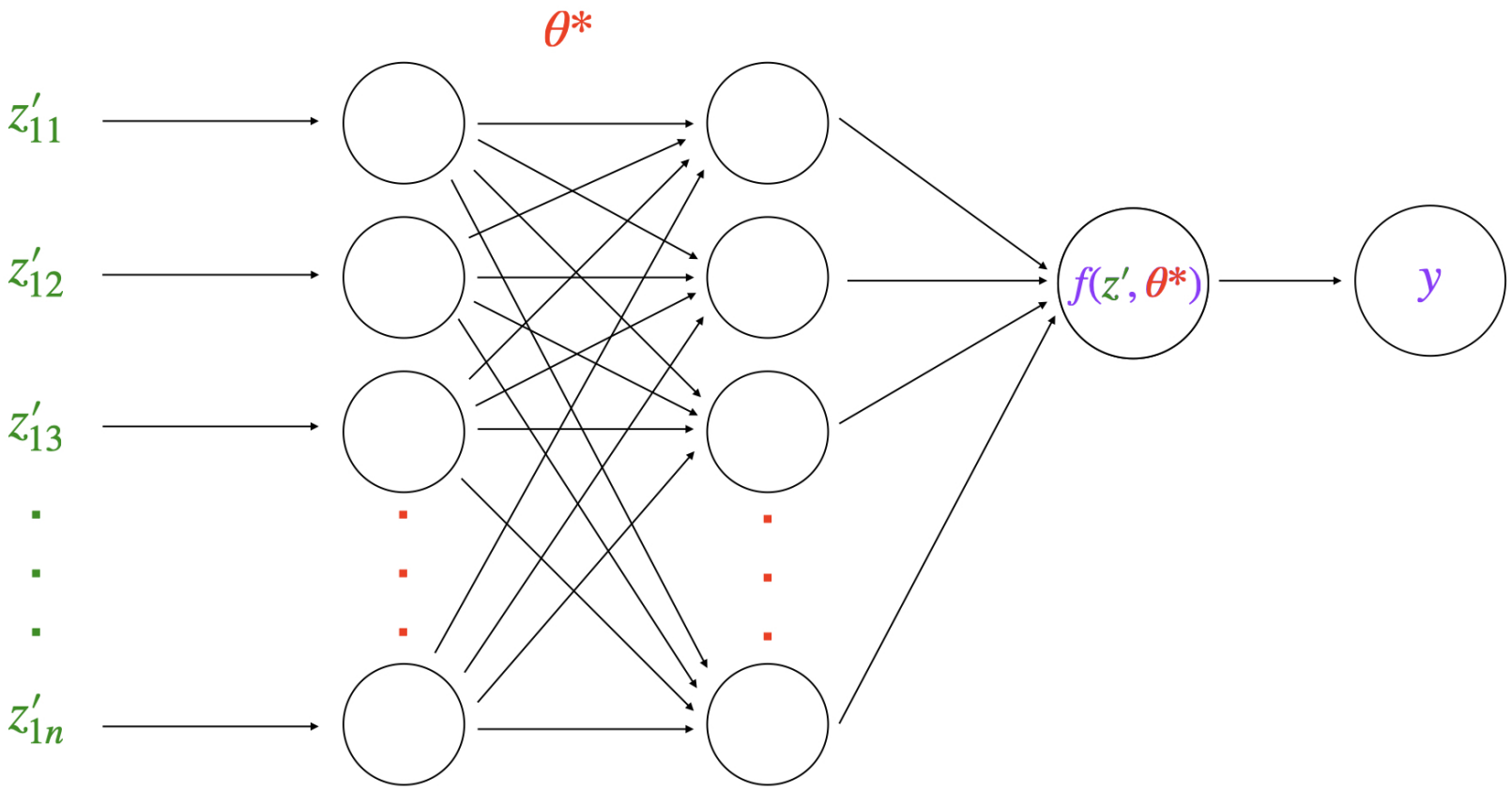
Recalling what is a neural network?



Recalling what is a neural network?



Recalling what is a neural network?



How is FL formulated?

How is FL formulated?

- **Example:** f can be a $0-1$ loss in query prediction

How is FL formulated?

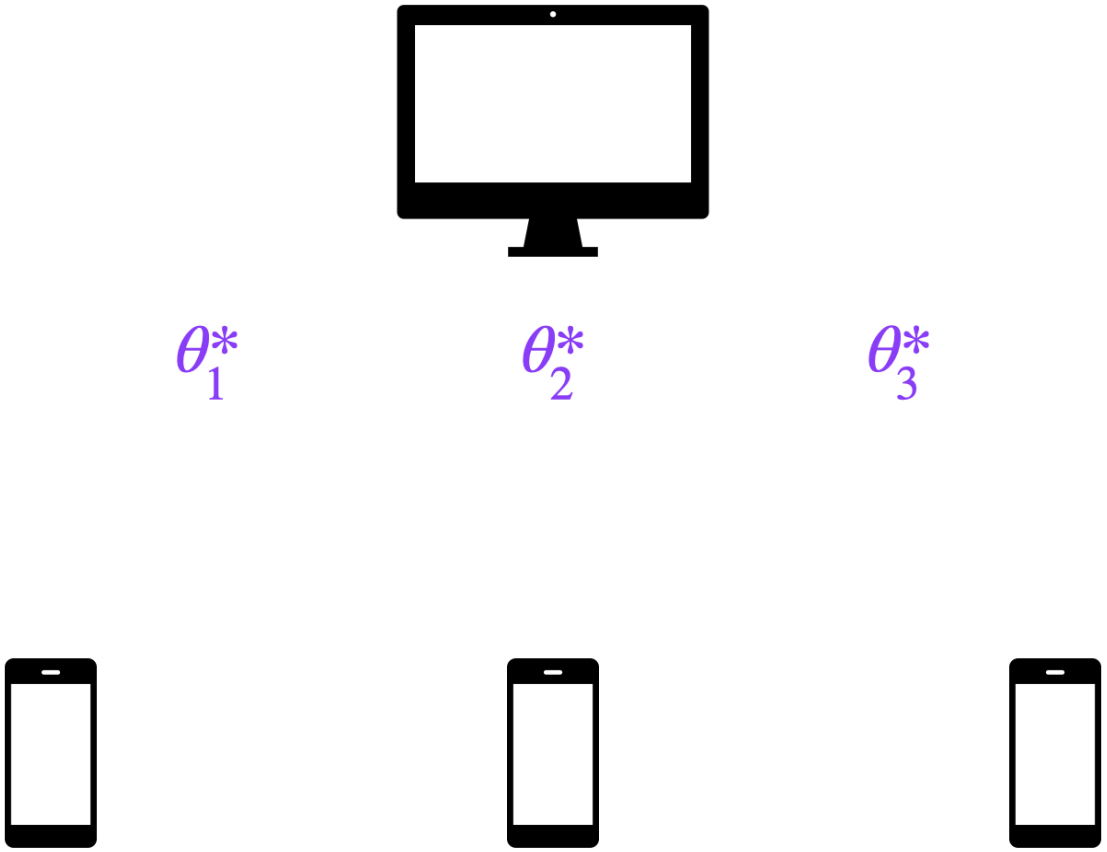
- **Example:** f can be a $0-1$ loss in query prediction

How is FL formulated?

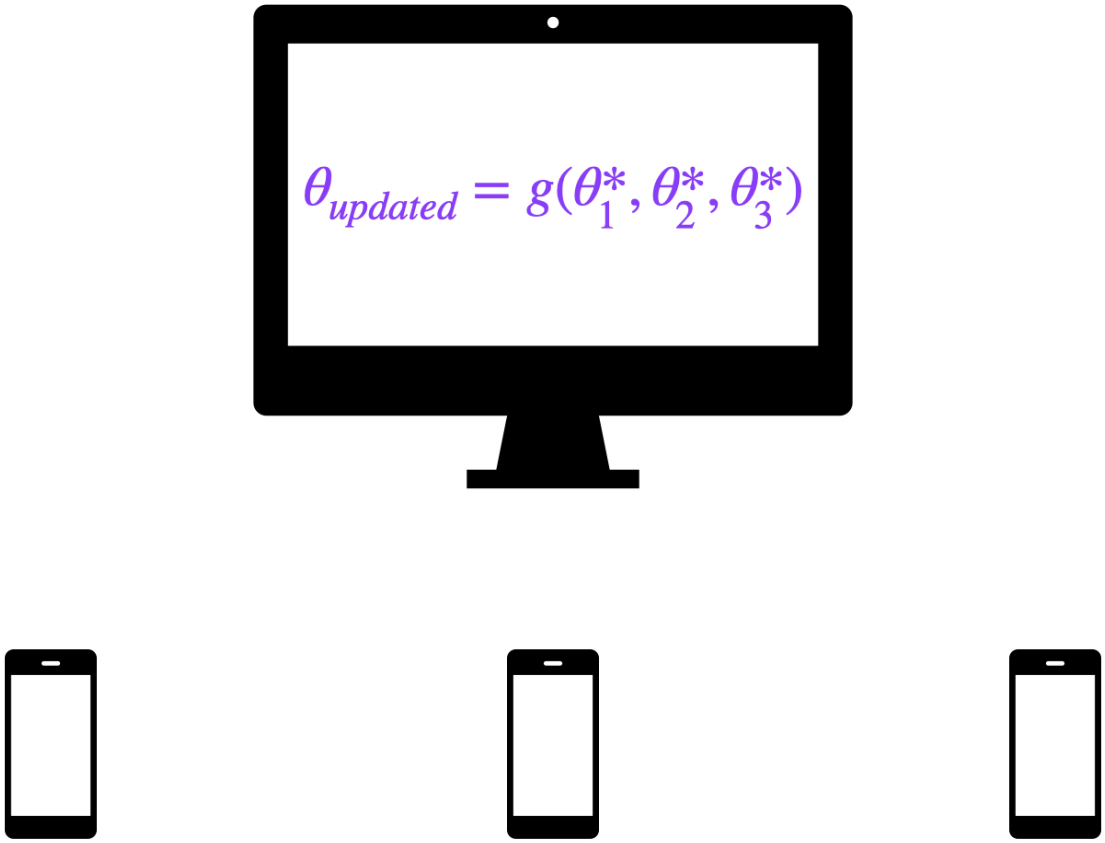
$$\theta_1^{t+1} = \theta_1^t - \eta_1 \nabla_{\theta_1} f_1(\theta_1, z_1) \qquad \theta_2^{t+1} = \theta_2^t - \eta_2 \nabla_{\theta_2} f_2(\theta_2, z_2) \qquad \theta_3^{t+1} = \theta_3^t - \eta_3 \nabla_{\theta_3} f_3(\theta_3, z_3)$$

How is FL formulated?

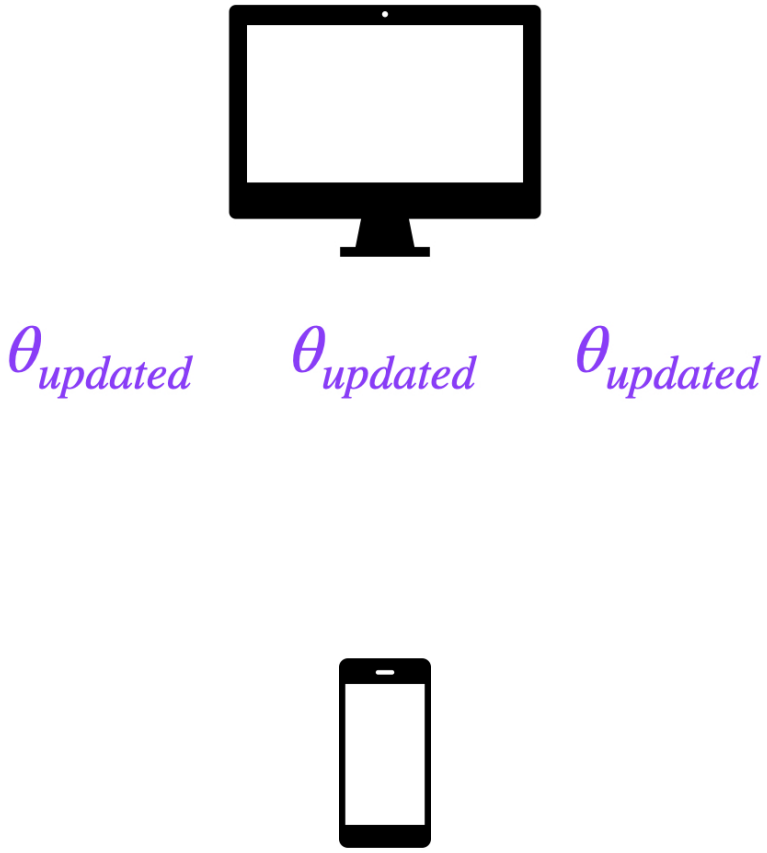
How is FL formulated?



How is FL formulated?

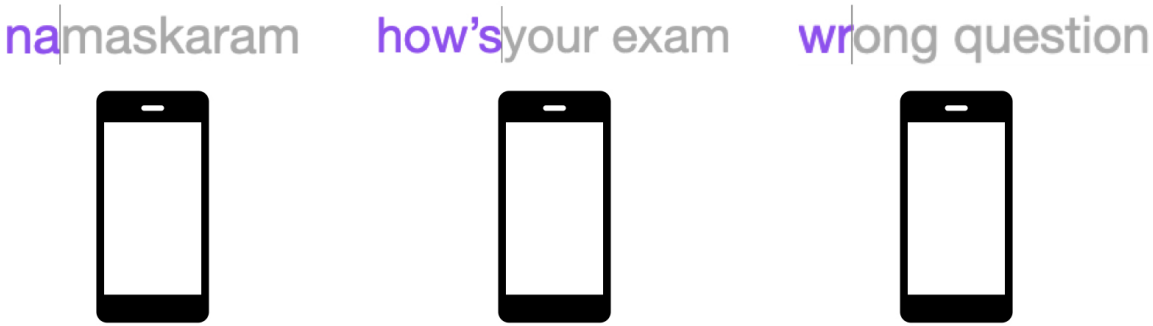


How is FL formulated?



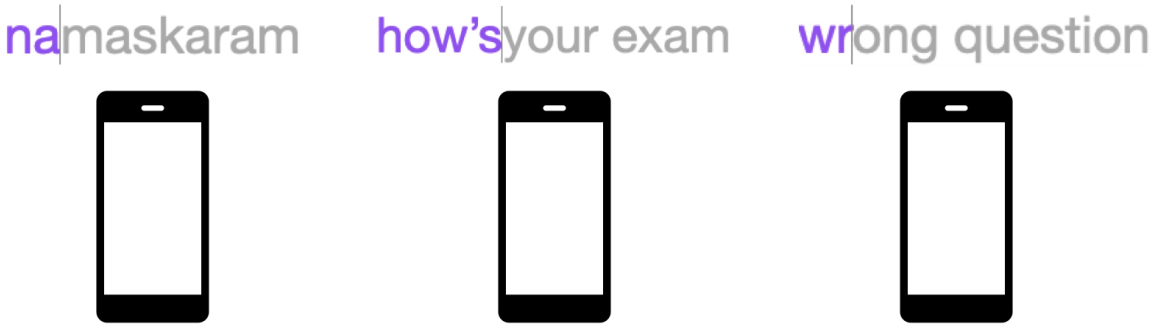
How is FL formulated?

How is FL formulated?



¶http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf

How is FL formulated?



