FedALA: Adaptive Local Aggregation for Personalized Federated Learning

Jianqing Zhang[1] Yang Hua[2] Hao Wang[3]

Tao Song[1] Zhengui Xue[1] Ruhui Ma[1] Haibing Guan[1]

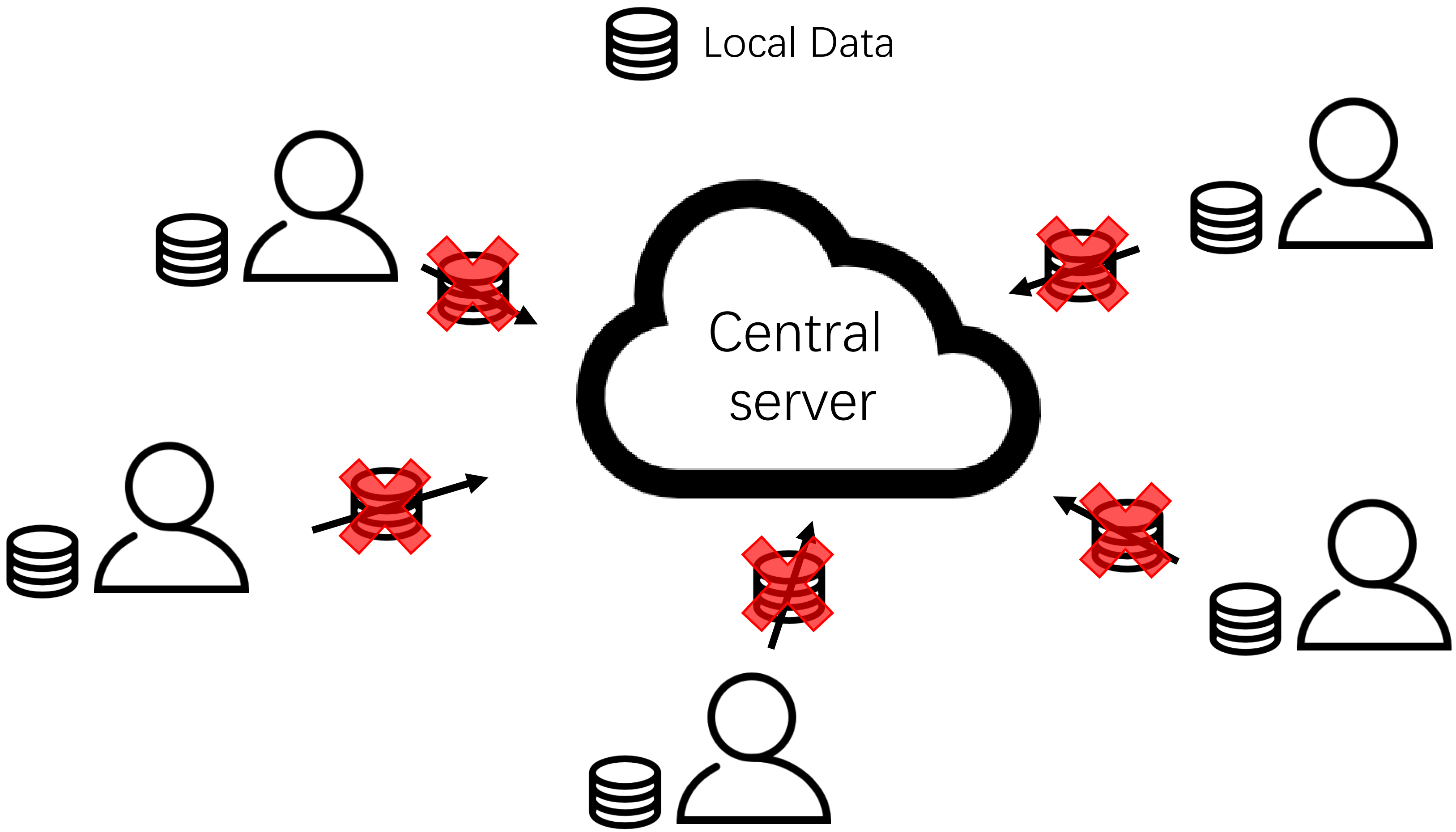
1 上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

2 QUEEN'S UNIVERSITY BELFAST

3 LSU

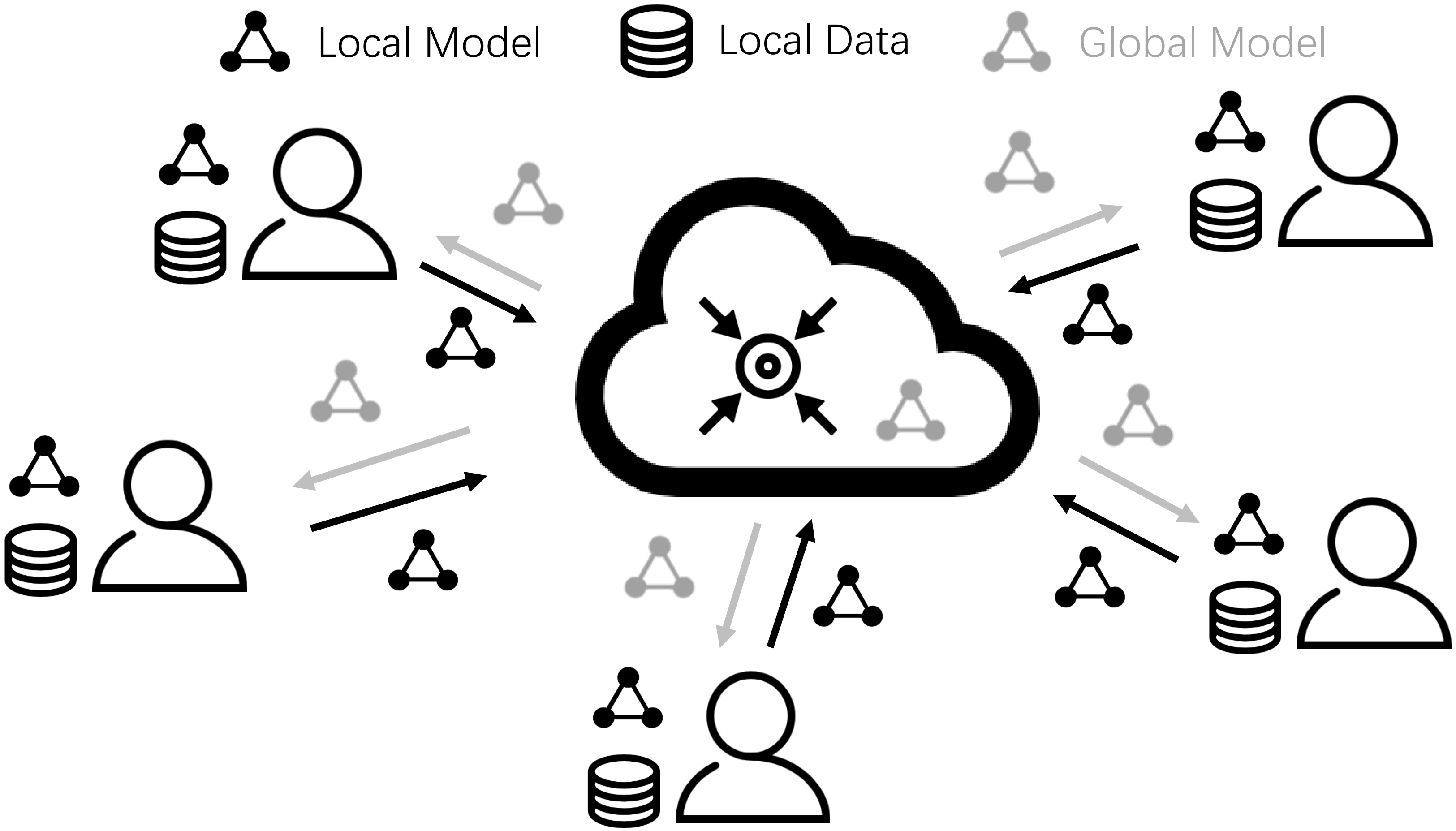
Federated Learning (FL)

- **Protect privacy** without uploading local data to the central server



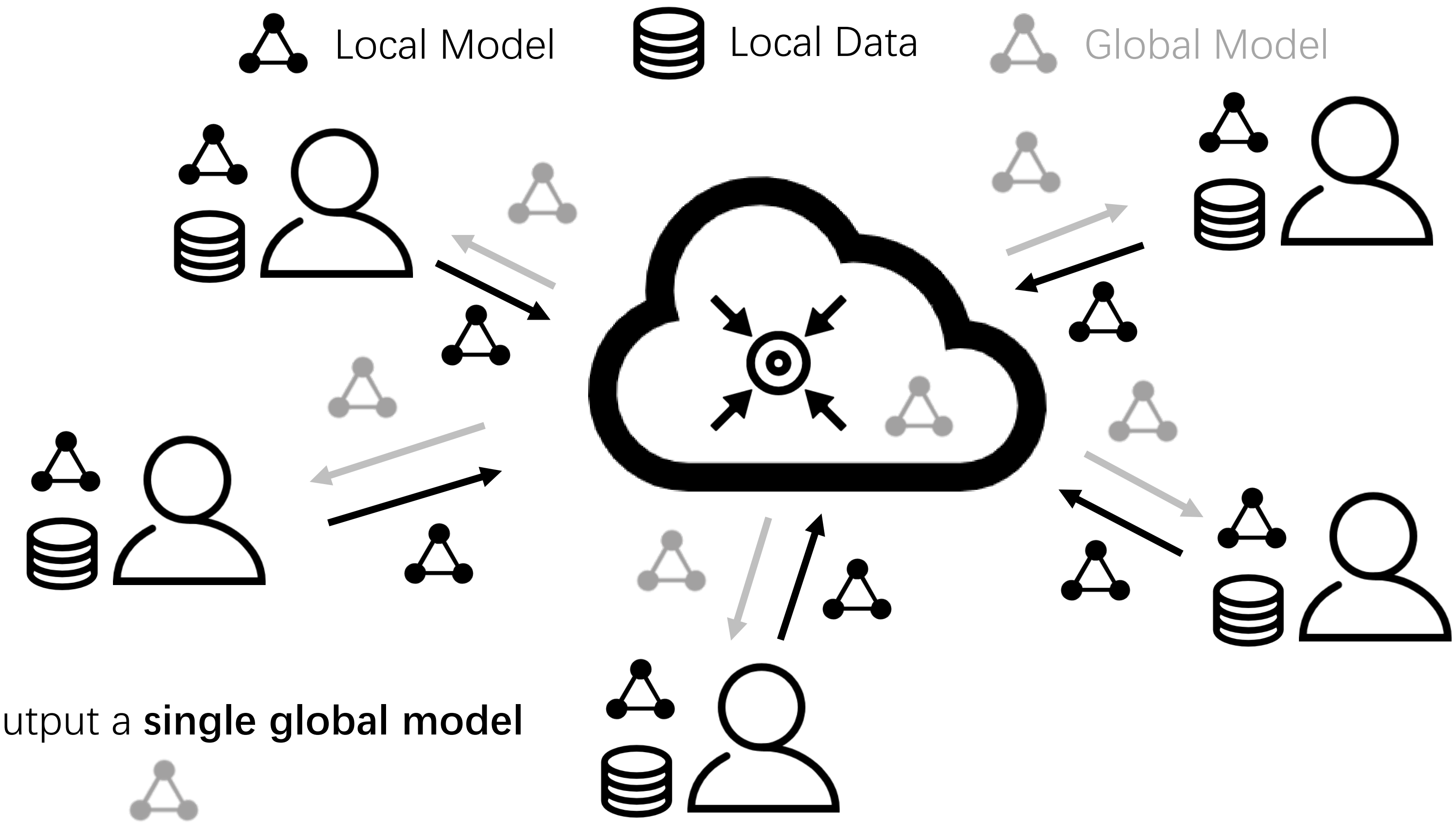
Federated Learning (FL)

- Learn an **AI model** among clients by sharing models with the server.



Federated Learning (FL)

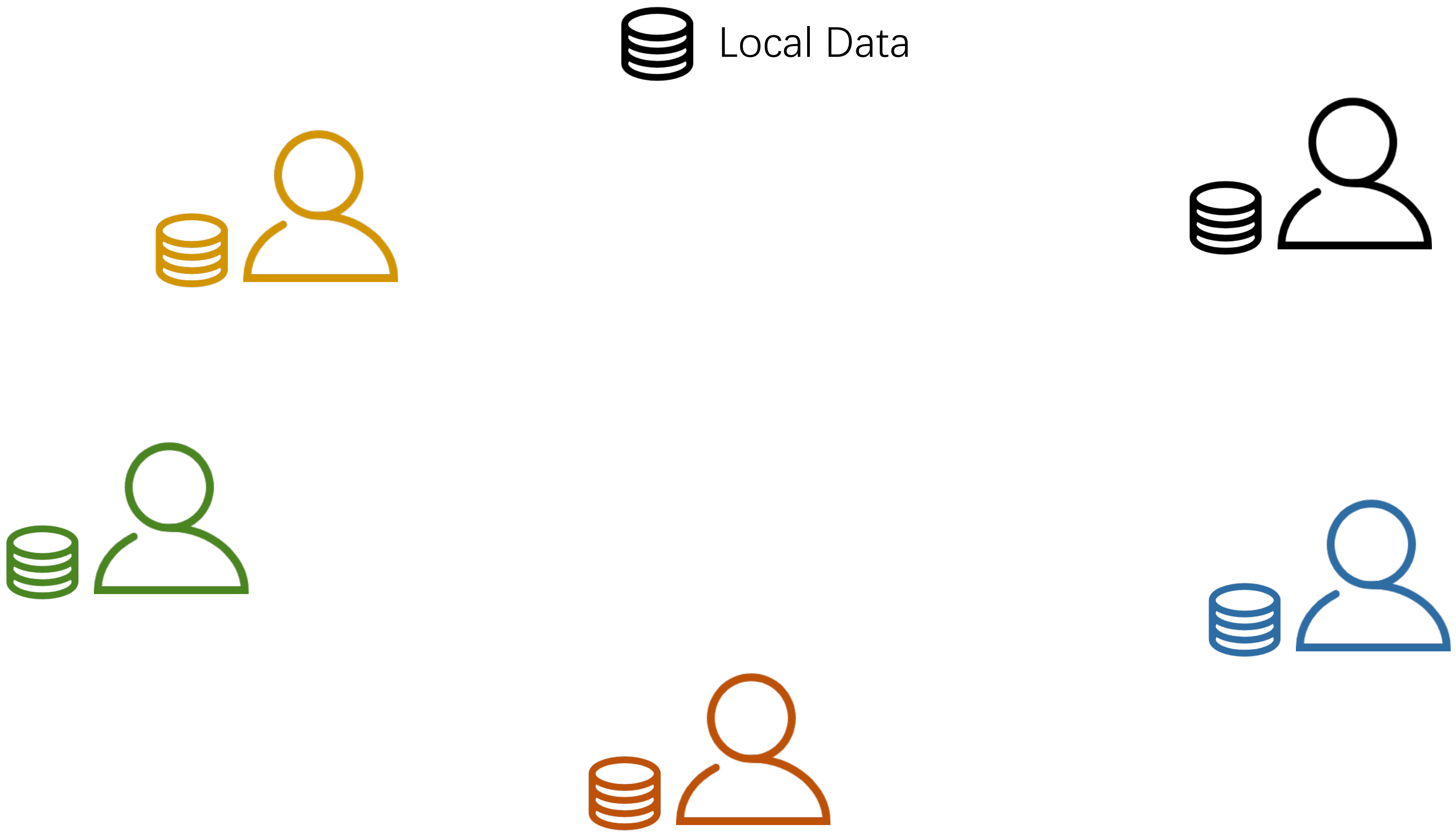
- Learn an **AI model** among clients by sharing models with the server.



Finally output a **single global model**

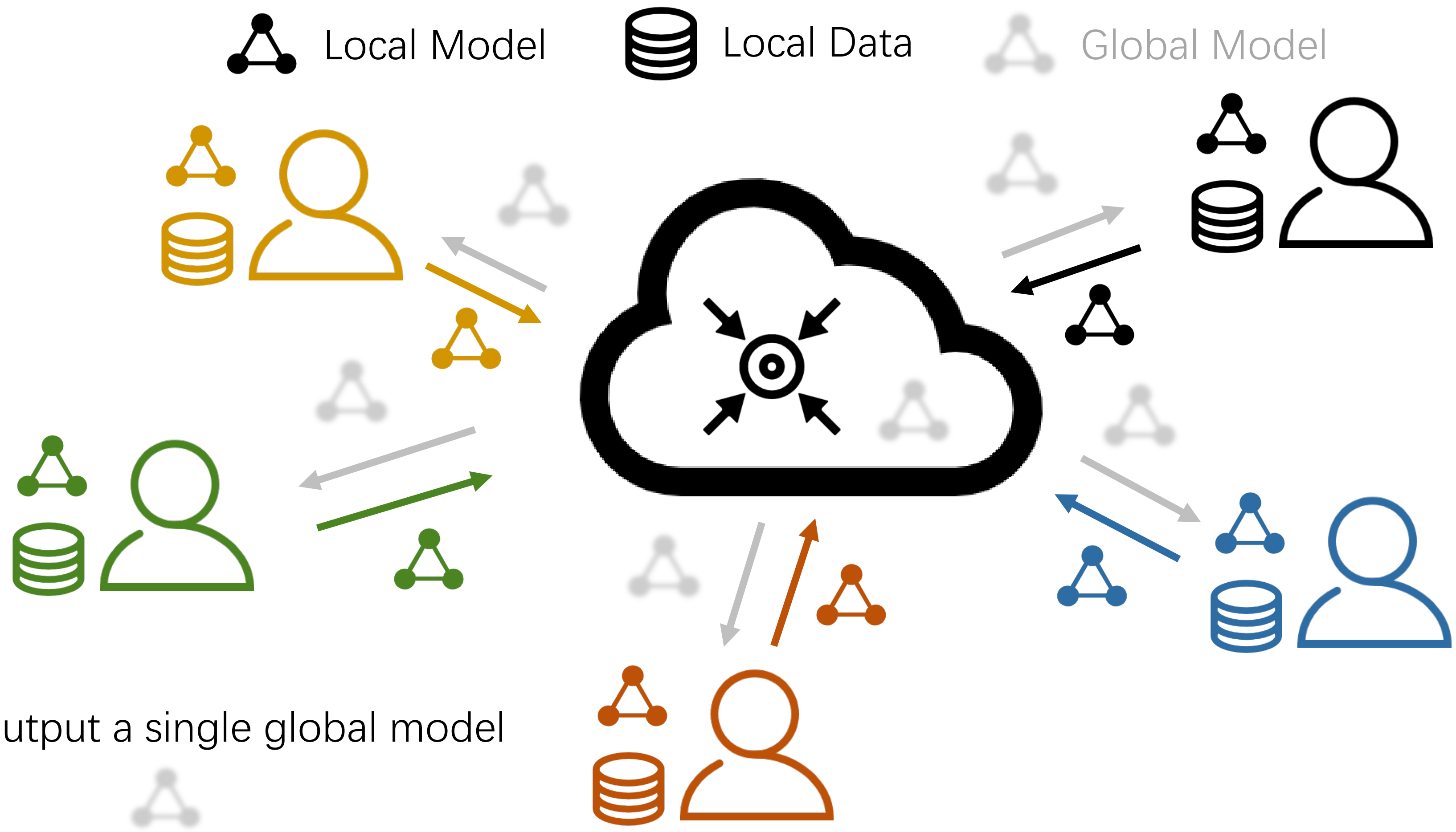
Issues in Federated Learning

- **Statistical heterogeneity**, such as non-IID and unbalanced data (colorful)