- FedAvg[¶]: $g(\theta_1^*, \theta_2^*, \theta_3^*) = \dfrac{\theta_1^* + \theta_2^* + \theta_3^*}{3}$

[¶] http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf

Our Work: Problem Setting

Our Work: Problem Setting

Our Work: Problem Setting

$$\min_{\theta} \mathbb{E}_{z_1 \sim D_1}\{L_1(z_1, \theta)\}$$

$$\min_{\theta} \mathbb{E}_{z_3 \sim D_3}\{L_3(z_3, \theta)\}$$

$$\min_{\theta} \mathbb{E}_{z_2 \sim D_2}\{L_2(z_2, \theta)\}$$

Our Work: Problem Setting

$$\min_{\theta} \mathbb{E}_{z_1 \sim D_1}\{L_1(z_1, \theta)\}$$

$$\min_{\theta} \mathbb{E}_{z_3 \sim D_3}\{L_3(z_3, \theta)\}$$

$$\min_{\theta} \mathbb{E}_{z_2 \sim D_2}\{L_2(z_2, \theta)\}$$

Our Work: Problem Setting

$$\min_{\theta} \hat{\mathbb{E}} L_1(z_1, \theta) \qquad \min_{\theta} \hat{\mathbb{E}} L_2(z_2, \theta) \qquad \min_{\theta} \hat{\mathbb{E}} L_3(z_3, \theta)$$

Our Work: Problem Setting

$$\min_{\theta} \hat{\mathbb{E}} L_1(z_1, \theta) \qquad \min_{\theta} \hat{\mathbb{E}} L_2(z_2, \theta) \qquad \min_{\theta} \hat{\mathbb{E}} L_3(z_3, \theta)$$

Our Work: Problem Setting

$$\min_{\theta} \hat{\mathbb{E}} L_1(z_1, \theta^{(1)}, \theta^{(sh)})$$

$$\min_{\theta} \hat{\mathbb{E}} L_3(z_3, \theta^{(3)}, \theta^{(sh)})$$

$$\min_{\theta} \hat{\mathbb{E}} L_2(z_2, \theta^{(2)}, \theta^{(sh)})$$

Our Work: Neural Network

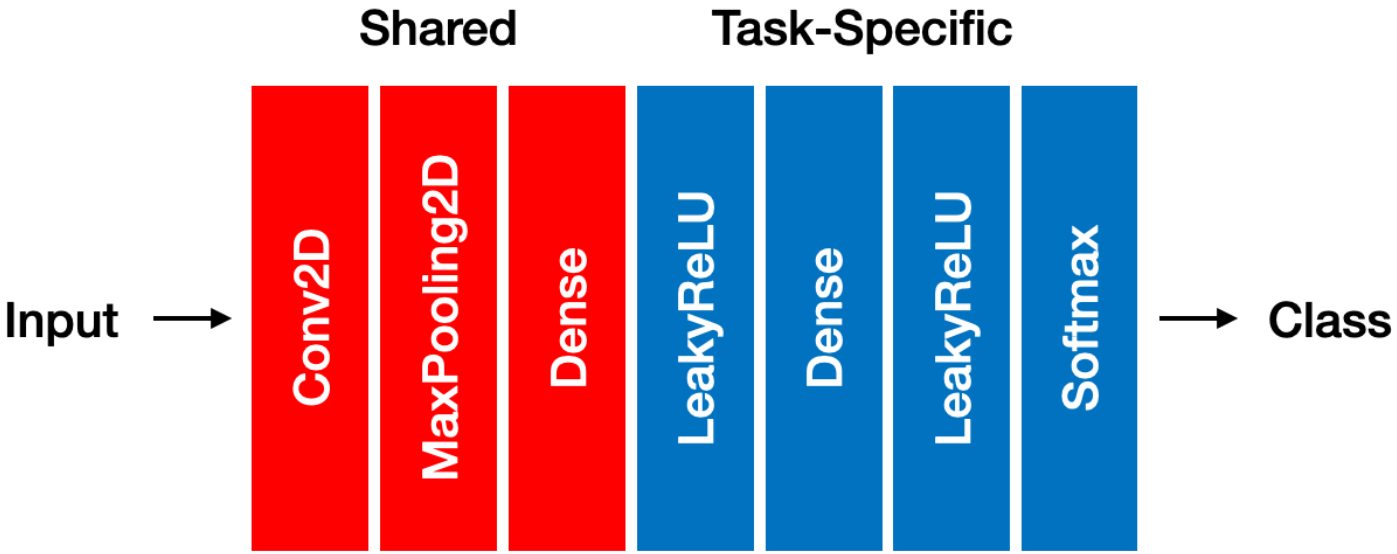


Figure: Shared and task-specific layers of the neural network.

Our Work: Bayesian Approach

Our Work: Bayesian Approach

$$\omega_{11} e^{-\omega_{11} \hat{\mathbb{E}} L_1(z_1, \theta^{(1)}, \theta^{(sh)})}$$

$$\omega_{13} e^{-\omega_{13} \hat{\mathbb{E}} L_3(z_3, \theta^{(1)}, \theta^{(sh)})}$$

$$\omega_{12} e^{-\omega_{12} \hat{\mathbb{E}} L_2(z_2, \theta^{(1)}, \theta^{(sh)})}$$

Our Work: Bayesian Approach

$$\omega_{11}\omega_{12}\omega_{13}e^{-(\omega_{11}\hat{\mathbb{E}}L_1(z_1,\theta^{(1)},\theta^{(sh)})+\omega_{12}\hat{\mathbb{E}}L_2(z_2,\theta^{(1)},\theta^{(sh)})+\omega_{13}\hat{\mathbb{E}}L_3(z_3,\theta^{(1)},\theta^{(sh)}))}$$

- We refer to the above distribution as $Q(\theta^{(1)}, \theta^{(sh)})$ and the true joint distribution as $P(\theta^{(1)}, \theta^{(sh)})$

Our Work: Bayesian Approach

$$\omega_{11}\omega_{12}\omega_{13}e^{-(\omega_{11}\hat{\mathbb{E}}L_1(z_1,\theta^{(1)},\theta^{(sh)})+\omega_{12}\hat{\mathbb{E}}L_2(z_2,\theta^{(1)},\theta^{(sh)})+\omega_{13}\hat{\mathbb{E}}L_3(z_3,\theta^{(1)},\theta^{(sh)}))}$$

- We refer to the above distribution as $Q(\theta^{(1)}, \theta^{(sh)})$ and the true joint distribution as $P(\theta^{(1)}, \theta^{(sh)})$

Our Work: Bayesian Approach

Our Work: Bayesian Approach

$$\omega_{21} e^{-\omega_{21} \hat{\mathbb{E}} L_1(z_1, \theta^{(2)}, \theta^{(sh)})}$$

$$\omega_{23} e^{-\omega_{23} \hat{\mathbb{E}} L_3(z_3, \theta^{(2)}, \theta^{(sh)})}$$

$$\omega_{22} e^{-\omega_{22} \hat{\mathbb{E}} L_2(z_2, \theta^{(2)}, \theta^{(sh)})}$$

Our Work: Bayesian Approach

$$\omega_{21}\omega_{22}\omega_{23}e^{-(\omega_{21}\hat{\mathbb{E}}L_1(z_1,\theta^{(2)},\theta^{(sh)})+\omega_{22}\hat{\mathbb{E}}L_2(z_2,\theta^{(2)},\theta^{(sh)})+\omega_{23}\hat{\mathbb{E}}L_3(z_3,\theta^{(2)},\theta^{(sh)}))}$$

- We refer to the above distribution as $Q(\theta^{(2)}, \theta^{(sh)})$ and the true joint distribution as $P(\theta^{(2)}, \theta^{(sh)})$

Our Work: Bayesian Approach

$$\omega_{21}\omega_{22}\omega_{23}e^{-(\omega_{21}\hat{\mathbb{E}}L_1(z_1,\theta^{(2)},\theta^{(sh)})+\omega_{22}\hat{\mathbb{E}}L_2(z_2,\theta^{(2)},\theta^{(sh)})+\omega_{23}\hat{\mathbb{E}}L_3(z_3,\theta^{(2)},\theta^{(sh)}))}$$

- We refer to the above distribution as $Q(\theta^{(2)}, \theta^{(sh)})$ and the true joint distribution as $P(\theta^{(2)}, \theta^{(sh)})$

Our Work: Bayesian Approach

Our Work: Bayesian Approach

$$\omega_{31} e^{-\omega_{31} \hat{\mathbb{E}} L_1(z_1, \theta^{(3)}, \theta^{(sh)})}$$

$$\omega_{33} e^{-\omega_{33} \hat{\mathbb{E}} L_3(z_3, \theta^{(3)}, \theta^{(sh)})}$$

$$\omega_{32} e^{-\omega_{32} \hat{\mathbb{E}} L_2(z_2, \theta^{(3)}, \theta^{(sh)})}$$

Our Work: Bayesian Approach

$$\omega_{31}\omega_{32}\omega_{33}e^{-(\omega_{31}\hat{\mathbb{E}}L_1(z_1,\theta^{(3)},\theta^{(sh)})+\omega_{32}\hat{\mathbb{E}}L_2(z_2,\theta^{(3)},\theta^{(sh)})+\omega_{33}\hat{\mathbb{E}}L_3(z_3,\theta^{(3)},\theta^{(sh)}))}$$

- We refer to the above distribution as $Q(\theta^{(3)}, \theta^{(sh)})$ and the true joint distribution as $P(\theta^{(3)}, \theta^{(sh)})$

Our Work: Bayesian Approach

$$\omega_{31}\omega_{32}\omega_{33}e^{-(\omega_{31}\hat{\mathbb{E}}L_1(z_1,\theta^{(3)},\theta^{(sh)})+\omega_{32}\hat{\mathbb{E}}L_2(z_2,\theta^{(3)},\theta^{(sh)})+\omega_{33}\hat{\mathbb{E}}L_3(z_3,\theta^{(3)},\theta^{(sh)}))}$$

- We refer to the above distribution as $Q(\theta^{(3)}, \theta^{(sh)})$ and the true joint distribution as $P(\theta^{(3)}, \theta^{(sh)})$

Our Work: Bayesian Approach

- Our objective:
  For a given $\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}$

$$\min_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \sum_{j=1}^{N} \log \omega_{ij} \tag{1}$$

- So our algorithm roughly is:
  1. Given shared neural network weights $\boldsymbol{\theta}^{(sh)}$, each edge-device can use its local data to solve $\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ to obtain the updated task-specific neural network weights
  2. Given the updated task-specific weights, the next step is to solve the optimization problem in (1) to find $\boldsymbol{\omega}_i^*$
  3. Given $\boldsymbol{\omega}_i^*$, the final step is to minimize $\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}^* \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ over $\boldsymbol{\theta}^{(sh)}$ to obtain the updated shared neural network weights
- Performance of the neural network obtained by solving the problem in (1) in comparison to the true optimization problem needs to be investigated

Our Work: Bayesian Approach

- Our objective:
  For a given $\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}$

$$\min_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \sum_{j=1}^{N} \log \omega_{ij} \tag{1}$$

- So our algorithm roughly is:
  1. Given shared neural network weights $\boldsymbol{\theta}^{(sh)}$, each edge-device can use its local data to solve $\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ to obtain the updated task-specific neural network weights
  2. Given the updated task-specific weights, the next step is to solve the optimization problem in (1) to find $\boldsymbol{\omega}_i^*$
  3. Given $\boldsymbol{\omega}_i^*$, the final step is to minimize $\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}^* \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ over $\boldsymbol{\theta}^{(sh)}$ to obtain the updated shared neural network weights
- Performance of the neural network obtained by solving the problem in (1) in comparison to the true optimization problem needs to be investigated

Our Work: Bayesian Approach

- Our objective:
  For a given $\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}$

$$\min_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \sum_{j=1}^{N} \log \omega_{ij} \tag{1}$$

- So our algorithm roughly is:
  1. Given shared neural network weights $\boldsymbol{\theta}^{(sh)}$, each edge-device can use its local data to solve $\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ to obtain the updated task-specific neural network weights
  2. Given the updated task-specific weights, the next step is to solve the optimization problem in (1) to find $\boldsymbol{\omega}_i^*$
  3. Given $\boldsymbol{\omega}_i^*$, the final step is to minimize $\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}^* \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ over $\boldsymbol{\theta}^{(sh)}$ to obtain the updated shared neural network weights
- Performance of the neural network obtained by solving the problem in (1) in comparison to the true optimization problem needs to be investigated

Our Work: Bayesian Approach

- Our objective:
  For a given $\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}$

$$\min_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \sum_{j=1}^{N} \log \omega_{ij} \tag{1}$$

- So our algorithm roughly is:
  1. Given shared neural network weights $\boldsymbol{\theta}^{(sh)}$, each edge-device can use its local data to solve $\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ to obtain the updated task-specific neural network weights
  2. Given the updated task-specific weights, the next step is to solve the optimization problem in (1) to find $\boldsymbol{\omega}_i^*$
  3. Given $\boldsymbol{\omega}_i^*$, the final step is to minimize $\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}^* \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ over $\boldsymbol{\theta}^{(sh)}$ to obtain the updated shared neural network weights
- Performance of the neural network obtained by solving the problem in (1) in comparison to the true optimization problem needs to be investigated

Our Work: Bayesian Approach

- Our objective:
  For a given $\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}$

$$\min_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \sum_{j=1}^{N} \log \omega_{ij} \tag{1}$$

- So our algorithm roughly is:
  1. Given shared neural network weights $\boldsymbol{\theta}^{(sh)}$, each edge-device can use its local data to solve $\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ to obtain the updated task-specific neural network weights
  2. Given the updated task-specific weights, the next step is to solve the optimization problem in (1) to find $\boldsymbol{\omega}_i^*$
  3. Given $\boldsymbol{\omega}_i^*$, the final step is to minimize $\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}^* \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ over $\boldsymbol{\theta}^{(sh)}$ to obtain the updated shared neural network weights
- Performance of the neural network obtained by solving the problem in (1) in comparison to the true optimization problem needs to be investigated

Our Work: Bayesian Approach

- Our objective:
  For a given $\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}$

$$\min_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \sum_{j=1}^{N} \log \omega_{ij} \tag{1}$$

- So our algorithm roughly is:
  1. Given shared neural network weights $\boldsymbol{\theta}^{(sh)}$, each edge-device can use its local data to solve $\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ to obtain the updated task-specific neural network weights
  2. Given the updated task-specific weights, the next step is to solve the optimization problem in (1) to find $\boldsymbol{\omega}_i^*$
  3. Given $\boldsymbol{\omega}_i^*$, the final step is to minimize $\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}^* \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$ over $\boldsymbol{\theta}^{(sh)}$ to obtain the updated shared neural network weights
- Performance of the neural network obtained by solving the problem in (1) in comparison to the true optimization problem needs to be investigated

Our Work: Theoretical Guarantees

Definition: $\log - \exp$ complexity of the neural network

$$\mathcal{R}_i(\boldsymbol{\theta}) := \log \mathbb{E}_Q \sup_{\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}} \frac{\exp\left\{\mathbb{E}_{z \sim \mathcal{D}_i} L_i(z, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})\right\}}{\Pi_{i=1}^{N} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})}, \; i = \{1, \cdots, N\}$$

Our Work: Theoretical Guarantees

Theorem

For a given neural network $\boldsymbol{\theta}$, and the $\log - \exp$ complexity, the following bound holds with a probability of at least $1-\delta$, $(\delta > 0)$

$$\inf_{\boldsymbol{\theta}} \mathbb{E}_{z_i \sim \mathcal{D}_i}\{L_i(z_i, \boldsymbol{\theta})\} \leq \inf_{\boldsymbol{\theta}^{(sh)}} \Bigg[\mathtt{Obj}_i(\boldsymbol{\theta}^{(sh)}) + \mathcal{R}_i(\boldsymbol{\theta})$$

$$+ \sup_{\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}, \boldsymbol{\omega}_i} \mathtt{KL}(Q||P) + l_{max} \sqrt{\sum_{j=1}^{N} \frac{\omega_{ij}^2}{2n_j^2} \log\left(\frac{1}{\delta}\right)} - N \Bigg],$$

where $\mathtt{KL}(Q||P)$ is the KL-divergence between the distributions Q and P,

$\mathtt{Obj}_i(\boldsymbol{\theta}^{(sh)}) := \inf_{\boldsymbol{\omega}_i} \sum_{j=1}^{N} \left[\omega_{ij} \inf_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \log \omega_{ij} \right]$