Issues in Federated Learning

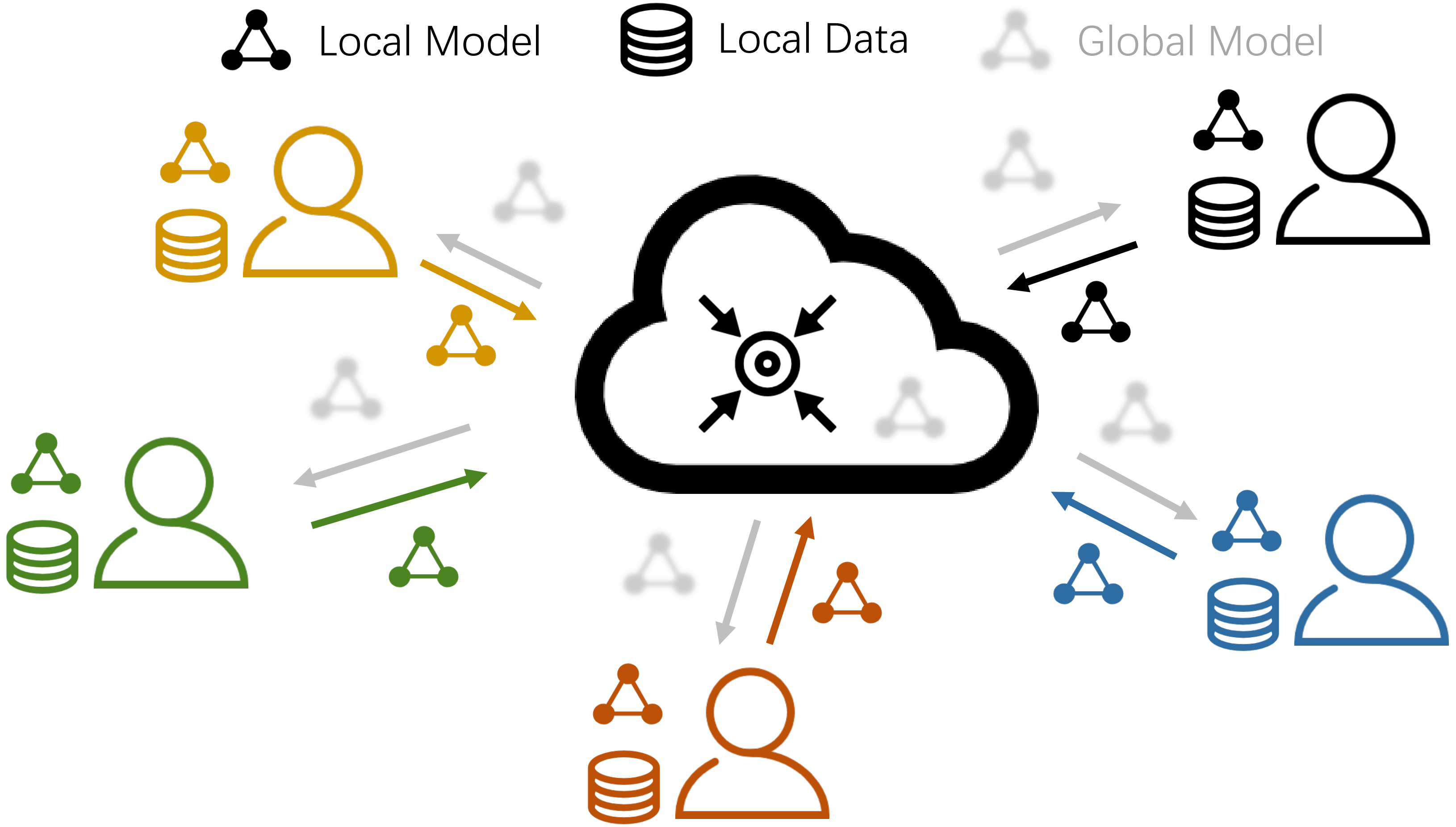
- **Poor generalization ability** (blurred) of the single global model on each client



Finally output a single global model

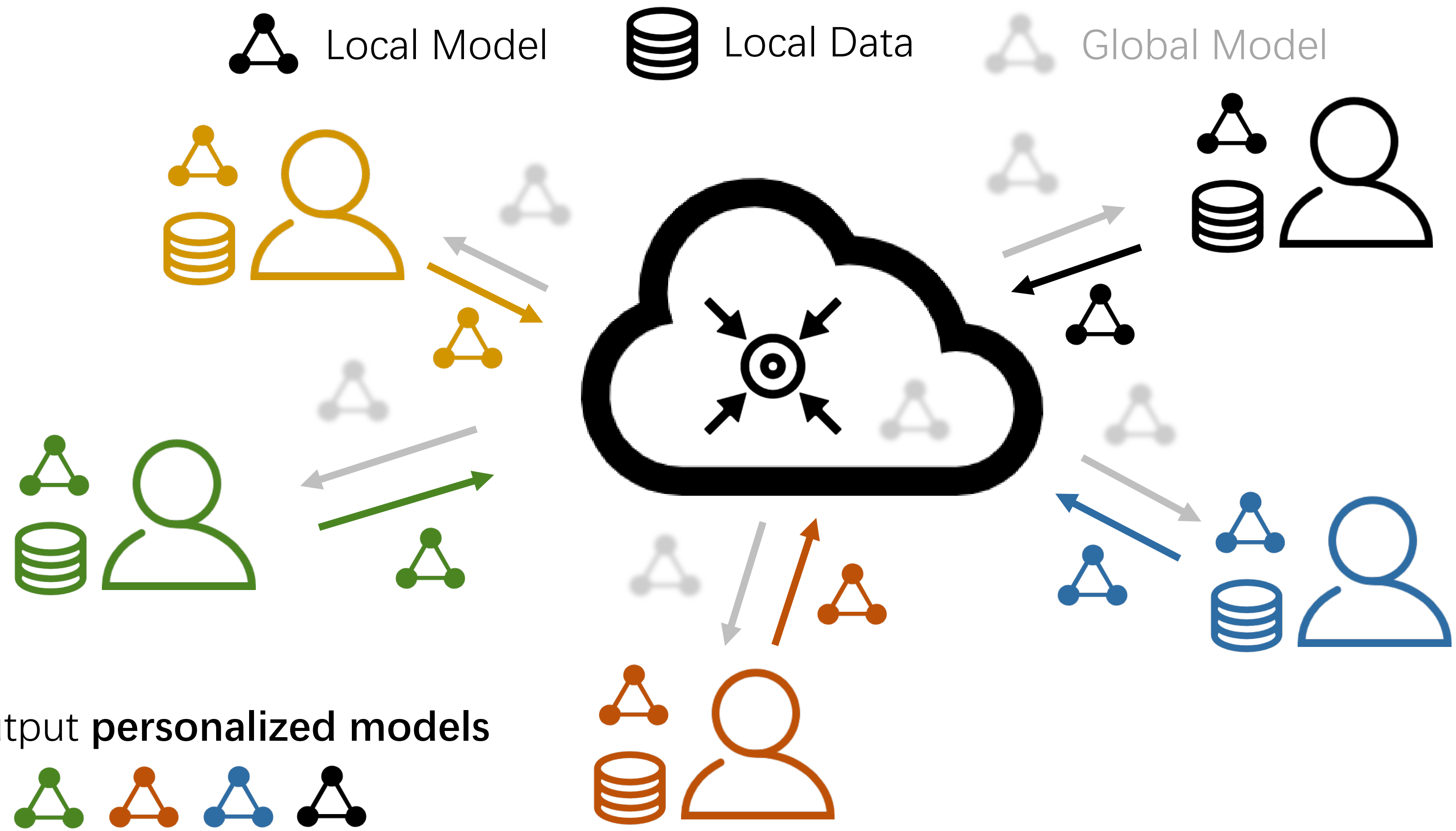
Personalized Federated Learning (pFL)

- Tackle the **statistical heterogeneity** issue
- Achieve **personalized requirements**



Personalized Federated Learning (pFL)

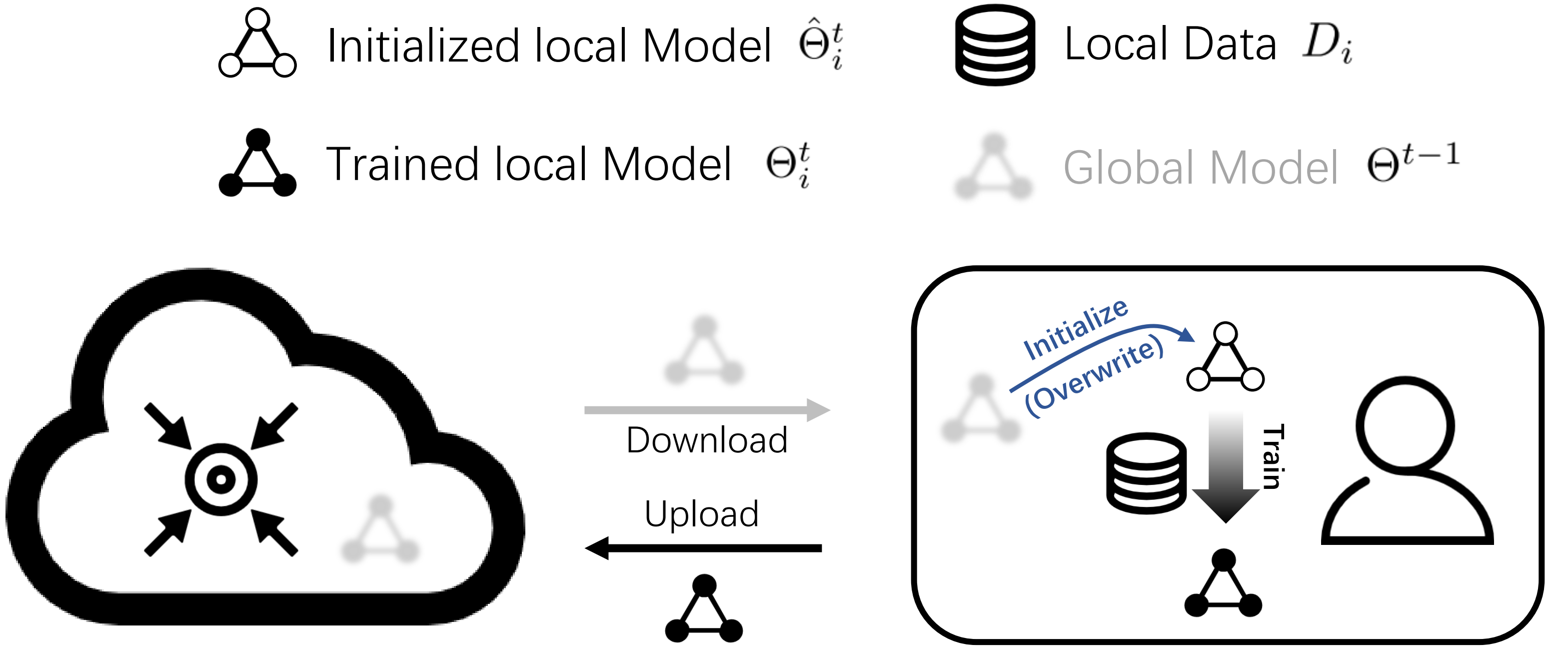
- Tackle the statistical heterogeneity issue
- Achieve personalized requirements



Finally output **personalized models**

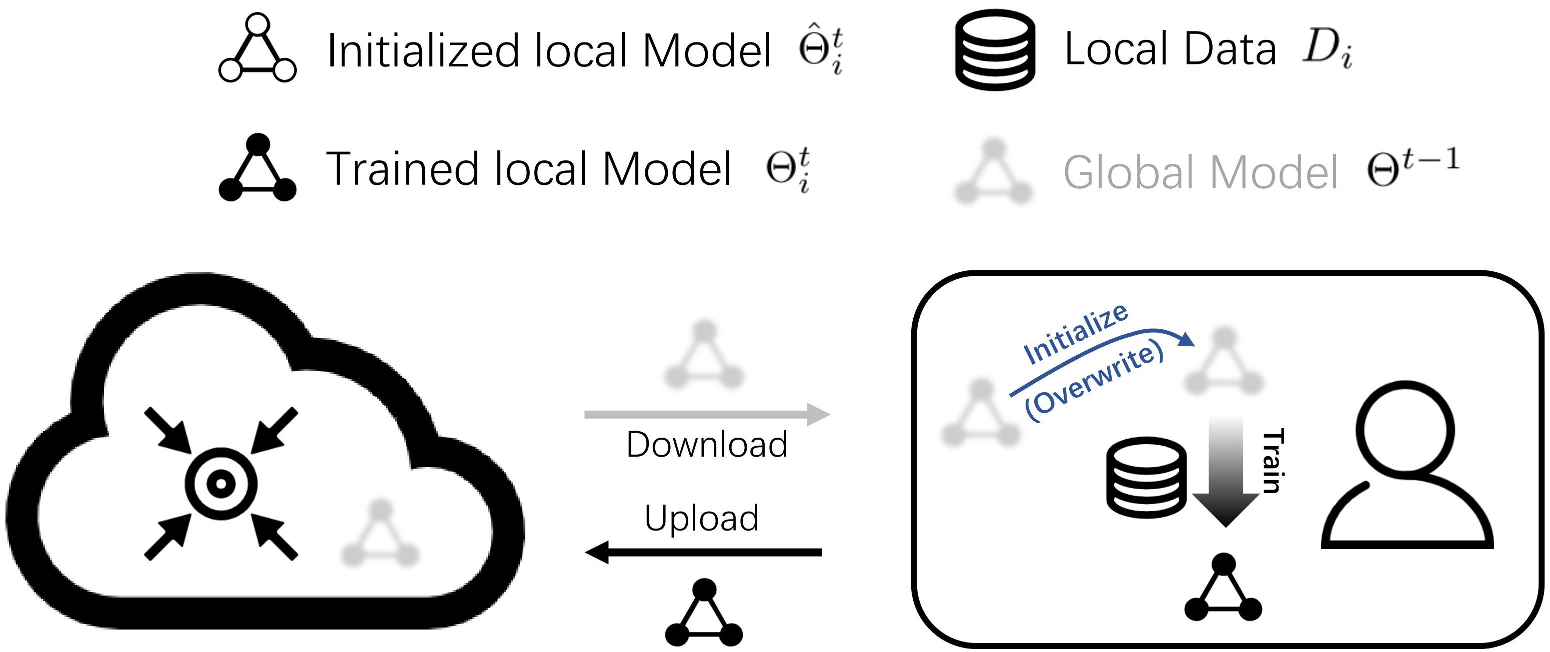
Motivation of FedALA

- Original workflow in FL
  - **Overwrite** local model with the **entire global model** for local initialization in each iteration



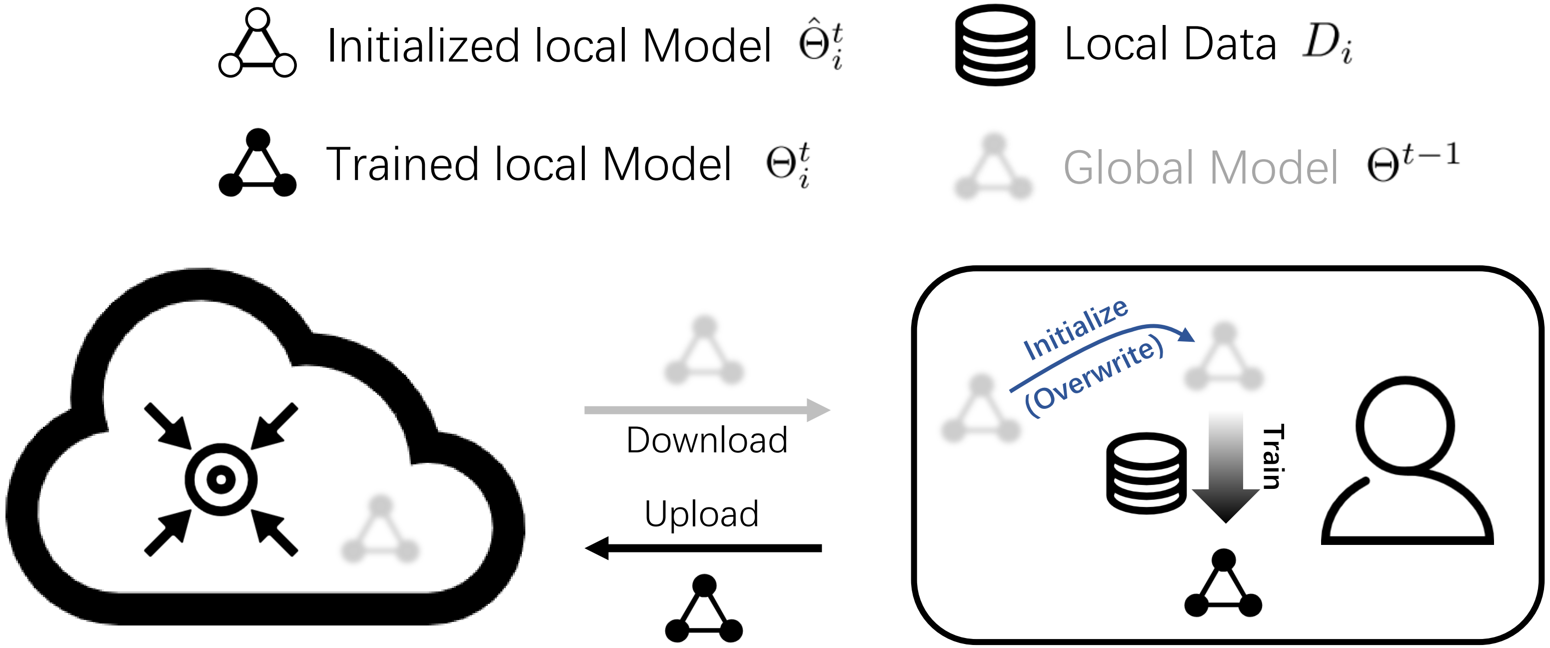
Motivation of FedALA

- Original workflow in FL
  - However, only the desired information that improves the quality of the local model is beneficial for the client



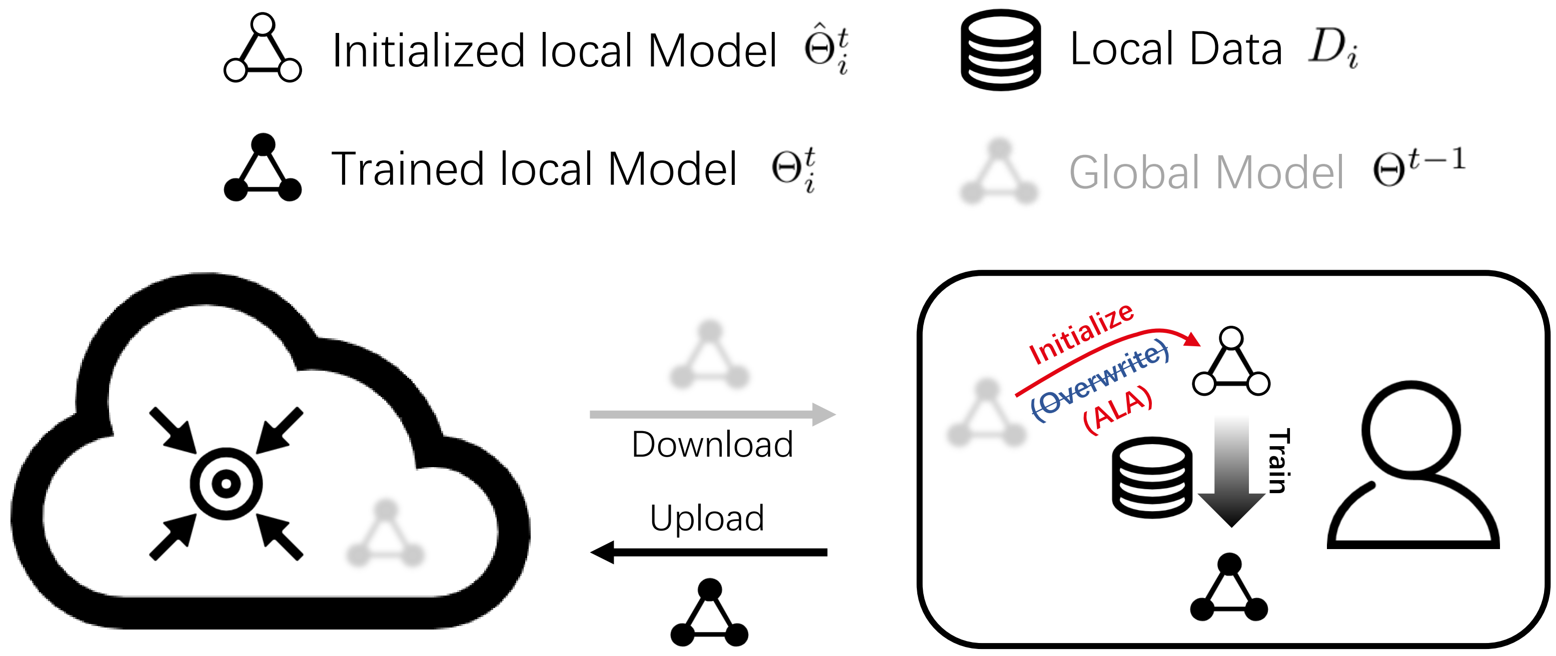
Motivation of FedALA

- Original workflow in FL
  - Both the desired and undesired information exist in the global model, resulting in poor generalization ability