Our Work: Omni-Fedge

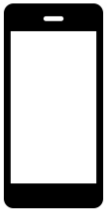
Algorithm 1: Omni-Fedge

1 `Omni-Fedge():`
2 INITIALIZE $\boldsymbol{\theta}_{sh}$ and BROADCAST (BC) to all nodes.
3 **for** $t \in \{1, 2, \ldots, \ldots\}$ **do**
4 **for** $i = 1, 2, \ldots, N$ **do**
5 $\boldsymbol{\theta}_t^{(i)} = \arg\min_{\boldsymbol{\theta}^{(i)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)})$
6 Each device i BCs $\boldsymbol{\theta}_t^{(i)}$ to all other nodes $l = 1, 2, \ldots, N$ through FS.
7 COMPUTE AND SEND $\hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(j)}, \boldsymbol{\theta}^{(sh)})$ to all nodes.
8 `Minimize-Objective()`
9 to get $\boldsymbol{\omega}_i$ for all i.
10 At each node, COMPUTE $\sum_{j=1}^{N} \omega_{ji}^* \nabla_{\boldsymbol{\theta}_t^{(sh)}} \hat{\mathbb{E}} L_i(z_i, \boldsymbol{\theta}^{(j)}, \boldsymbol{\theta}^{(sh)})$ and BC it to all nodes through FS.
11 Perform GRADIENT UPDATE $\boldsymbol{\theta}_{t+1}^{(sh)} := \boldsymbol{\theta}_t^{(sh)} - \eta^{com} \gamma_t^{(i)}$, where $\gamma_t^{(i)} := \frac{1}{N}\left(\sum_{l=1}^{N} \sum_{j=1}^{N} \omega_{jl}^* \nabla_{\boldsymbol{\theta}_t^{(sh)}} \hat{\mathbb{E}} L_l(z_l, \boldsymbol{\theta}^{(j)}, \boldsymbol{\theta}^{(sh)})\right)$
12 GO TO step 3.
13 `Minimize-Objective():`
14 COMPUTE $\boldsymbol{\omega}_i^* = \arg\min_{\boldsymbol{\omega}_i} \left(\sum_{j=1}^{N} \omega_{ij} \hat{\mathbb{E}} L_j(z_j, \boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(sh)}) - \log \prod_{j=1}^{N} \omega_{ij}\right)$

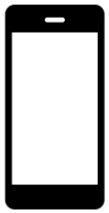
Figure: Omni-Fedge Algorithm

Our Work: Omni-Fedge

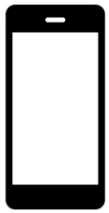
$$\hat{\theta}^{(1)} = arg \min_{\theta^{(1)}} \hat{\mathbb{E}} L_1(z_1, \theta^{(1)}, \theta^{(sh)}))$$



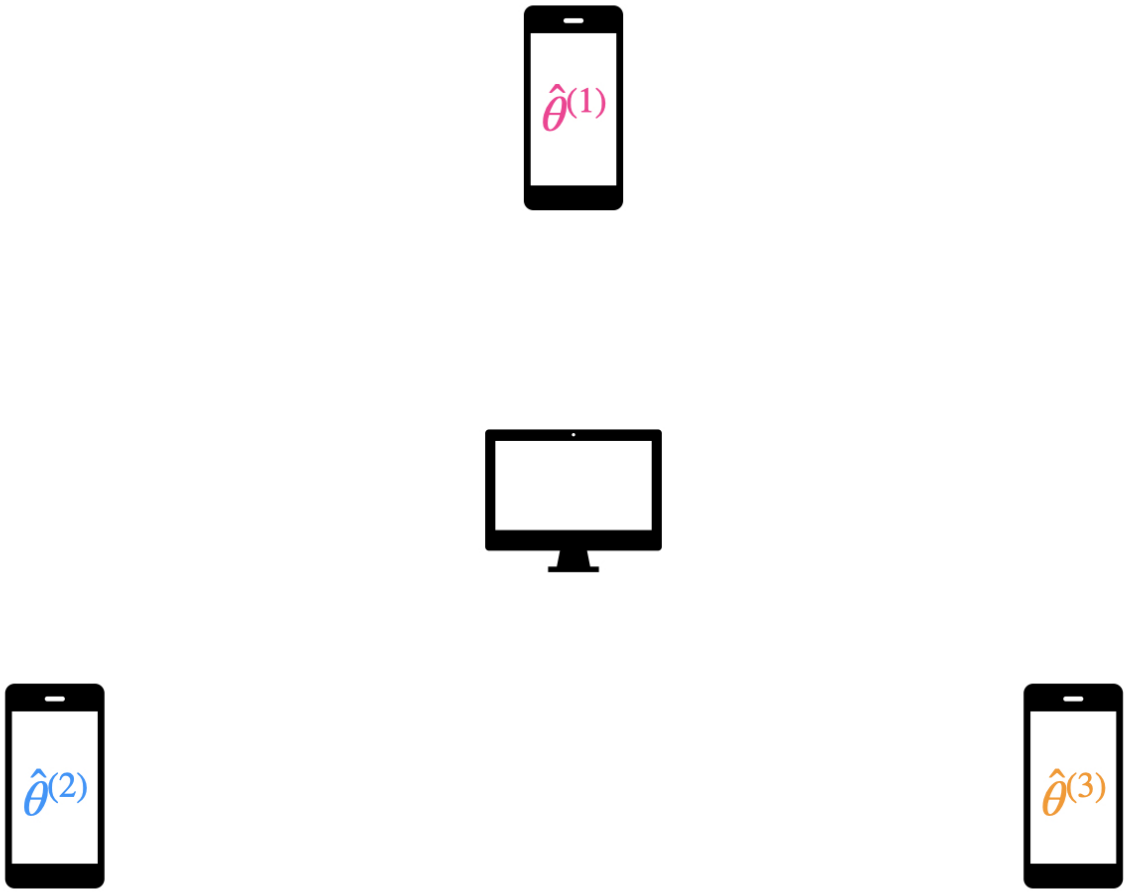
$$\hat{\theta}^{(2)} = arg \min_{\theta^{(2)}} \hat{\mathbb{E}} L_2(z_2, \theta^{(2)}, \theta^{(sh)}))$$



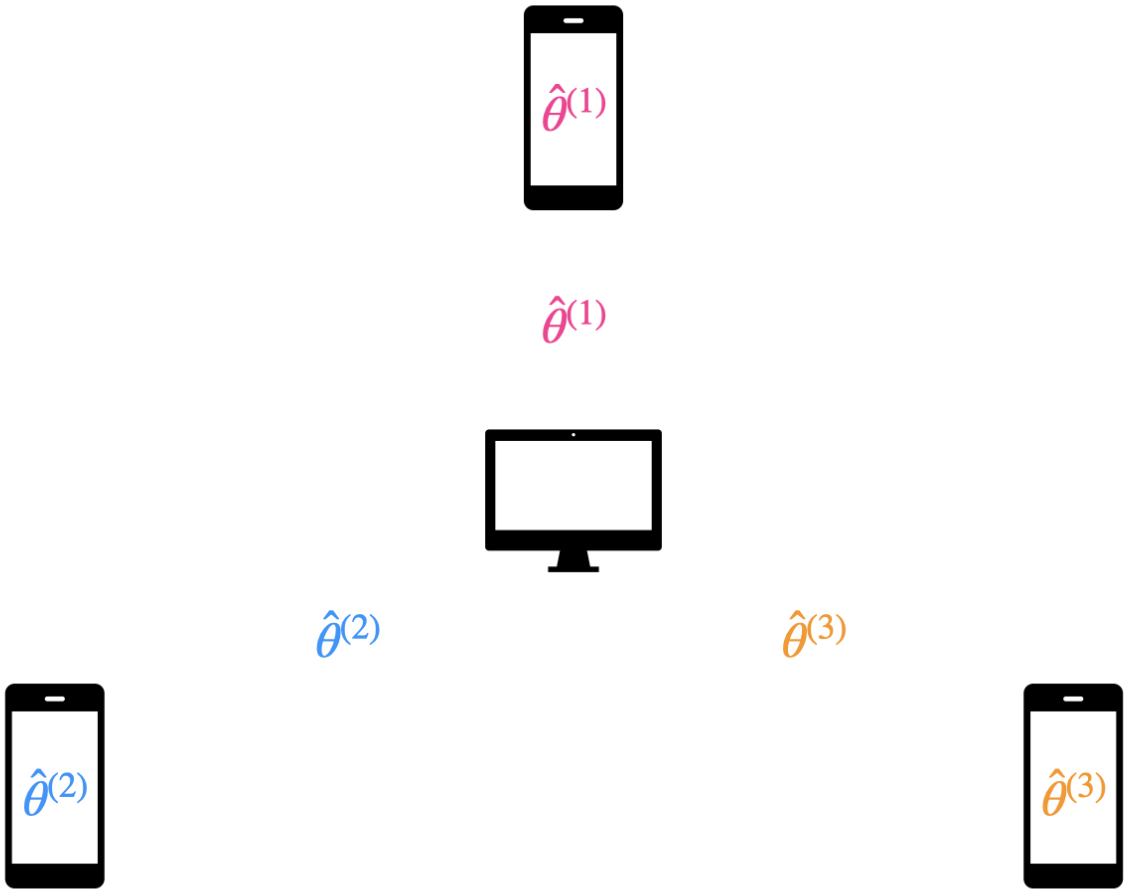
$$\hat{\theta}^{(3)} = arg \min_{\theta^{(3)}} \hat{\mathbb{E}} L_3(z_3, \theta^{(3)}, \theta^{(sh)}))$$



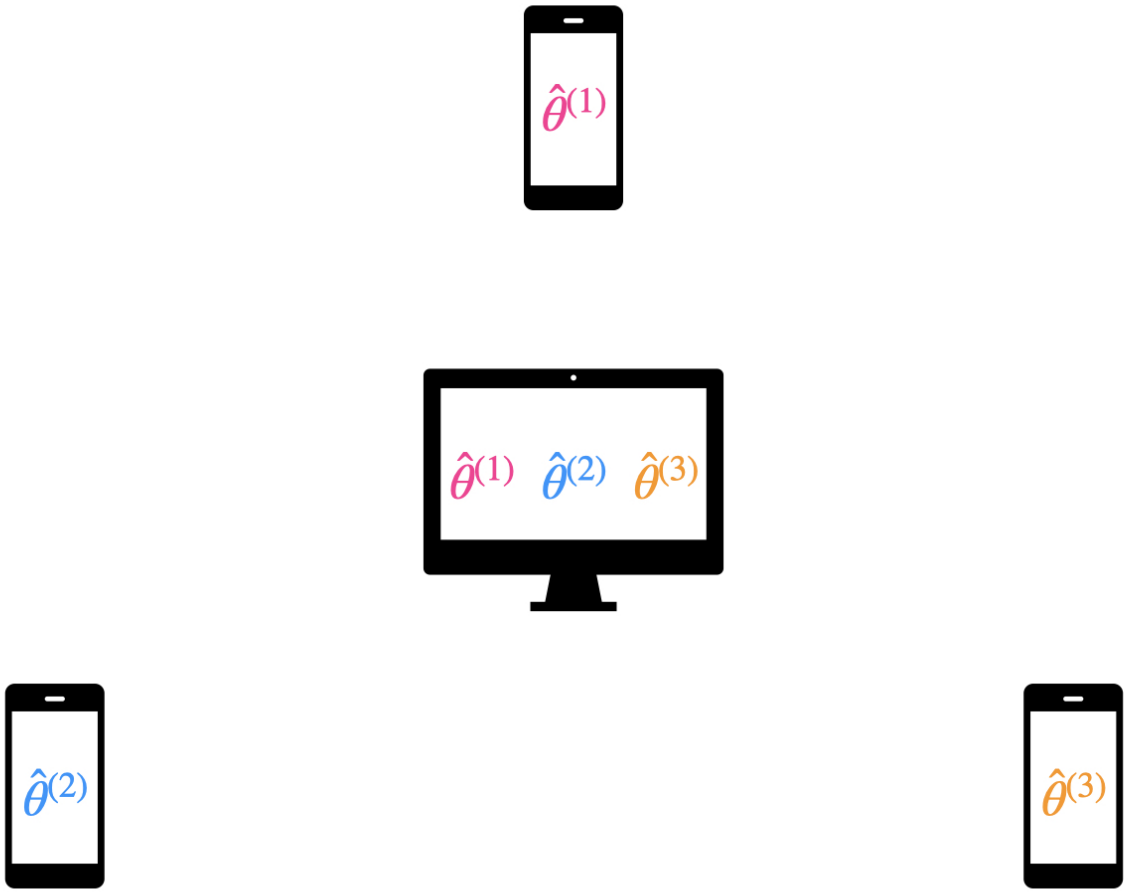
Our Work: Omni-Fedge



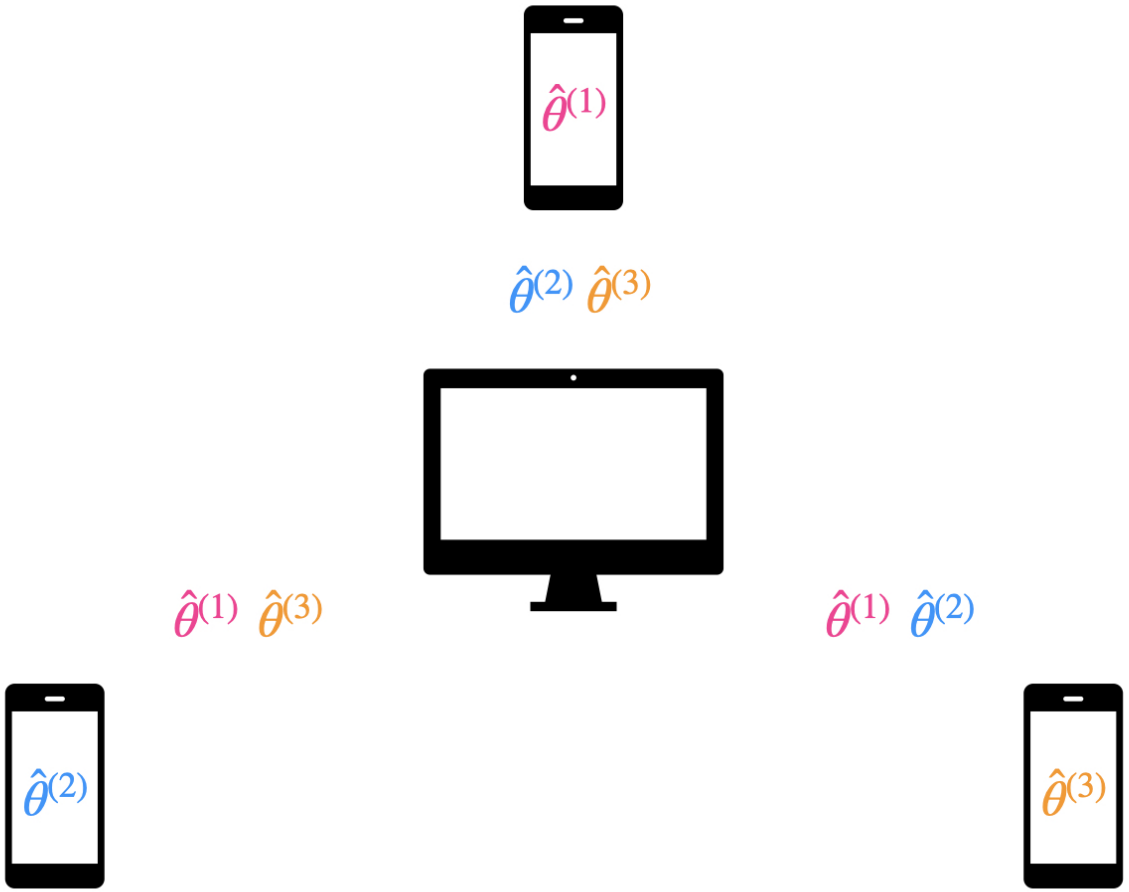
Our Work: Omni-Fedge



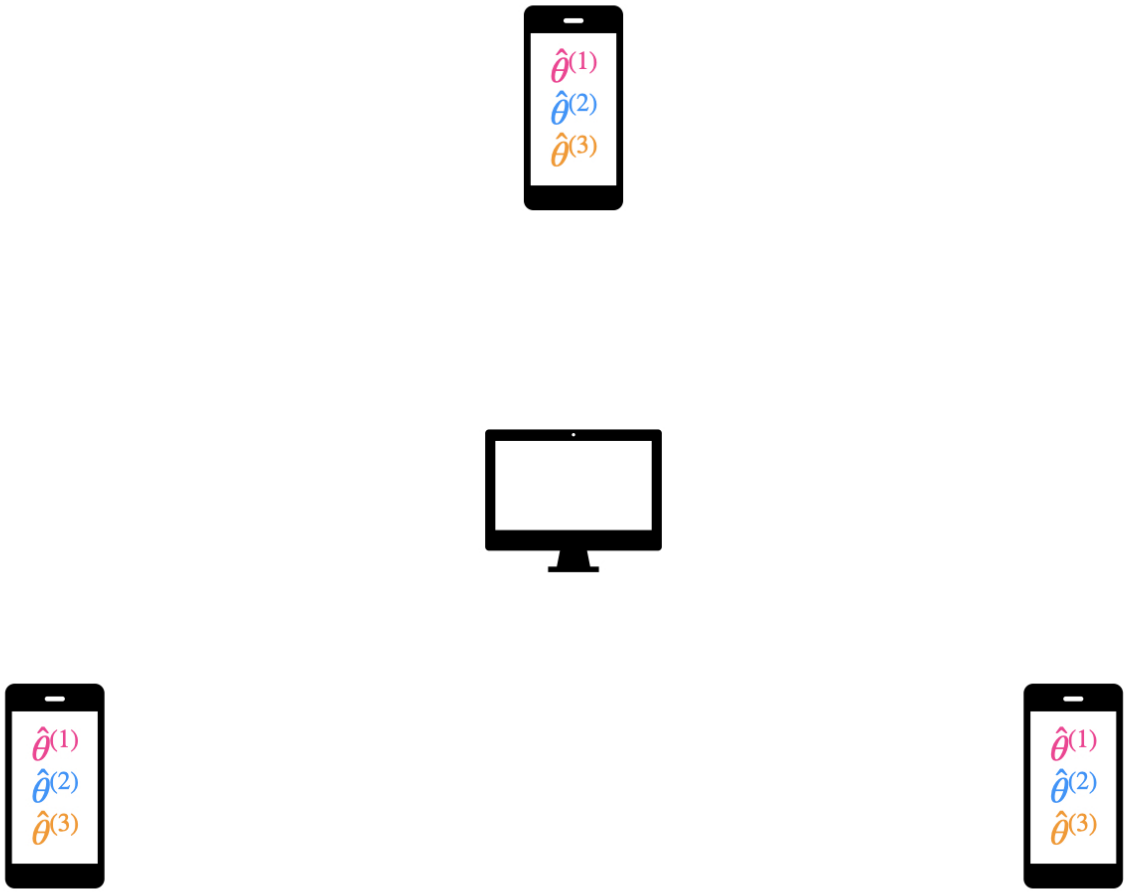
Our Work: Omni-Fedge



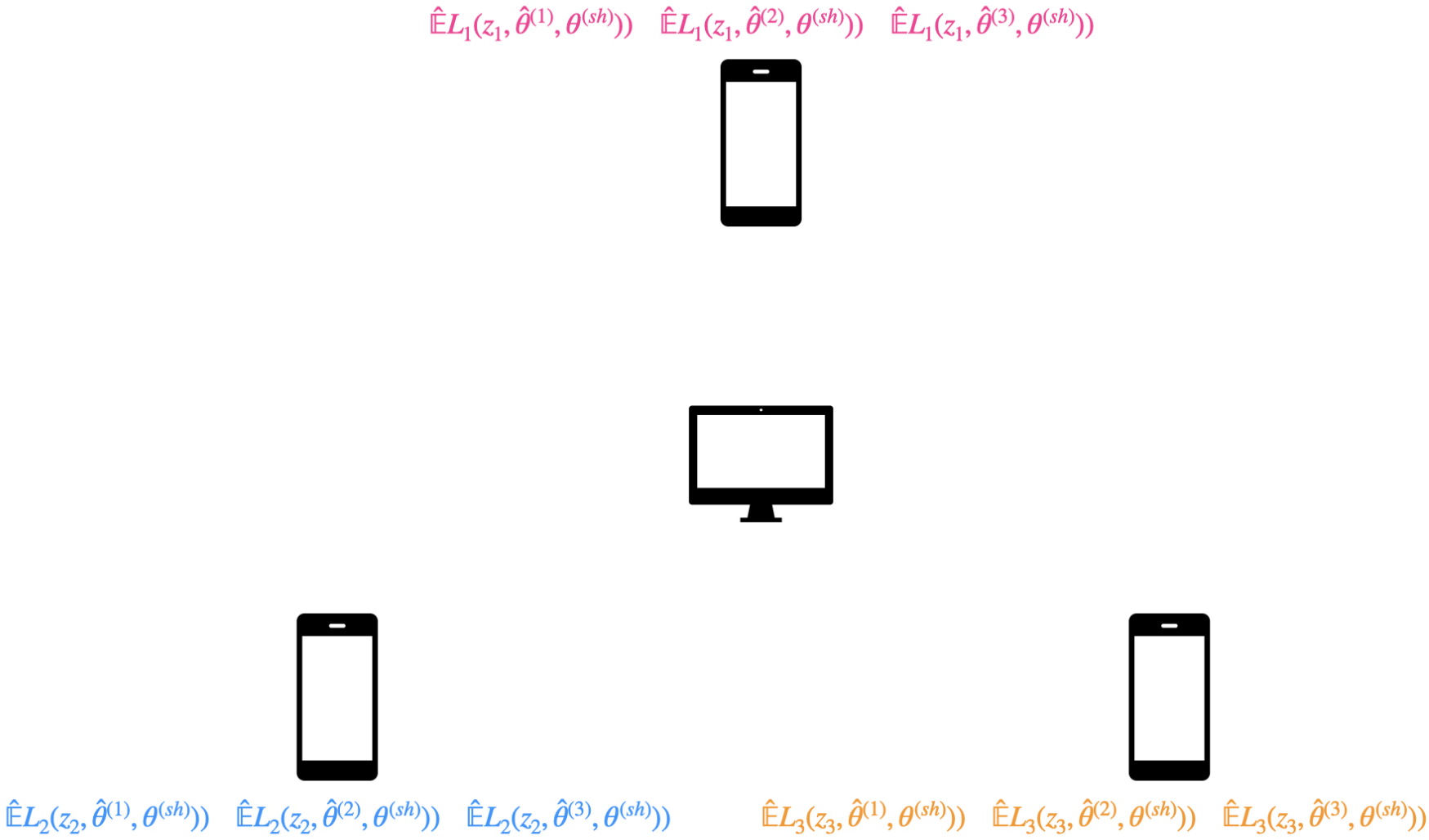
Our Work: Omni-Fedge



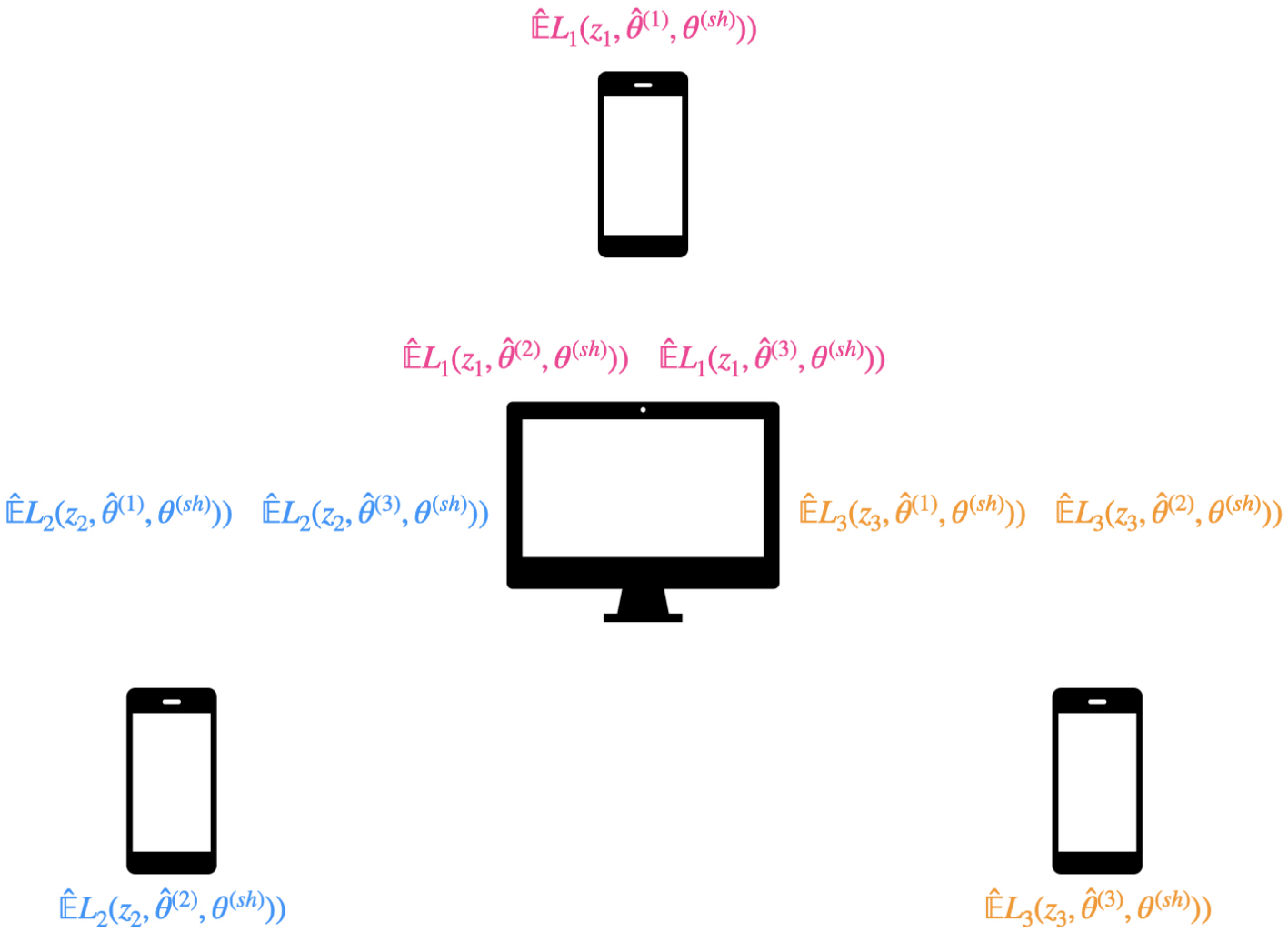
Our Work: Omni-Fedge



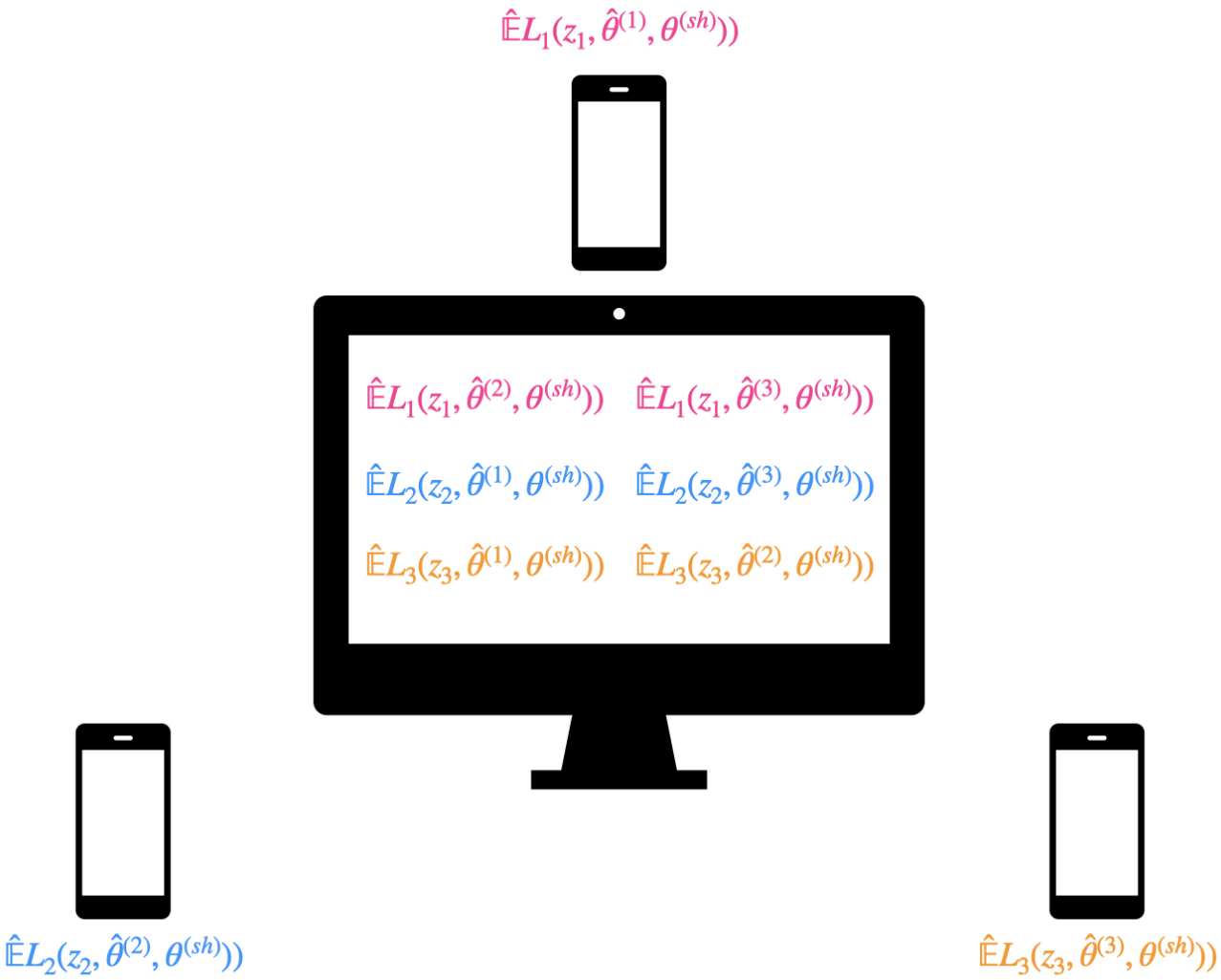
Our Work: Omni-Fedge



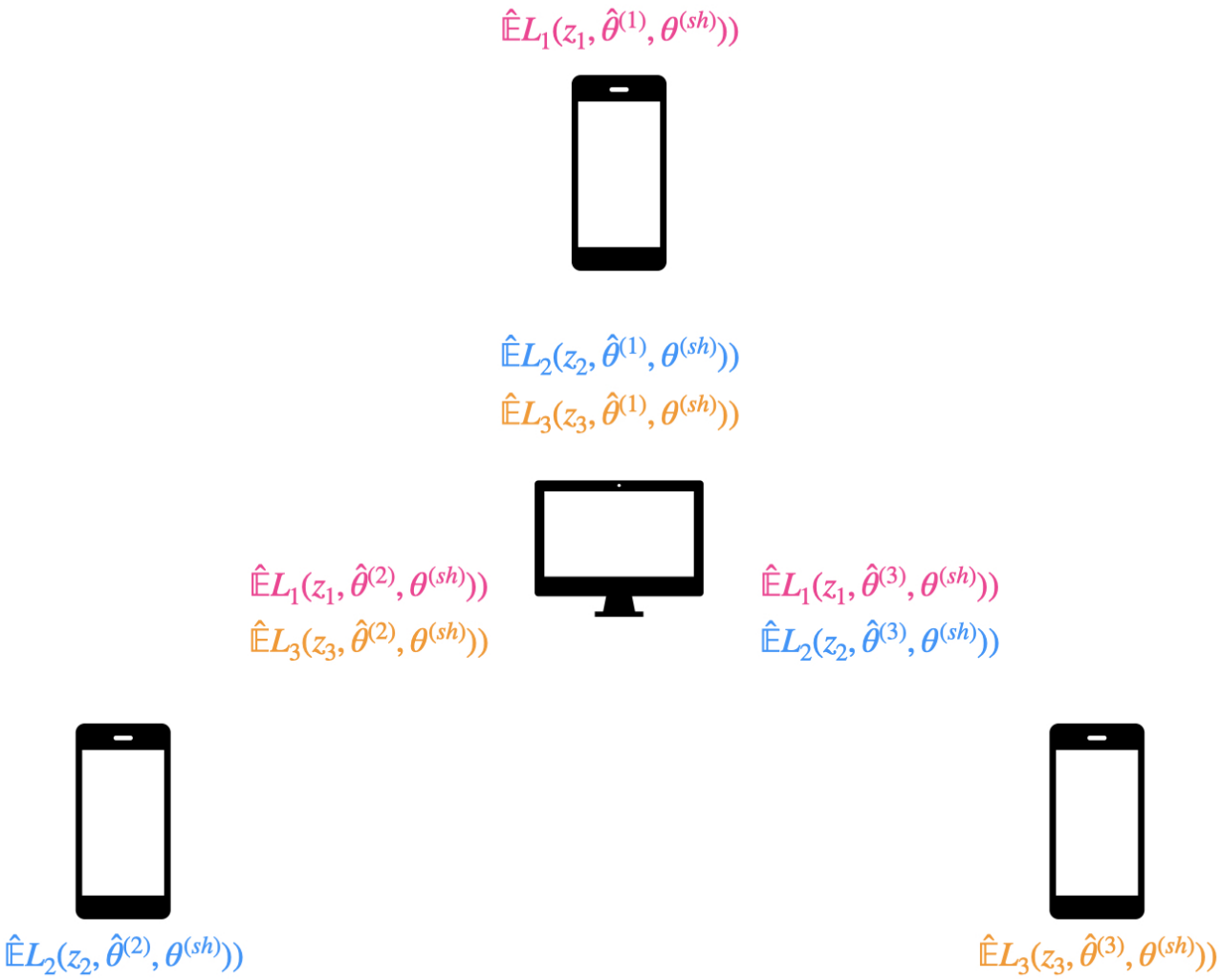
Our Work: Omni-Fedge



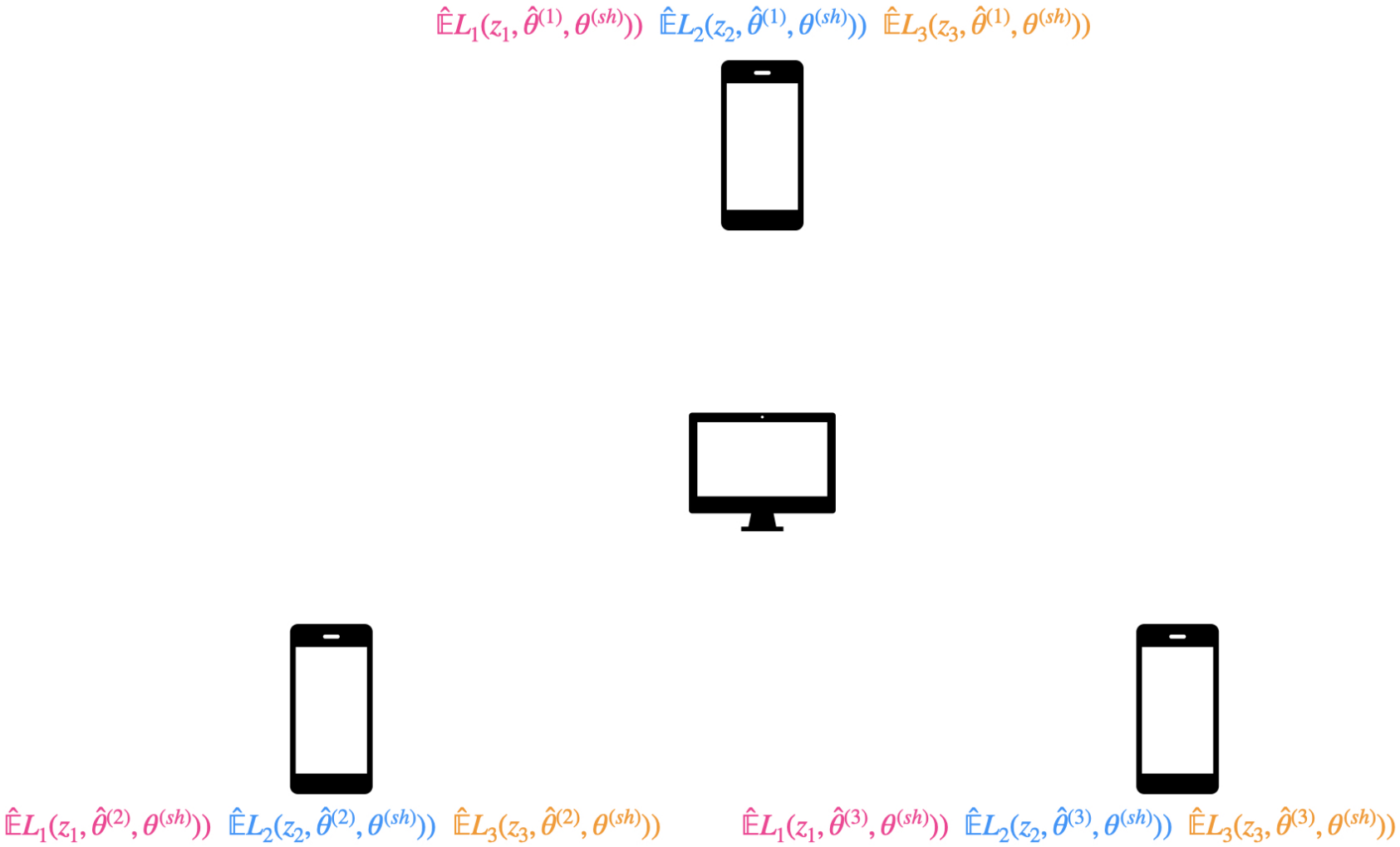
Our Work: Omni-Fedge



Our Work: Omni-Fedge

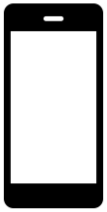


Our Work: Omni-Fedge

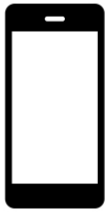


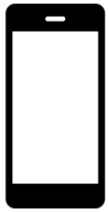
Our Work: Omni-Fedge

$$\omega_1^* = arg \min_{\omega_{11},\omega_{12},\omega_{13}} \left(\omega_{11}\hat{\mathbb{E}}L_1(z_1,\hat{\theta}^{(1)},\theta^{(sh)})) + \omega_{12}\hat{\mathbb{E}}L_2(z_2,\hat{\theta}^{(1)},\theta^{(sh)}) + \omega_{13}\hat{\mathbb{E}}L_3(z_3,\hat{\theta}^{(1)},\theta^{(sh)})) - \log \omega_{11}\omega_{12}\omega_{13} \right)$$



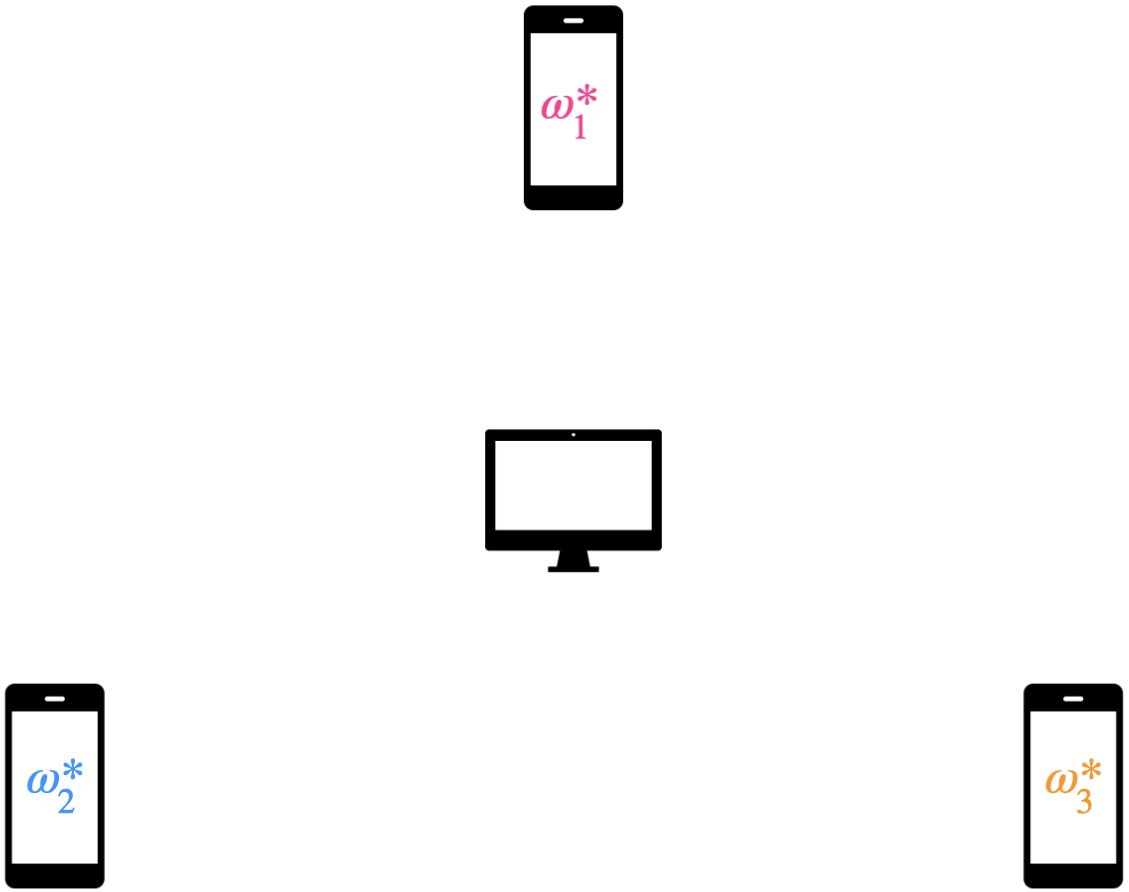
$$\omega_2^* = arg \min_{\omega_{21},\omega_{22},\omega_{23}} \left(\omega_{21}\hat{\mathbb{E}}L_1(z_1,\hat{\theta}^{(2)},\theta^{(sh)})) + \omega_{22}\hat{\mathbb{E}}L_2(z_2,\hat{\theta}^{(2)},\theta^{(sh)}) + \omega_{23}\hat{\mathbb{E}}L_3(z_3,\hat{\theta}^{(2)},\theta^{(sh)})) - \log \omega_{21}\omega_{22}\omega_{23} \right)$$



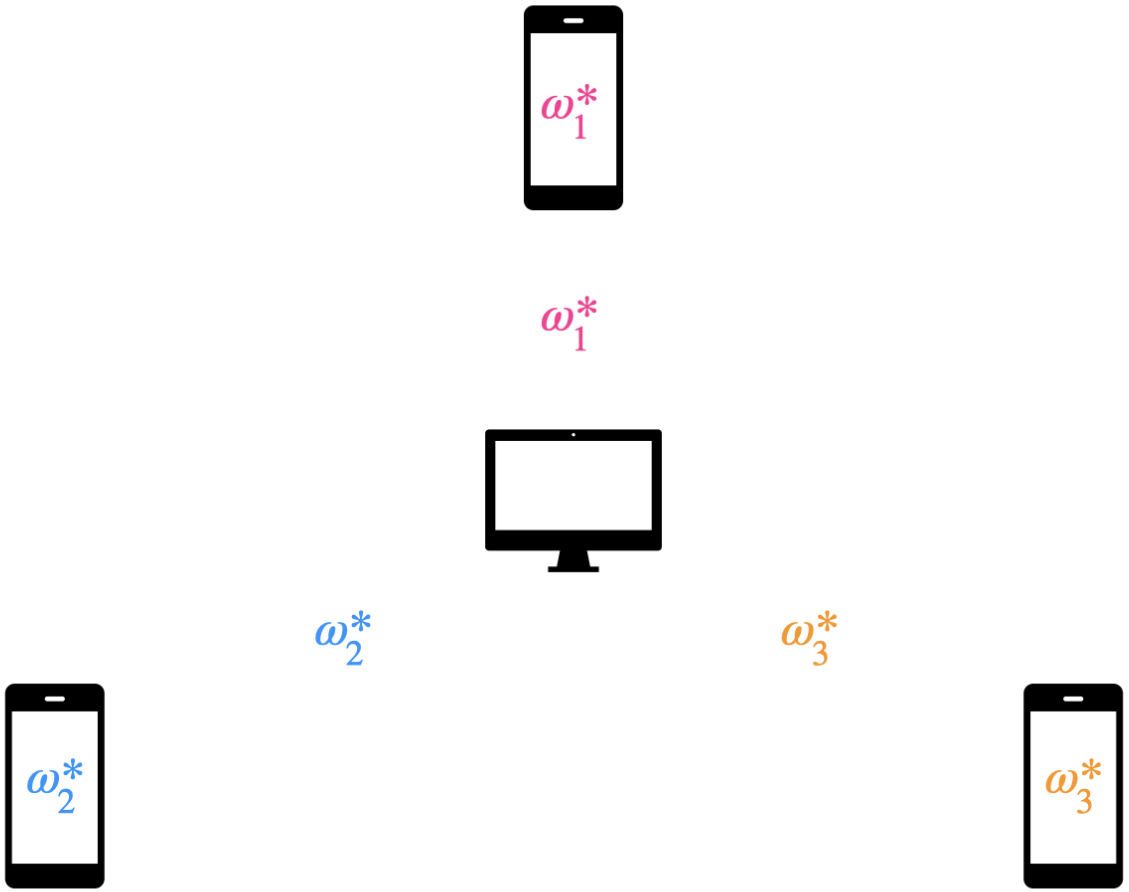


$$\omega_3^* = arg \min_{\omega_{31},\omega_{32},\omega_{33}} \left(\omega_{31}\hat{\mathbb{E}}L_1(z_1,\hat{\theta}^{(3)},\theta^{(sh)})) + \omega_{32}\hat{\mathbb{E}}L_2(z_2,\hat{\theta}^{(3)},\theta^{(sh)}) + \omega_{33}\hat{\mathbb{E}}L_3(z_3,\hat{\theta}^{(3)},\theta^{(sh)})) - \log \omega_{31}\omega_{32}\omega_{33} \right)$$