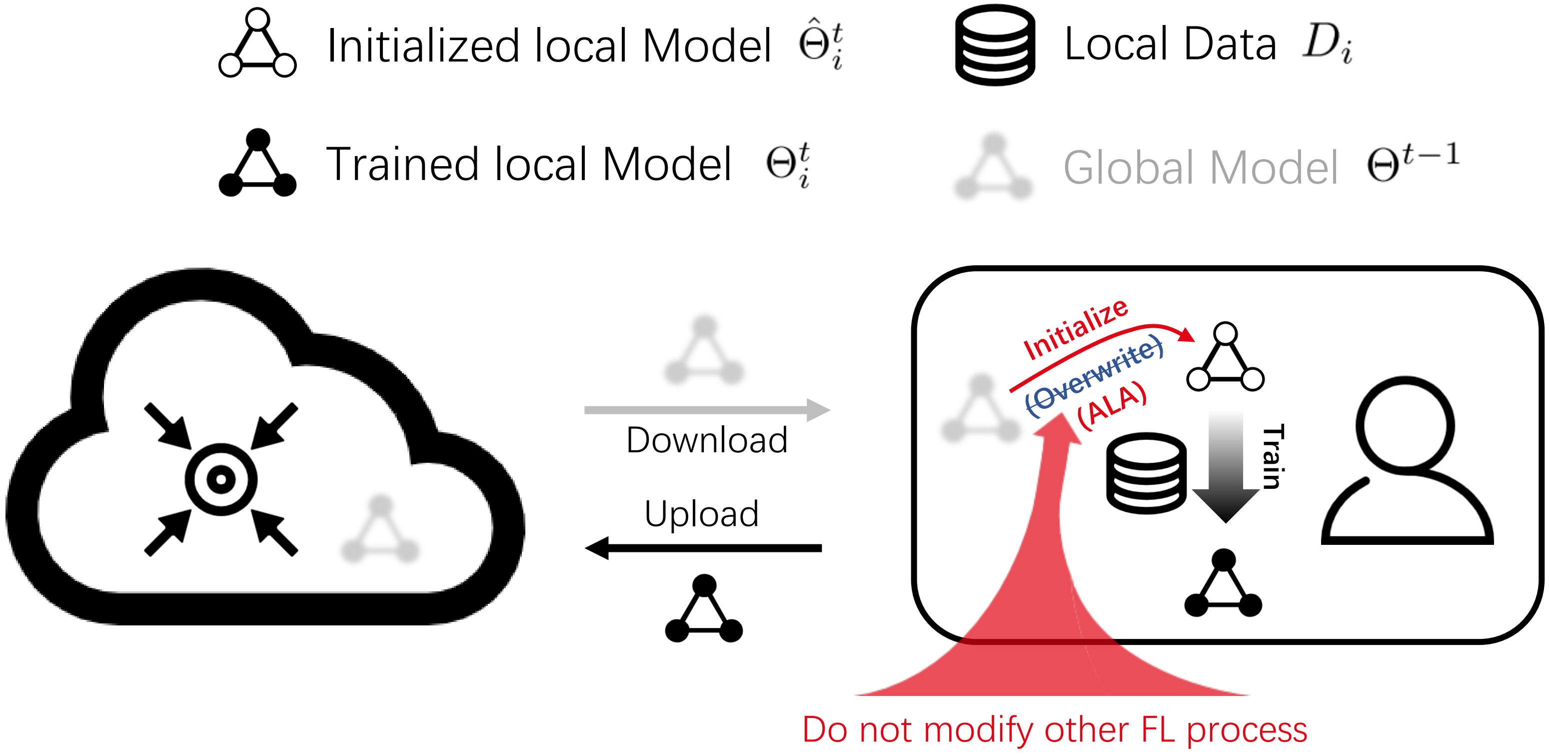
Motivation of FedALA

- Original workflow in FL
  - Both the desired and undesired information exist in the global model, resulting in poor generalization ability



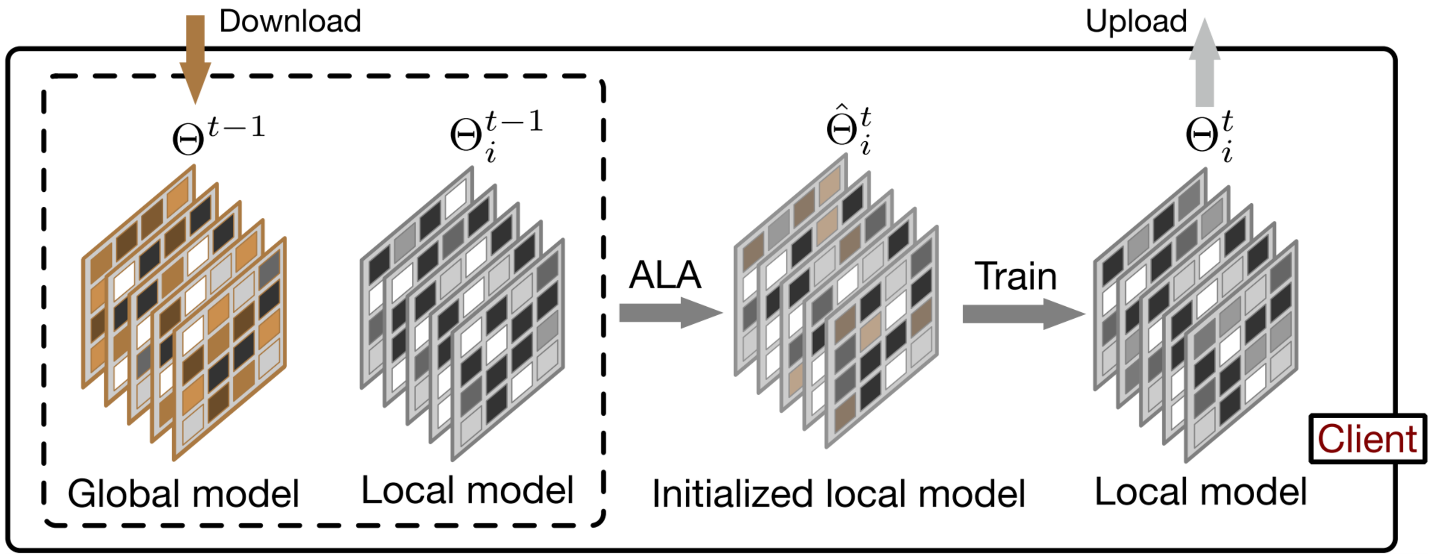
Motivation of FedALA

- Original workflow in FL
  - Both the desired and undesired information exist in the global model, resulting in poor generalization ability



FedALA: overview

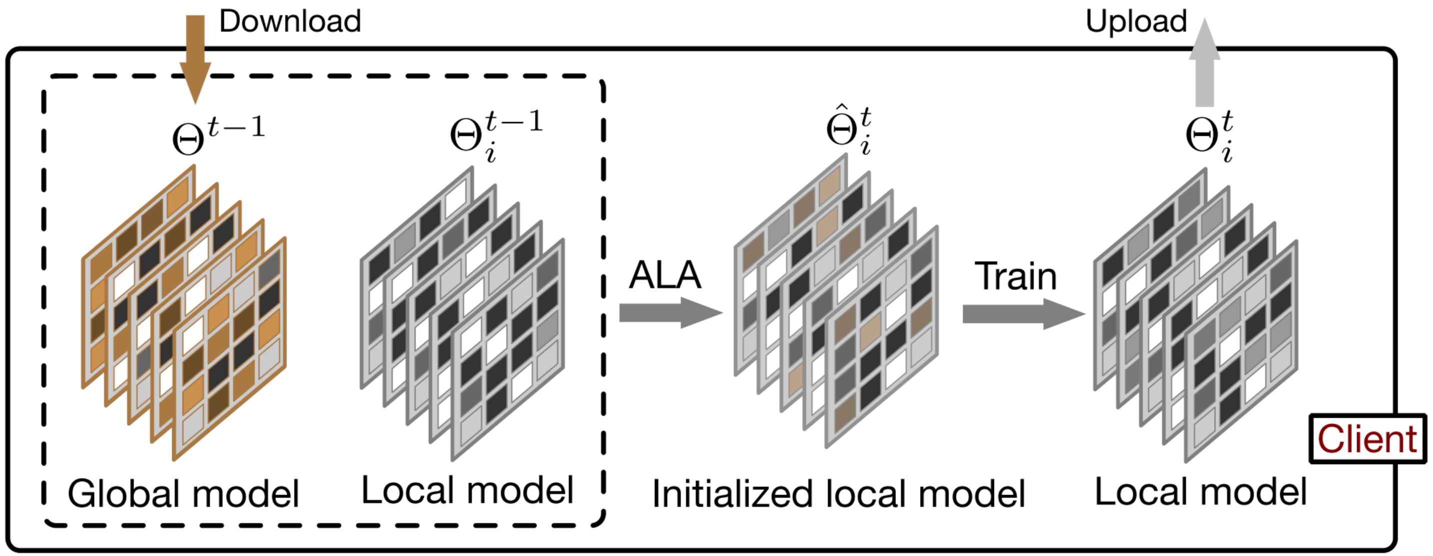
- **ALA**: adaptively aggregate the global model and local model for initialization



Workflow on the client in one iteration

FedALA: overview

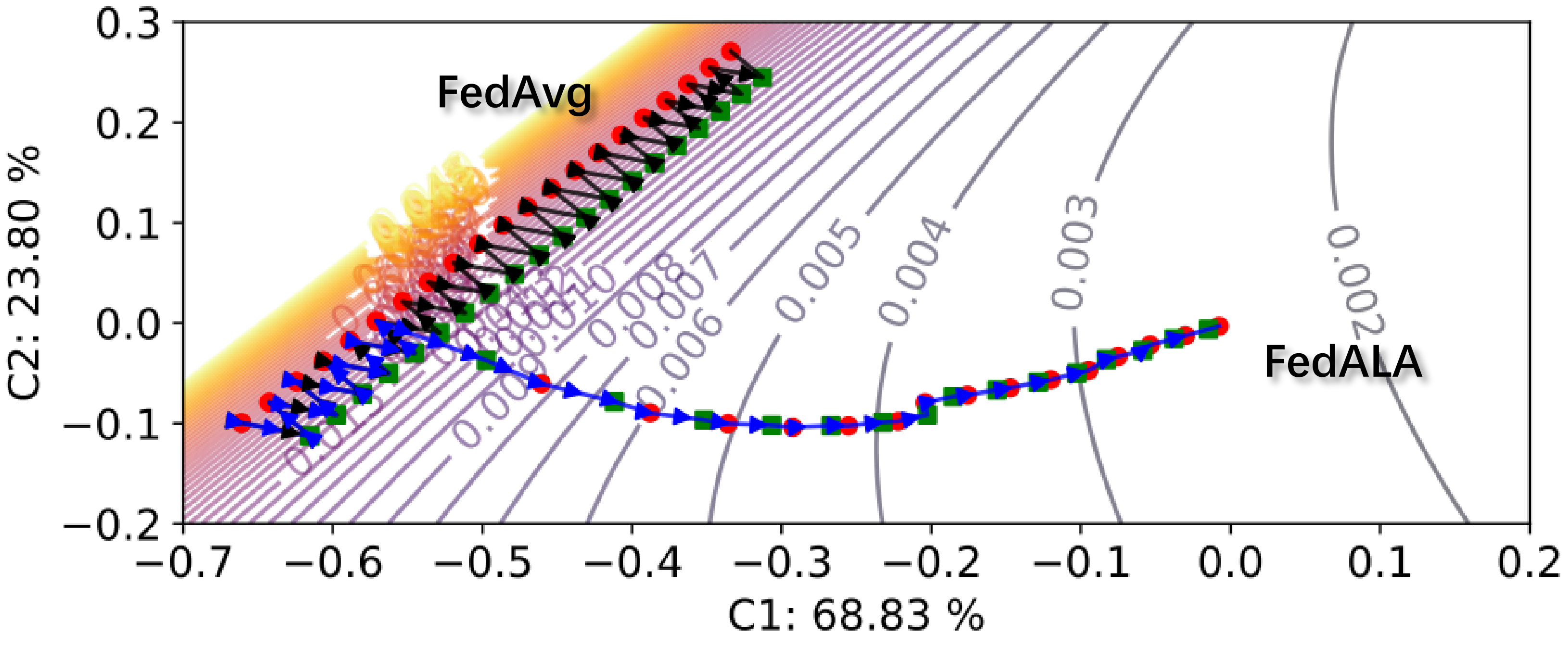
- **ALA**: adaptively aggregate the global model and local model for initialization
- **Train**: train the local model on the local data



Workflow on the client in one iteration

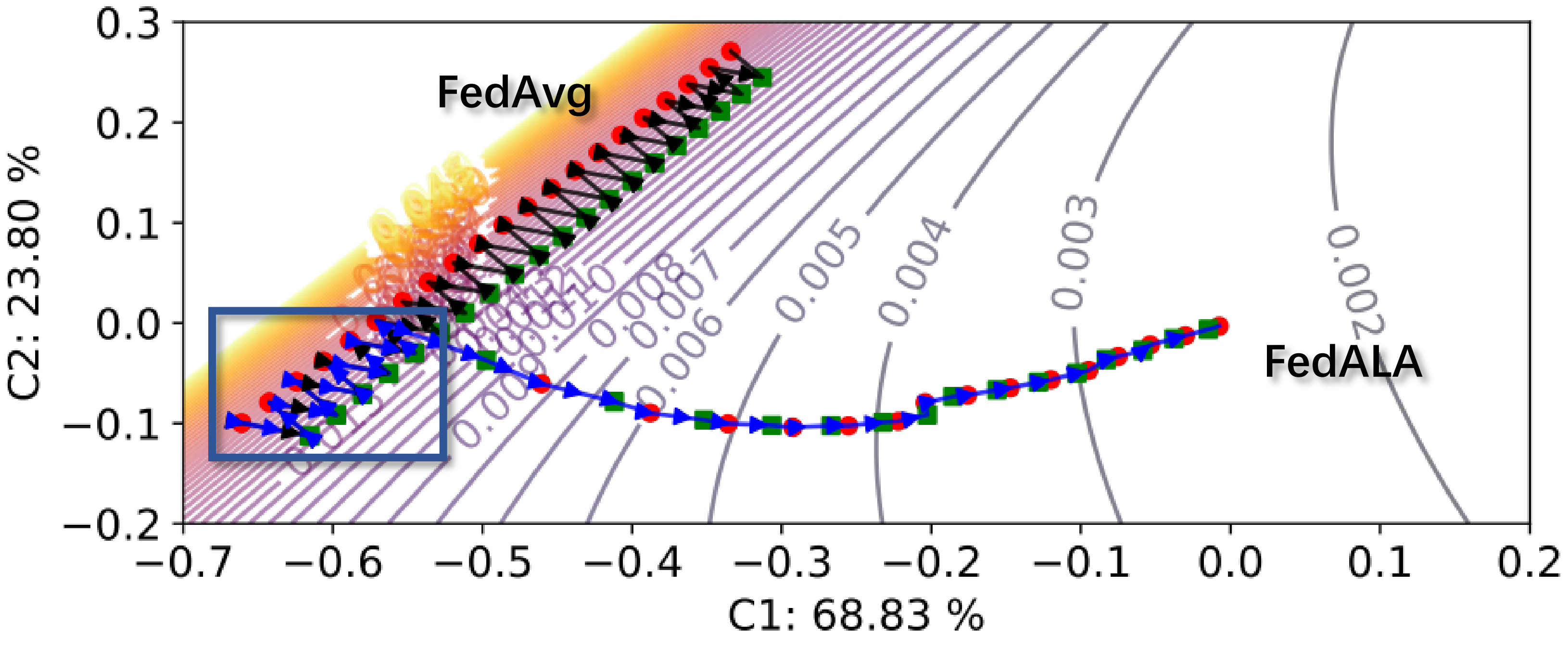
FedALA: overview

- Learning trajectory on one client: **FedAvg** vs. **FedALA**



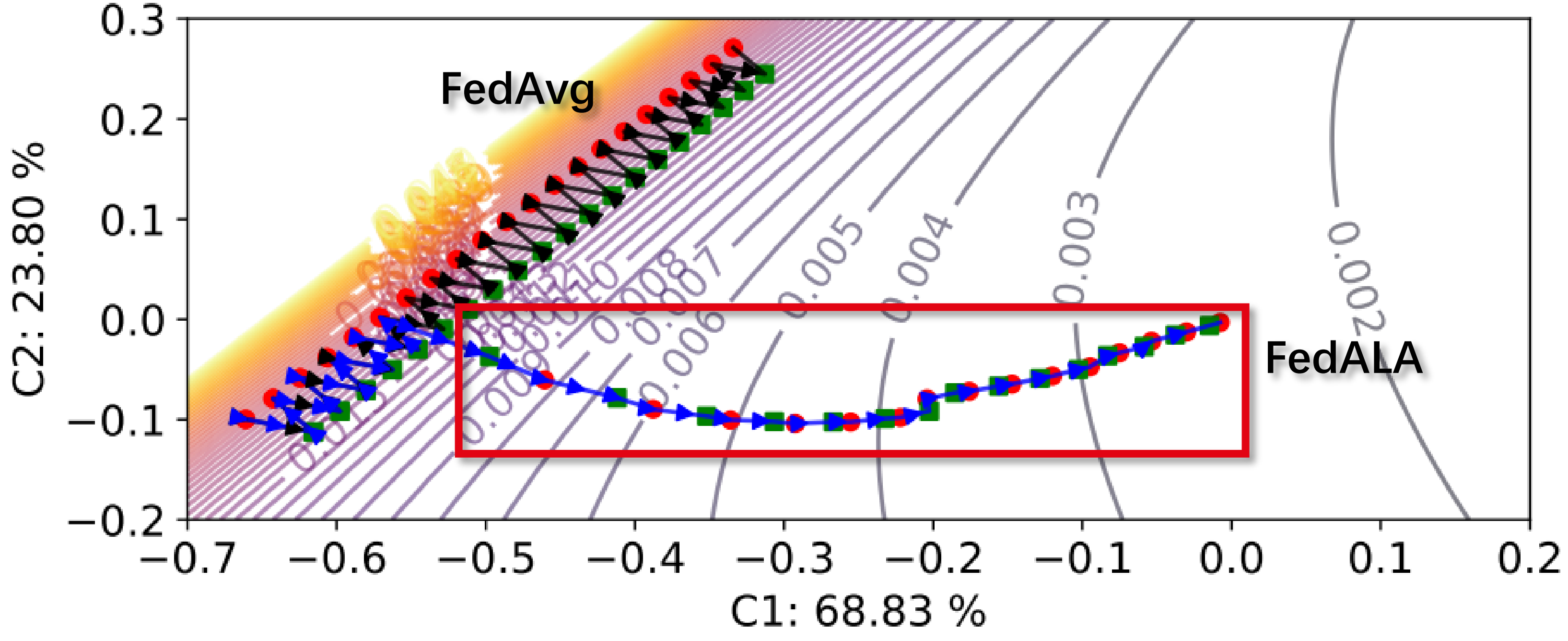
FedALA: overview

- Learning trajectory on one client: **FedAvg** vs. **FedALA**
- Deactivate **ALA** for **FedALA** in early iterations