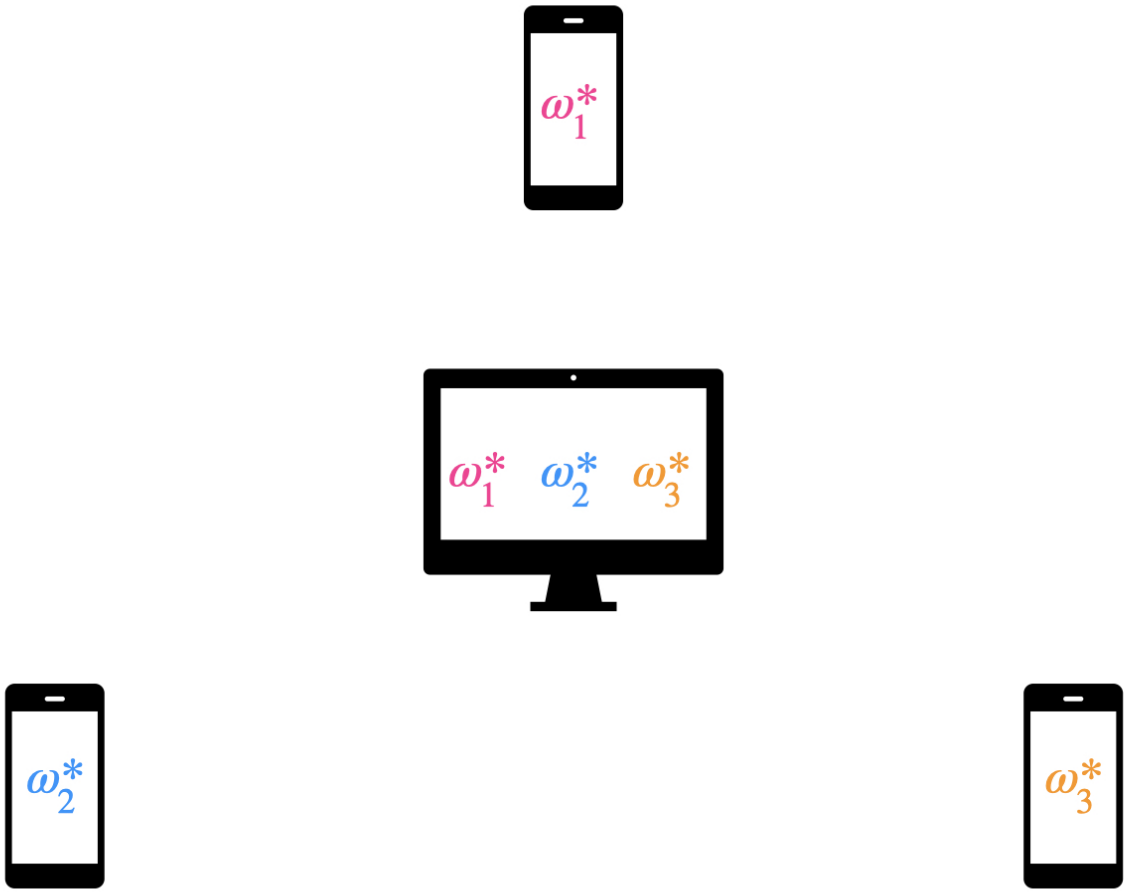
Our Work: Omni-Fedge



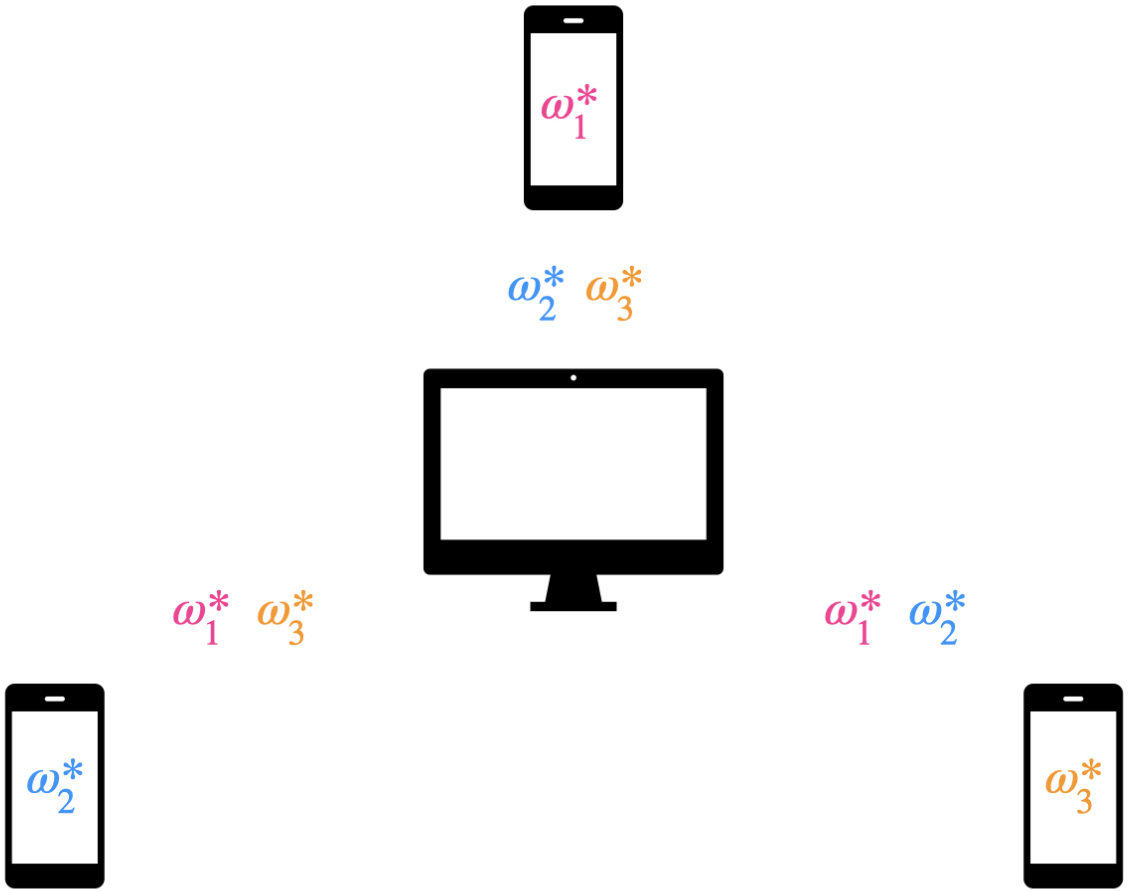
Our Work: Omni-Fedge



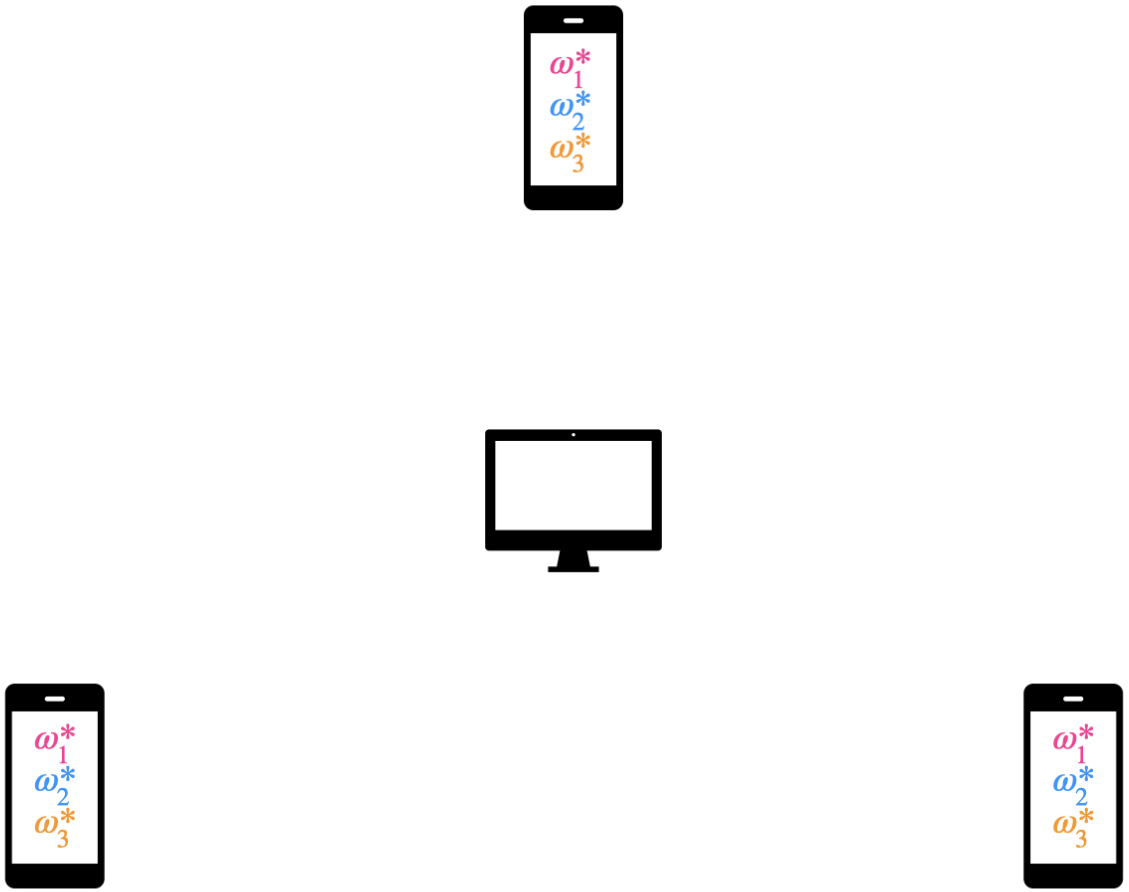
Our Work: Omni-Fedge



Our Work: Omni-Fedge

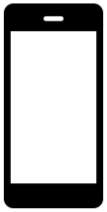


Our Work: Omni-Fedge

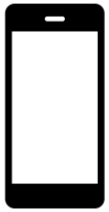


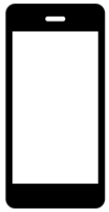
Our Work: Omni-Fedge

$$\alpha_1 = \omega_{11}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_1(z_1, \hat{\theta}^{(1)}, \theta^{(sh)})) + \omega_{21}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_1(z_1, \hat{\theta}^{(2)}, \theta^{(sh)}) + \omega_{31}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_1(z_1, \hat{\theta}^{(3)}, \theta^{(sh)}))$$



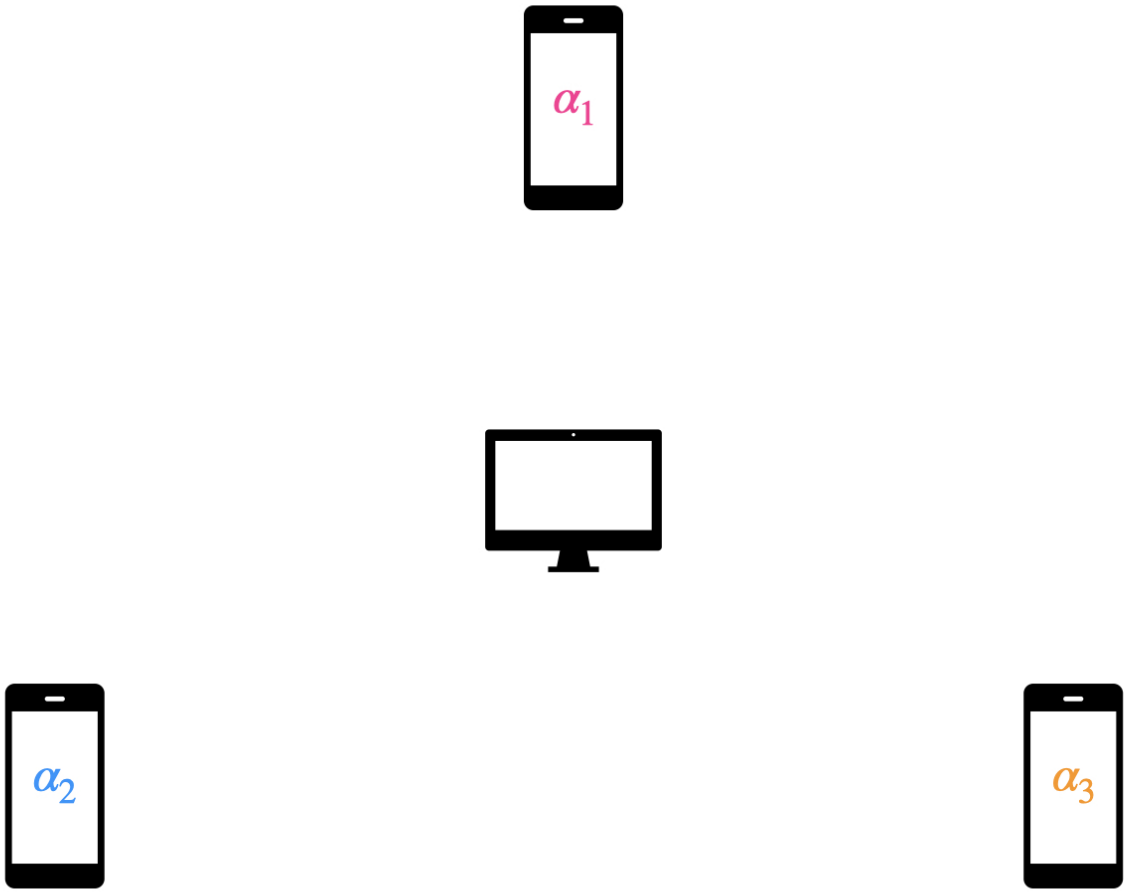
$$\alpha_2 = \omega_{12}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_2(z_2, \hat{\theta}^{(1)}, \theta^{(sh)})) + \omega_{22}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_2(z_2, \hat{\theta}^{(2)}, \theta^{(sh)}) + \omega_{32}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_2(z_2, \hat{\theta}^{(3)}, \theta^{(sh)}))$$



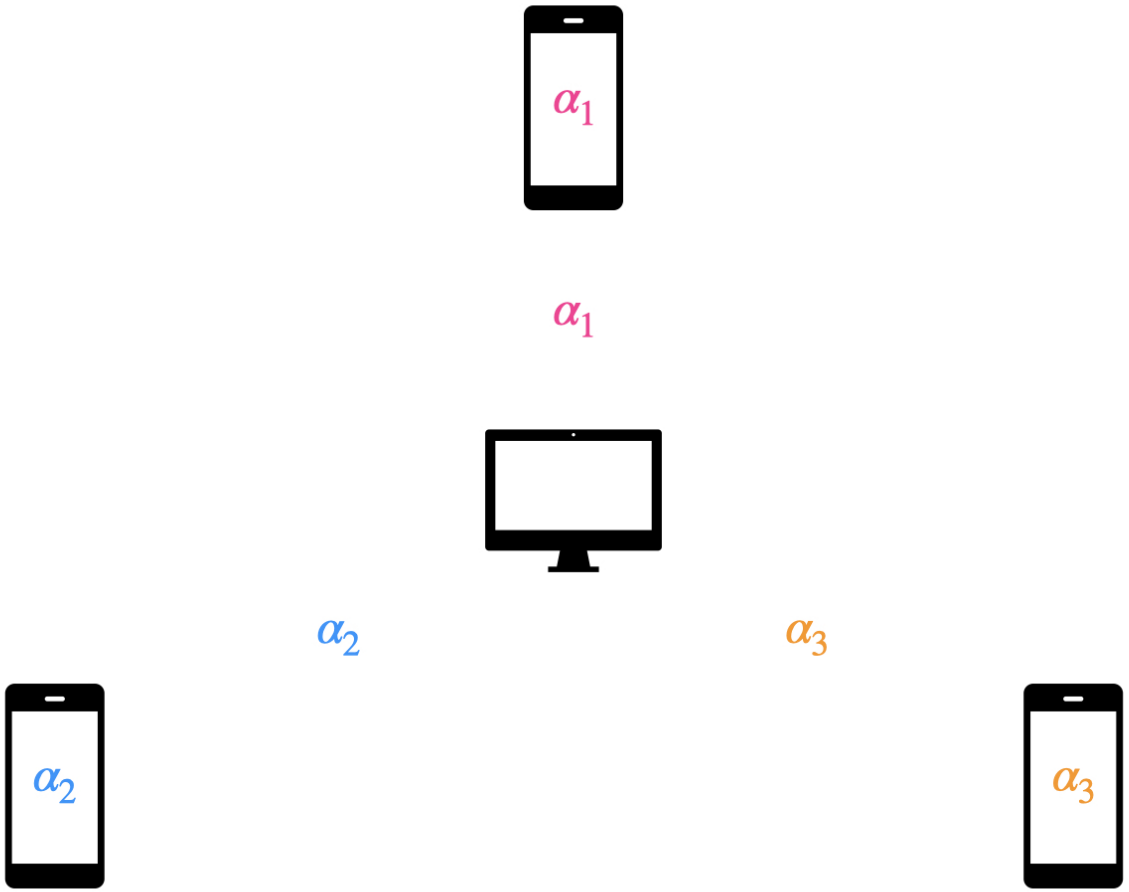


$$\alpha_3 = \omega_{13}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_3(z_3, \hat{\theta}^{(1)}, \theta^{(sh)})) + \omega_{23}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_3(z_3, \hat{\theta}^{(2)}, \theta^{(sh)}) + \omega_{33}^* \nabla_{\theta^{(sh)}} \hat{\mathbb{E}} L_3(z_3, \hat{\theta}^{(3)}, \theta^{(sh)}))$$

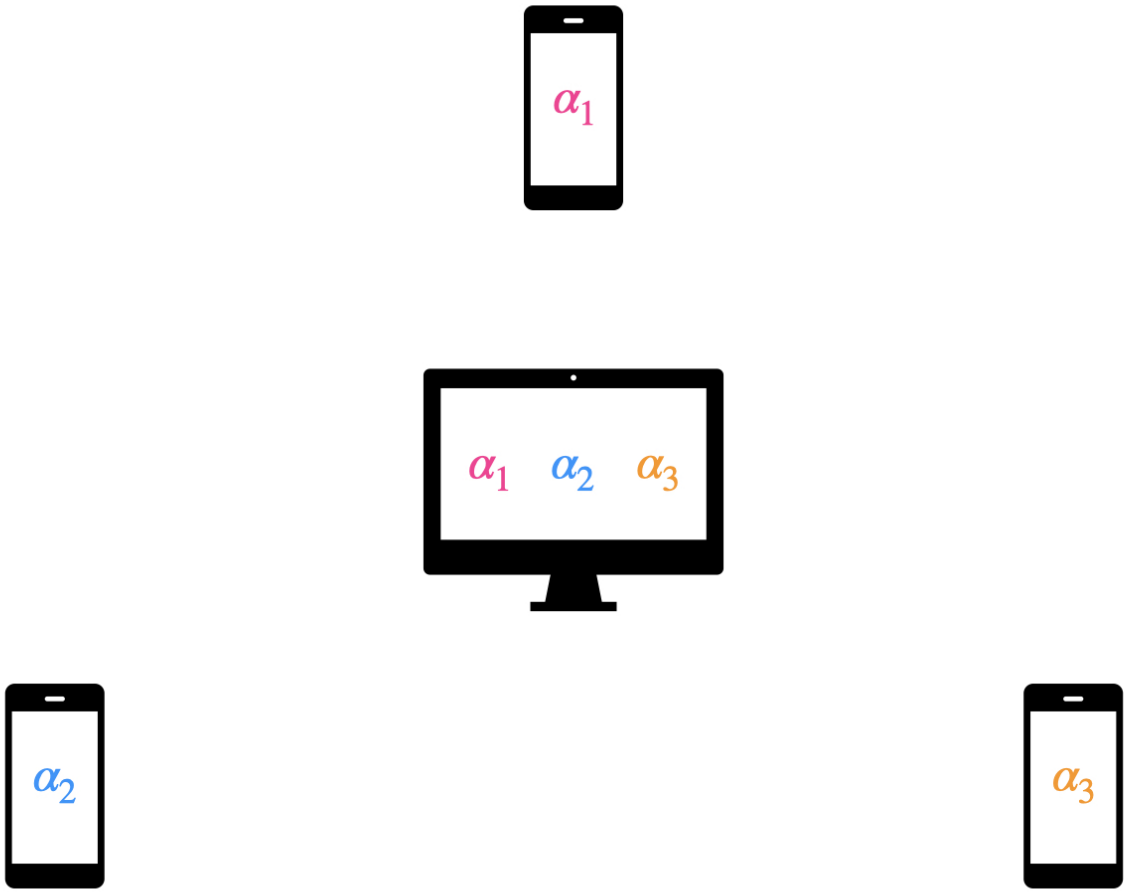
Our Work: Omni-Fedge



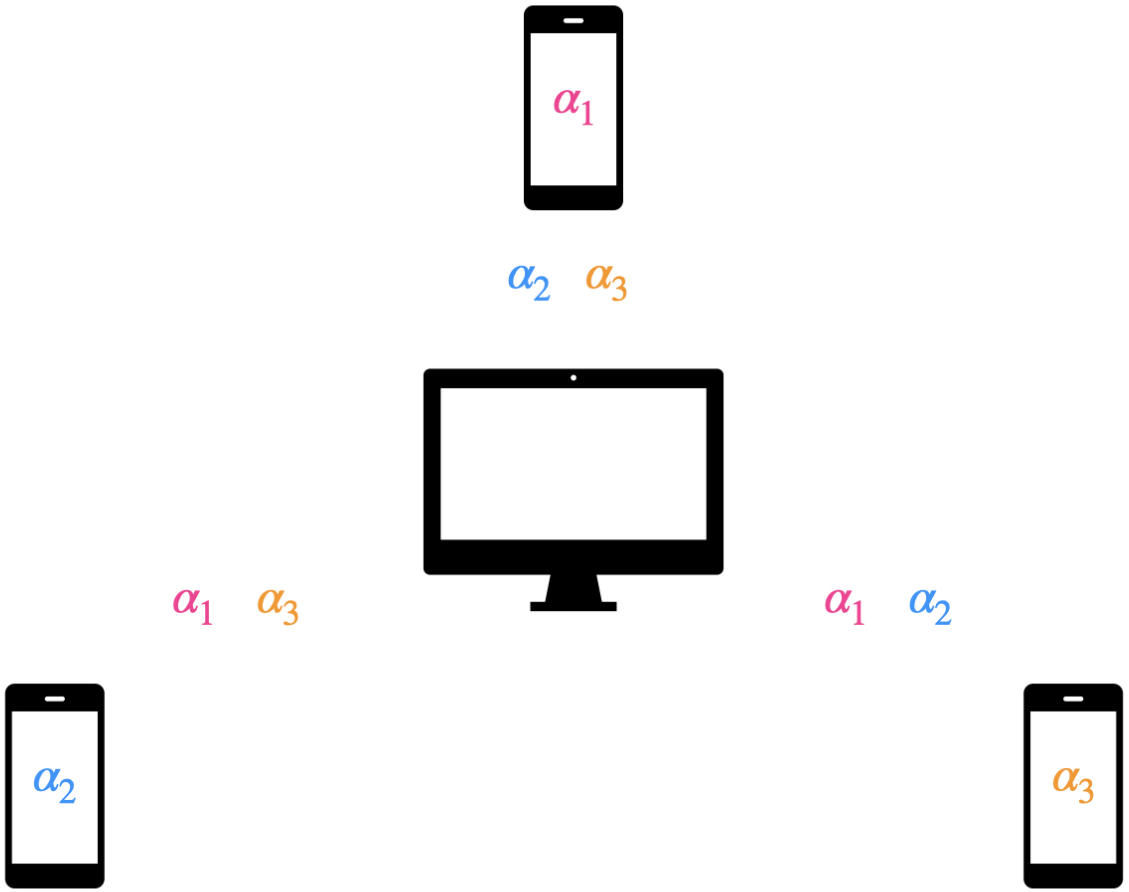
Our Work: Omni-Fedge



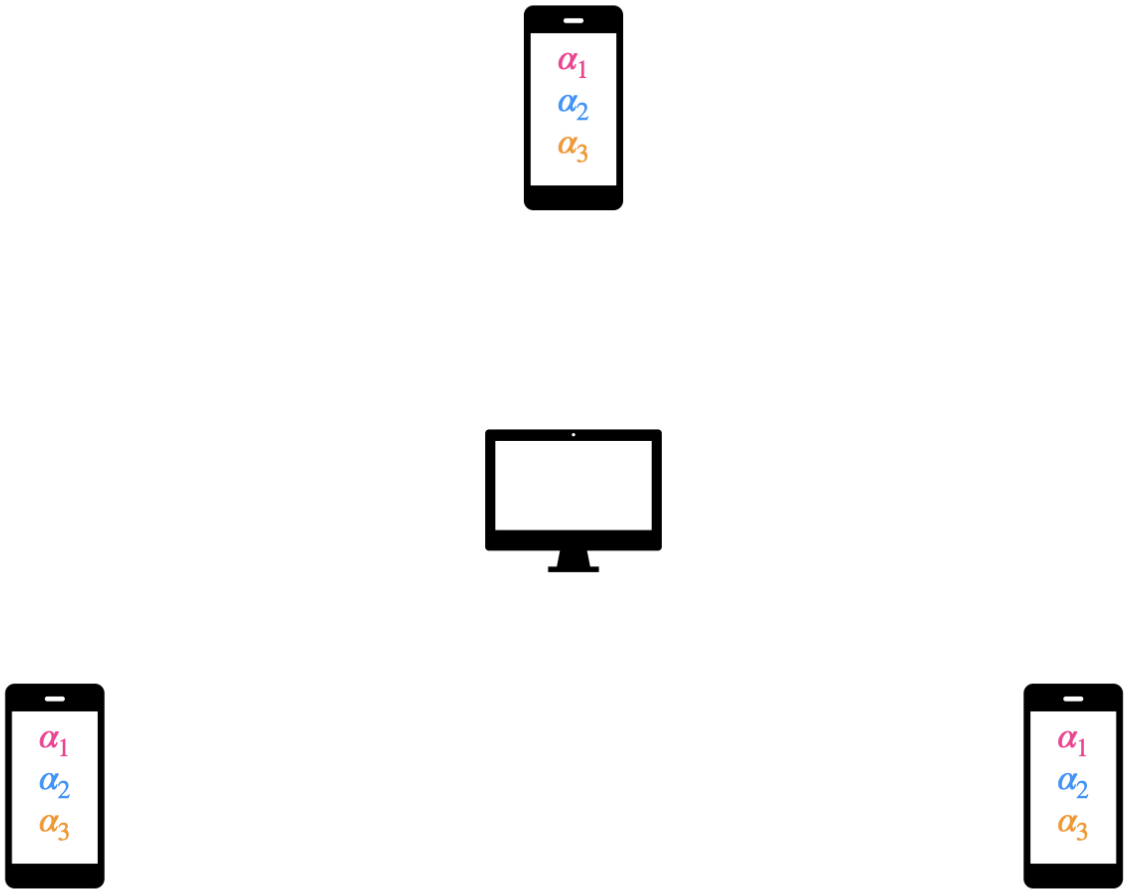
Our Work: Omni-Fedge



Our Work: Omni-Fedge

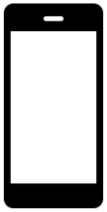


Our Work: Omni-Fedge

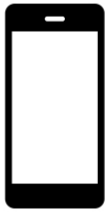


Our Work: Omni-Fedge

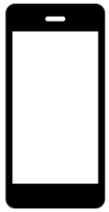
$$\theta_{t+1}^{(sh)} = \theta_t^{(sh)} - \eta_{com}\frac{(\alpha_1 + \alpha_2 + \alpha_3)}{3}$$



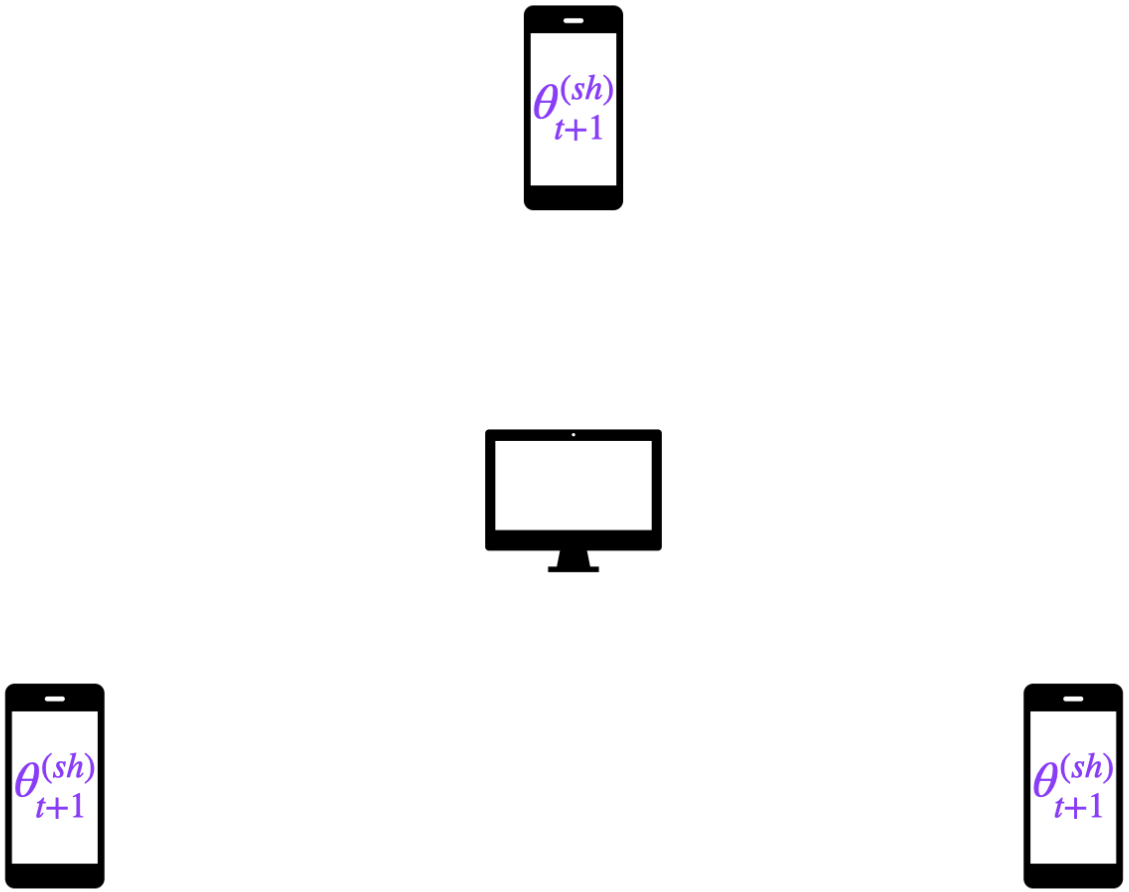
$$\theta_{t+1}^{(sh)} = \theta_t^{(sh)} - \eta_{com}\frac{(\alpha_1 + \alpha_2 + \alpha_3)}{3}$$



$$\theta_{t+1}^{(sh)} = \theta_t^{(sh)} - \eta_{com}\frac{(\alpha_1 + \alpha_2 + \alpha_3)}{3}$$



Our Work: Omni-Fedge



Our Work: Experimental Results with Omni-Fedge

Type of Data	Training samples per node	Testing samples per node
MNIST I.I.D.	622	778
MNIST non-I.I.D.	673	842
FMNIST I.I.D.	622	778
FMNIST non-I.I.D.	331	414

Our Work: Experimental Results with Omni-Fedge

Type of Data	Training samples per node	Testing samples per node
MNIST I.I.D.	622	778
MNIST non-I.I.D.	673	842
FMNIST I.I.D.	622	778
FMNIST non-I.I.D.	331	414