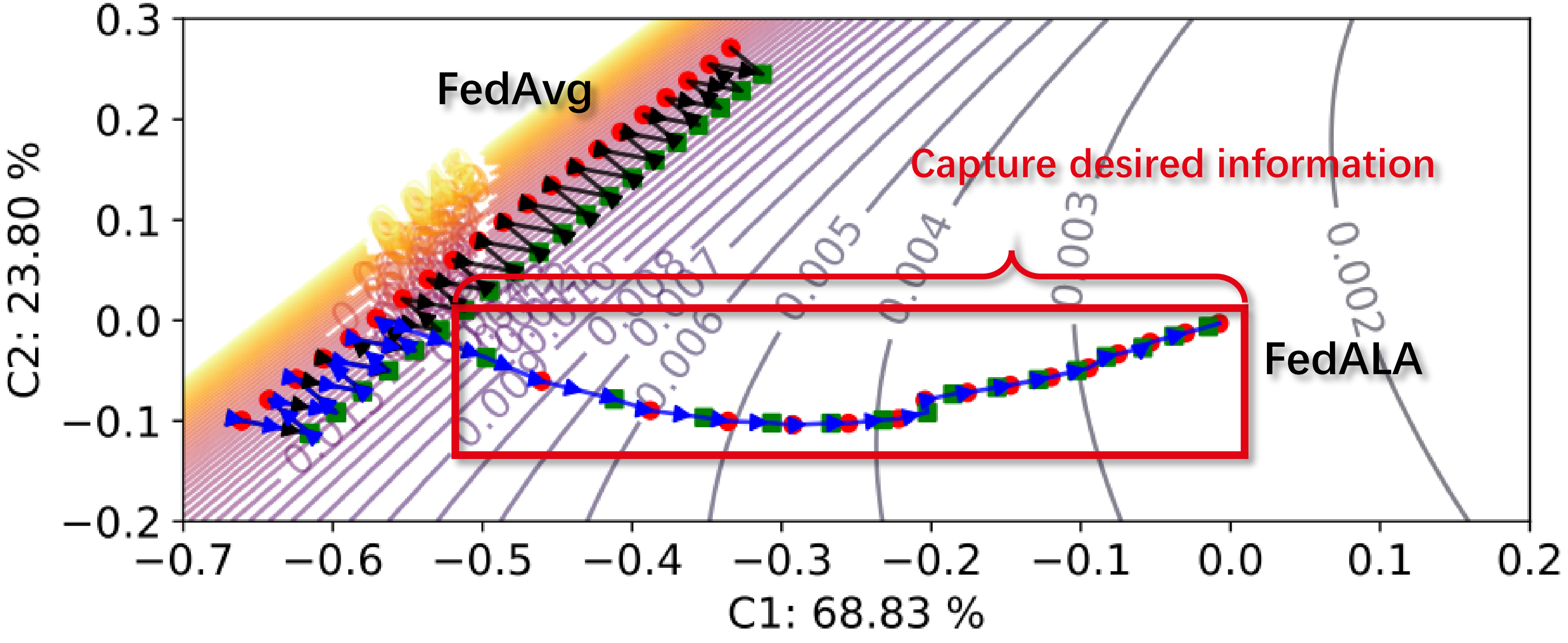
FedALA: overview

- Learning trajectory on one client: **FedAvg** vs. **FedALA**
- <u>Activate</u> **ALA** in the subsequent iterations



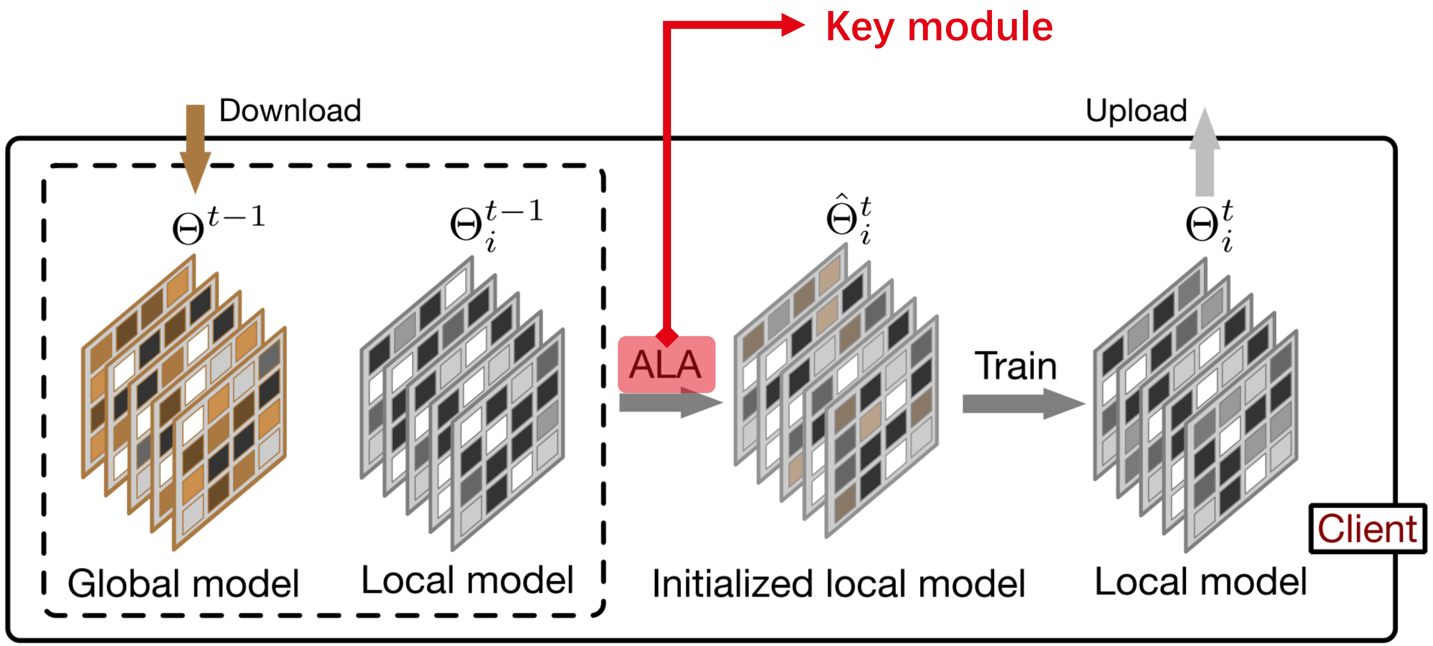
FedALA: overview

- Learning trajectory on one client: **FedAvg** vs. **FedALA**
- <u>Activate</u> **ALA** in the subsequent iterations



FedALA: overview

- **ALA**: adaptively aggregate the global model and local model for initialization
- **Train**: train the local model based on the initialized local model



Workflow on the client in one iteration

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$
\begin{aligned}
\hat{\Theta}_i^t &:= \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2}, \\
s.t. \quad & w_1^q + w_2^q = 1, \forall \text{ valid } q
\end{aligned}
$$

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2},$$
$$s.t. \quad w_1^q + w_2^q = 1, \forall \text{ valid } q$$

Trainable weights

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot \boxed{W_{i,1}} + \Theta^{t-1} \odot \boxed{W_{i,2}},$$
$$s.t. \quad \boxed{w_1^q + w_2^q = 1, \forall \text{ valid } q}$$

Trainable weights

Hard to learn weights with constraints

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot \boxed{W_{i,1}} + \Theta^{t-1} \odot \boxed{W_{i,2}},$$
$$s.t. \quad \boxed{w_1^q + w_2^q = 1, \forall \text{ valid } q}$$

Trainable weights

Hard to learn weights with constraints

- Combine $W_{i,1}$ and $W_{i,2}$

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot \boxed{W_i}$$

Trainable weights

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2},$$
$$s.t. \quad w_1^q + w_2^q = 1, \forall \text{ valid } q$$

Trainable weights

Hard to learn weights with constraints

- Combine $W_{i,1}$ and $W_{i,2}$

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

Trainable weights

Called "update"

View the local aggregation as an **update process** for old local model

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2},$$
$$s.t. \quad w_1^q + w_2^q = 1, \forall \text{ valid } q$$

Trainable weights

Hard to learn weights with constraints

- Combine $W_{i,1}$ and $W_{i,2}$

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

Trainable weights

Called "update"

  - remove constraints

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2},$$
$$s.t. \quad w_1^q + w_2^q = 1, \forall \text{ valid } q$$

Trainable weights

Hard to learn weights with constraints

- Combine $W_{i,1}$ and $W_{i,2}$

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

Trainable weights

Called "update"

  - remove constraints
  - with weight clipping[1]

$$\begin{aligned} \sigma(w) &= \max(0, \min(1, w)) \\ & w \in [0,1], \forall w \in W_i \end{aligned}$$

[1] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. ICLR, 2017.

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2},$$
$$s.t. \quad w_1^q + w_2^q = 1, \forall \text{ valid } q$$

Trainable weights

Hard to learn weights with constraints

- Combine $W_{i,1}$ and $W_{i,2}$

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

Trainable weights

Called "update"

ALA covers the entire model

  - remove constraints
  - with weight clipping[1]

$$\sigma(w) = \max(0, \min(1, w))$$
$$w \in [0,1], \forall w \in W_i$$

[1] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. ICLR, 2017.

FedALA: ALA module

- Element-wisely aggregate the global model and local model in an adaptive way

$$\hat{\Theta}_i^t := \Theta_i^{t-1} \odot W_{i,1} + \Theta^{t-1} \odot W_{i,2},$$
$$s.t. \quad w_1^q + w_2^q = 1, \forall \text{ valid } q$$

Trainable weights

Hard to learn weights with constraints

- Combine $W_{i,1}$ and $W_{i,2}$

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

Trainable weights

Called "update"

ALA covers the entire model

How to reduce computation overhead?

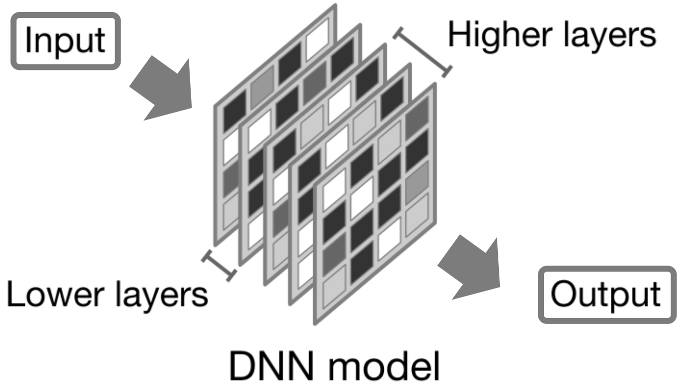
  - remove constraints
  - with weight clipping[1]

$$\boxed{\begin{aligned} \sigma(w) &= \max(0, \min(1, w)) \\ & w \in [0, 1], \forall w \in W_i \end{aligned}}$$

[1] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. ICLR, 2017.