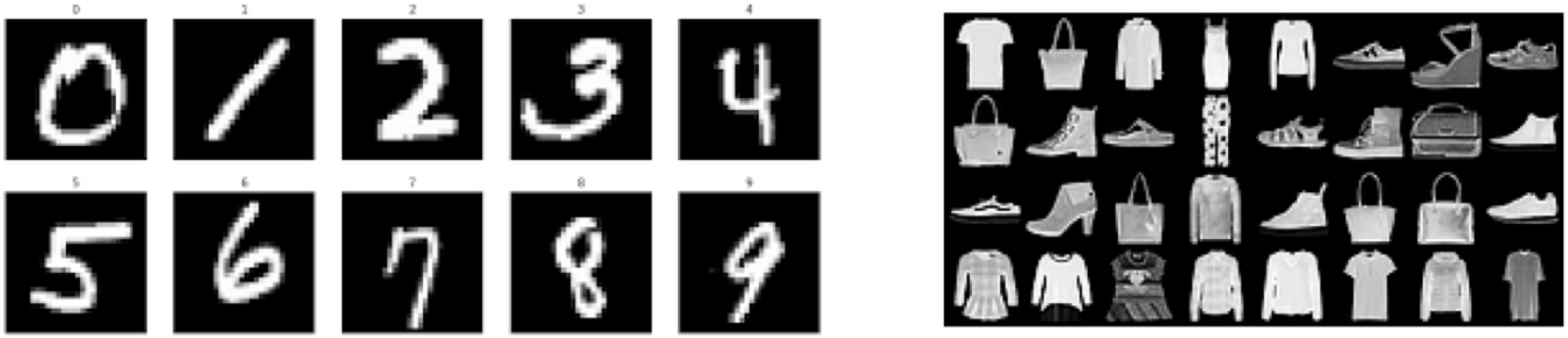


Figure: Sample MNIST and FMNIST images

Our Work: Experimental Results with Omni-Fedge

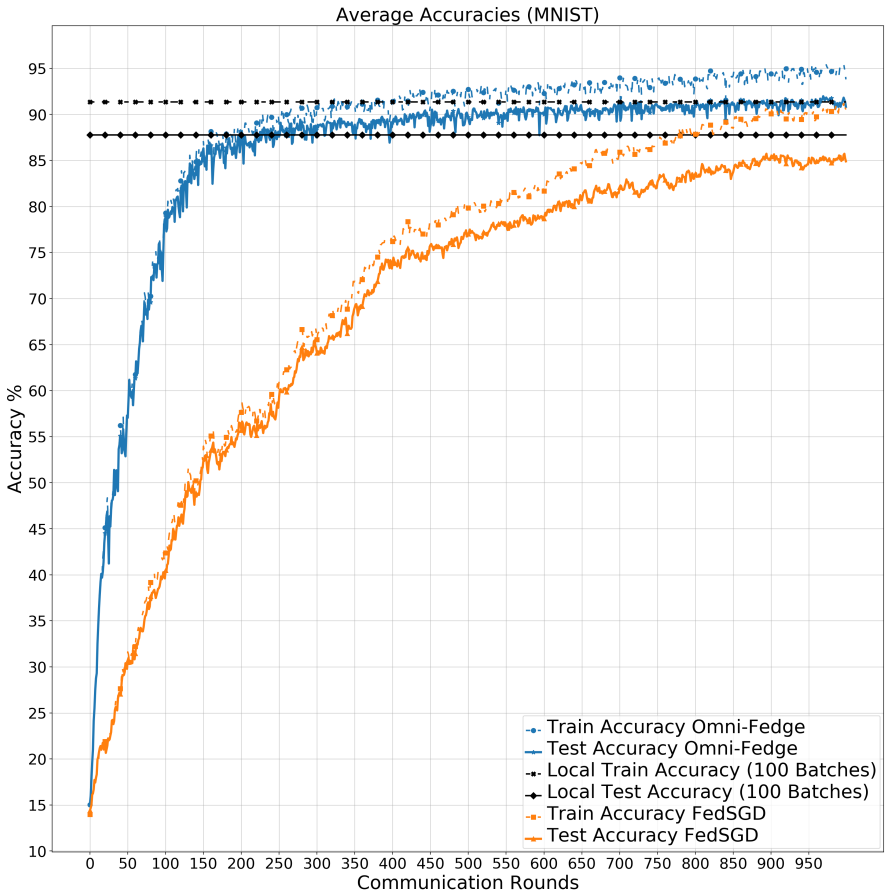


Figure: Results for non-IID data using MNIST

Our Work: Experimental Results with Omni-Fedge

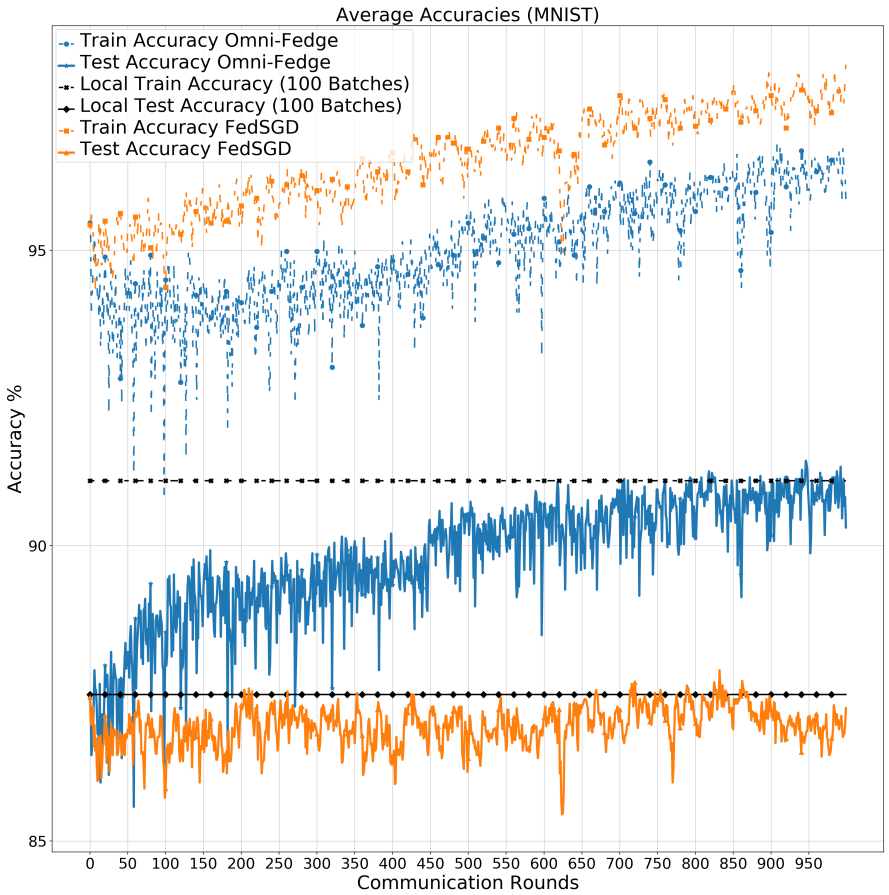


Figure: Results for IID data using MNIST

Our Work: Experimental Results with Omni-Fedge

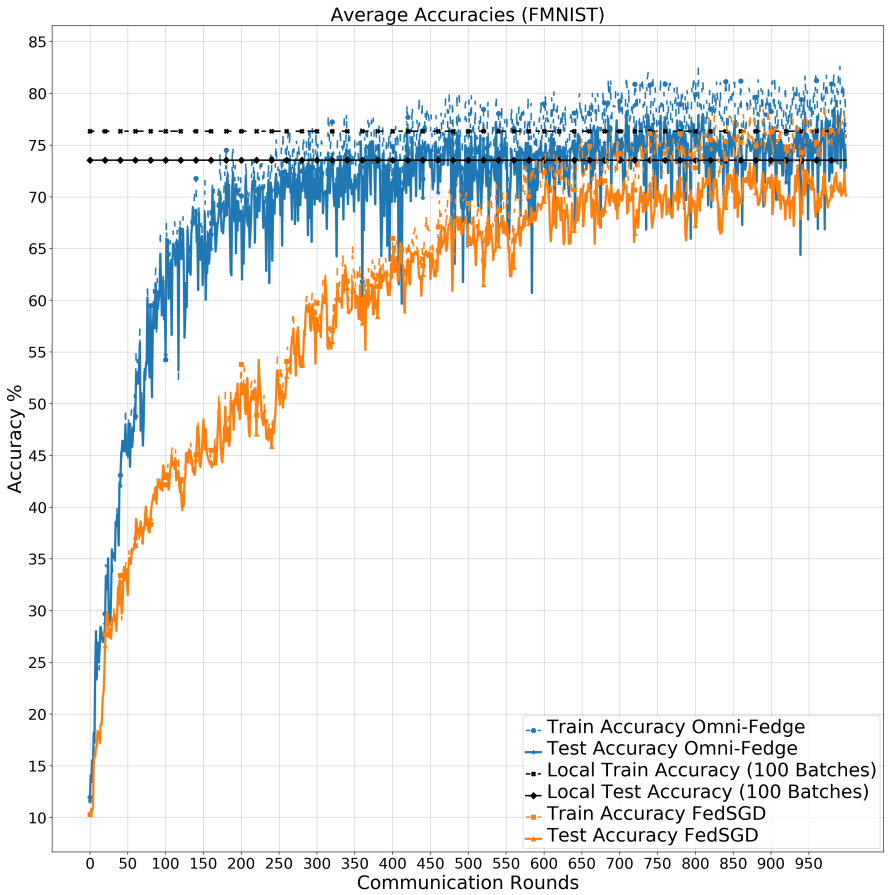


Figure: Results for non-IID data using FMNIST

Our Work: Experimental Results with Omni-Fedge

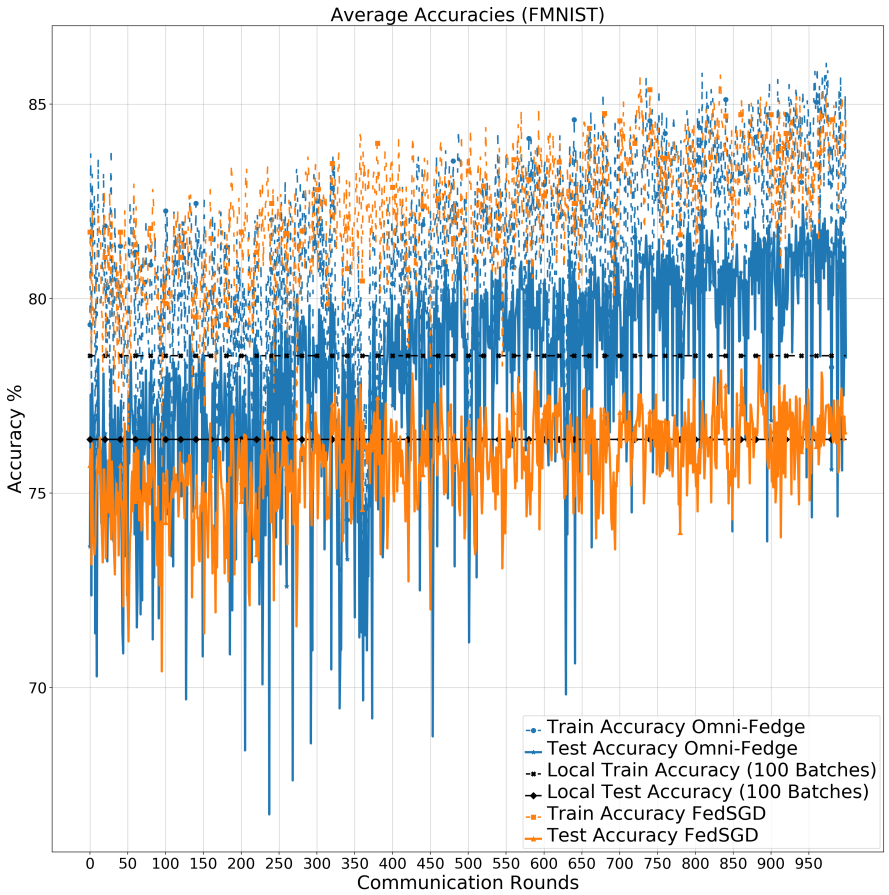


Figure: Results for IID data using FMNIST

Our Work: Experimental Results with Omni-Fedge

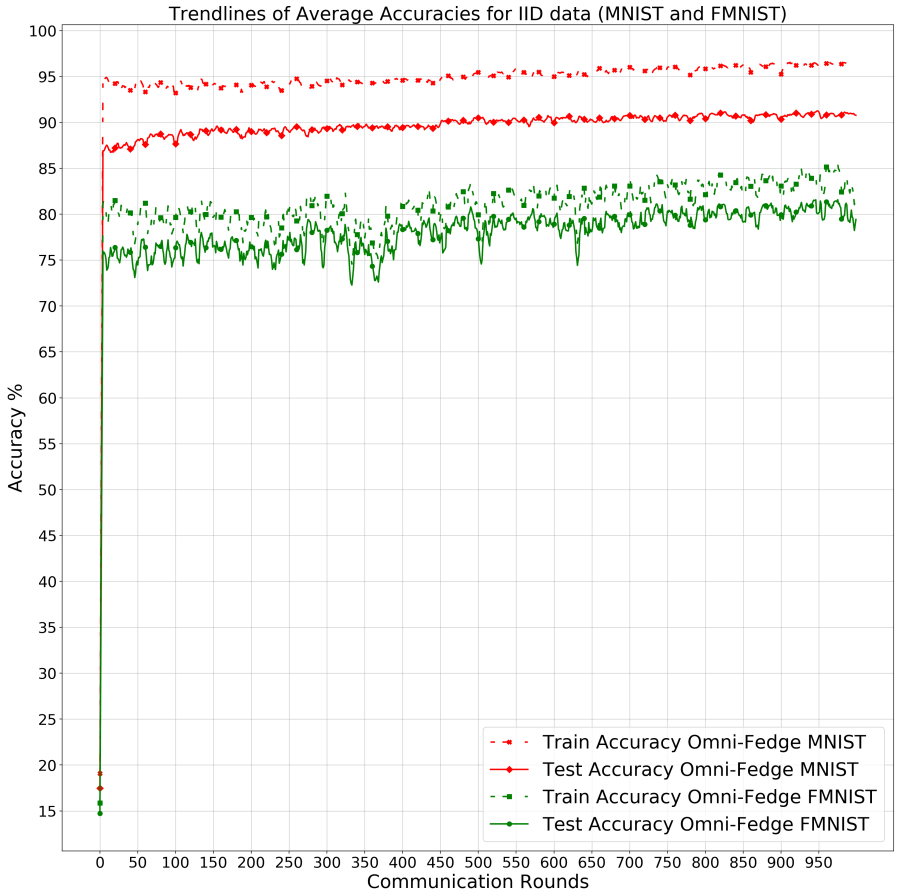


Figure: Trendlines of Average Accuracies for IID data using MNIST and FMNIST

Our Work: Experimental Results with Omni-Fedge

- Omni-Fedge outperforms local and FedSGD for both I.I.D. and non-I.I.D. scenarios
- It assigns higher weights for more relevant nodes, and assigns equal weights across all the nodes under I.I.D. case

Our Work: Experimental Results with Omni-Fedge

- Omni-Fedge outperforms local and FedSGD for both I.I.D. and non-I.I.D. scenarios
- It assigns higher weights for more relevant nodes, and assigns equal weights across all the nodes under I.I.D. case

Our Work: Conclusion

- Considered a specific FL problem under a non-I.I.D. data setting
- Provided a sound theoretical framework for the proposed algorithm based on a novel Bayesian approach
- Introduced a new complexity measure as a consequence of the PAC bound

Our Work: Conclusion

- Considered a specific FL problem under a non-I.I.D. data setting
- Provided a sound theoretical framework for the proposed algorithm based on a novel Bayesian approach
- Introduced a new complexity measure as a consequence of the PAC bound

Our Work: Conclusion

- Considered a specific FL problem under a non-I.I.D. data setting
- Provided a sound theoretical framework for the proposed algorithm based on a novel Bayesian approach
- Introduced a new complexity measure as a consequence of the PAC bound

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

Our Work: Current status and future

- Submitted a paper titled *"Federated Algorithm With Bayesian Approach: Omni-Fedge"* to ICASSP 2021, whose result is awaited
- Working on a journal paper in this regard
- Analysing the following:
  1. Stragglers
  2. $\log - \exp$ complexity
  3. Convergence
  4. Efficient communication

References

[1] Keith Bonawitz et al. "Towards federated learning at scale: System design". In: *arXiv preprint arXiv:1902.01046* (2019).

[2] Sebastian Caldas et al. "Expanding the reach of federated learning by reducing client resource requirements". In: *arXiv preprint arXiv:1812.07210* (2018).

[3] Meng Hao et al. "Efficient and privacy-enhanced federated learning for industrial artificial intelligence". In: *IEEE Transactions on Industrial Informatics* 16.10 (2019), pp. 6532–6542.

[4] Andrew Hard et al. "Federated learning for mobile keyboard prediction". In: *arXiv preprint arXiv:1811.03604* (2018).

[5] Alex Kendall, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7482–7491.

[6] Jakub Konečný et al. "Federated optimization: Distributed machine learning for on-device intelligence". In: *arXiv preprint arXiv:1610.02527* (2016).

References (cont.)

[7] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[8] Tian Li et al. "Federated learning: Challenges, methods, and future directions". In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60.

[9] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. "Agnostic federated learning". In: *arXiv preprint arXiv:1902.00146* (2019).

[10] Solmaz Niknam, Harpreet S Dhillon, and Jeffrey H Reed. "Federated learning for wireless communications: Motivation, opportunities, and challenges". In: *IEEE Communications Magazine* 58.6 (2020), pp. 46–51.

[11] Sebastian Ruder. "An overview of multi-task learning in deep neural networks". In: *arXiv preprint arXiv:1706.05098* (2017).

[12] Felix Sattler et al. "Robust and communication-efficient federated learning from non-iid data". In: *IEEE transactions on neural networks and learning systems* (2019).

References (cont.)

[13] Ozan Sener and Vladlen Koltun. "Multi-task learning as multi-objective optimization". In: *Advances in Neural Information Processing Systems*. 2018, pp. 527–538.

[14] Changjian Shui et al. "A principled approach for learning task similarity in multitask learning". In: *arXiv preprint arXiv:1903.09109* (2019).

[15] Virginia Smith et al. "Federated multi-task learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4424–4434.

[16] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].

[17] Timothy Yang et al. "Applied federated learning: Improving google keyboard query suggestions". In: *arXiv preprint arXiv:1812.02903* (2018).

[18] Yue Zhao et al. "Federated learning with non-iid data". In: *arXiv preprint arXiv:1806.00582* (2018).

Thank you!