FedALA: ALA module

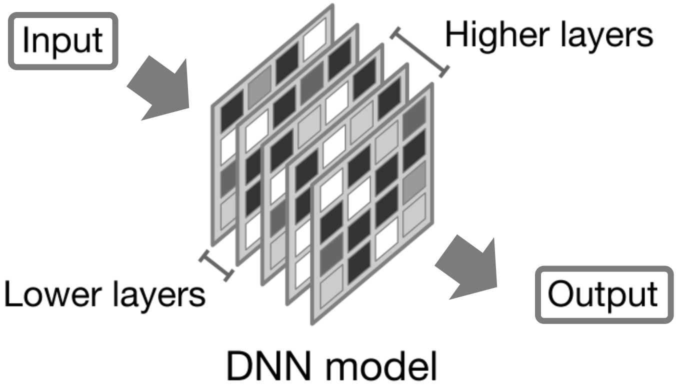
- The lower layers in the DNN learn more general information than the higher layers[2]



[2] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? NeurIPS, 2014.

FedALA: ALA module

- The lower layers in the DNN learn more general information than the higher layers[2]



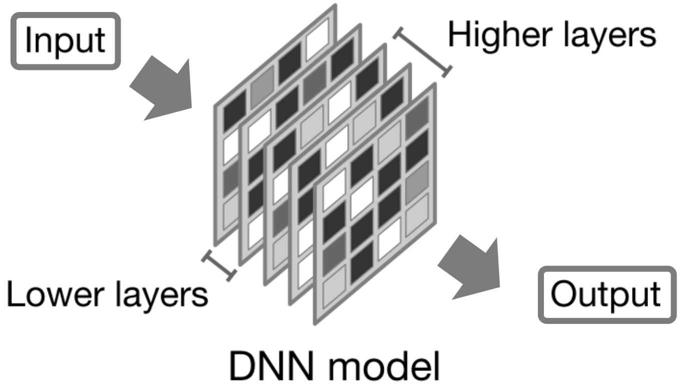
- **Only apply ALA on p higher layers**

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p]$$

[2] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? NeurIPS, 2014.

FedALA: ALA module

- The lower layers in the DNN learn more general information than the higher layers[2]



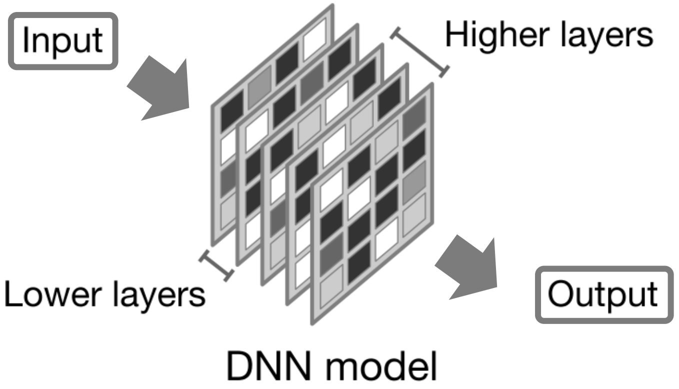
- Only apply ALA on p higher layers
- Still overwrite the lower layers with global parameters

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p]$$

[2] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? NeurIPS, 2014.

FedALA: ALA module

- The lower layers in the DNN learn more general information than the higher layers[2]



- Only apply ALA on p higher layers
- Still overwrite the lower layers with global parameters

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p]$$

Fewer weights to train in ALA

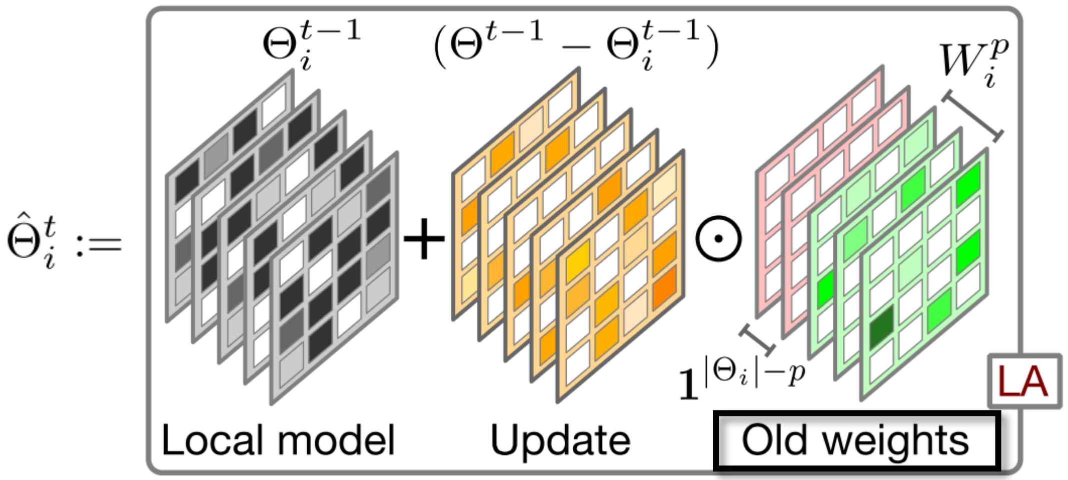
Less computation overhead

[2] Yosinski J, Clune J, Bengio Y, et al. How transferable are features in deep neural networks? NeurIPS, 2014.

FedALA: ALA module

- Only apply ALA on p higher layers
- Still overwrite the lower layers with global parameters

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p]$$

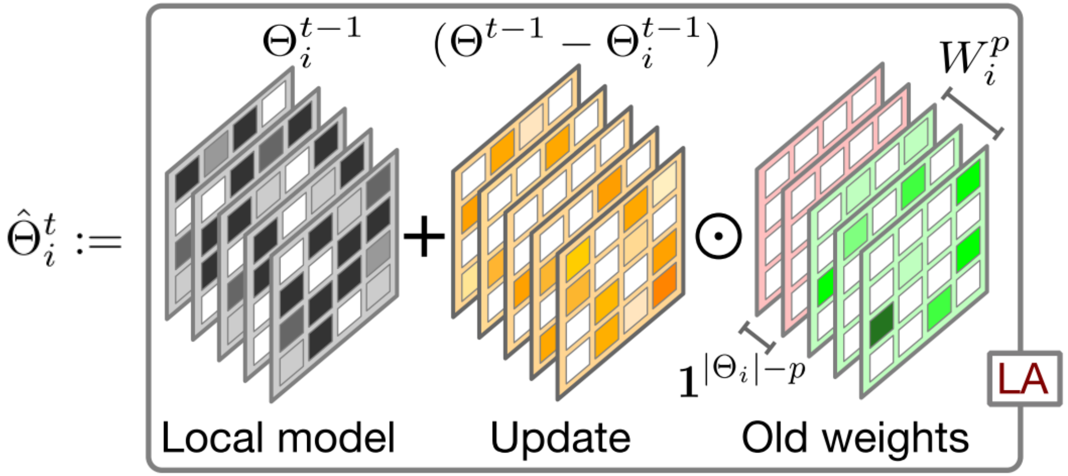


Local aggregation (LA)

FedALA: ALA module

- Only apply ALA on p higher layers
- Still overwrite the lower layers with global parameters

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p]$$



How to train weights?

Local aggregation (LA)

FedALA: ALA module

- Train weights to **reduce local loss** $\mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$ to find **client desired information**

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^t; \Theta^{t-1})$$

FedALA: ALA module

- Train weights to reduce local loss $\mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$ to find client desired information

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^t; \Theta^{t-1})$$

How to further reduce computation overhead?

FedALA: ALA module

- Train weights to reduce local loss $\mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$ to find client desired information

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^t; \Theta^{t-1})$$

How to further reduce computation overhead?

- **Randomly** sample ***s***% data from **local dataset** D_i^t to form a **sub-dataset** $D_i^{s,t}$

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

FedALA: ALA module

- Train weights to reduce local loss $\mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$ to find client desired information

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^t; \Theta^{t-1})$$

How to further reduce computation overhead?

- **Randomly** sample $s\%$ data from local dataset D_i^t to form a **sub-dataset** $D_i^{s,t}$

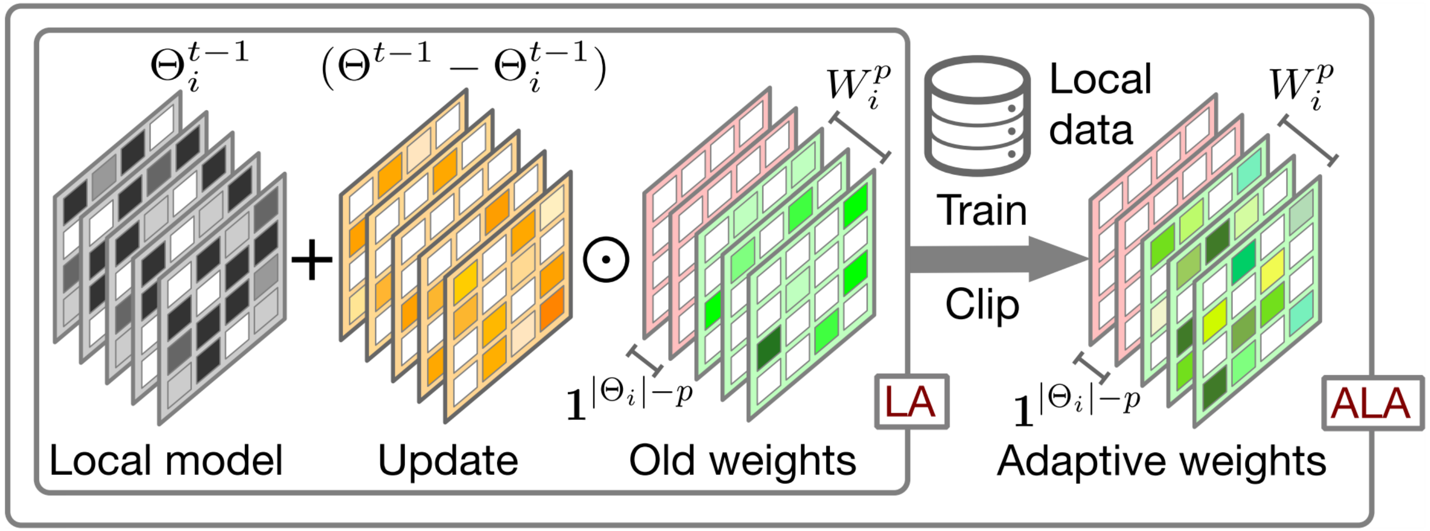
$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, \boxed{D_i^{s,t}}; \Theta^{t-1})$$

Covers all the data when t accumulates from 1 to T

FedALA: ALA module

- Randomly sample $s\%$ data from local dataset D_i^t to form a sub-dataset $D_i^{s,t}$

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

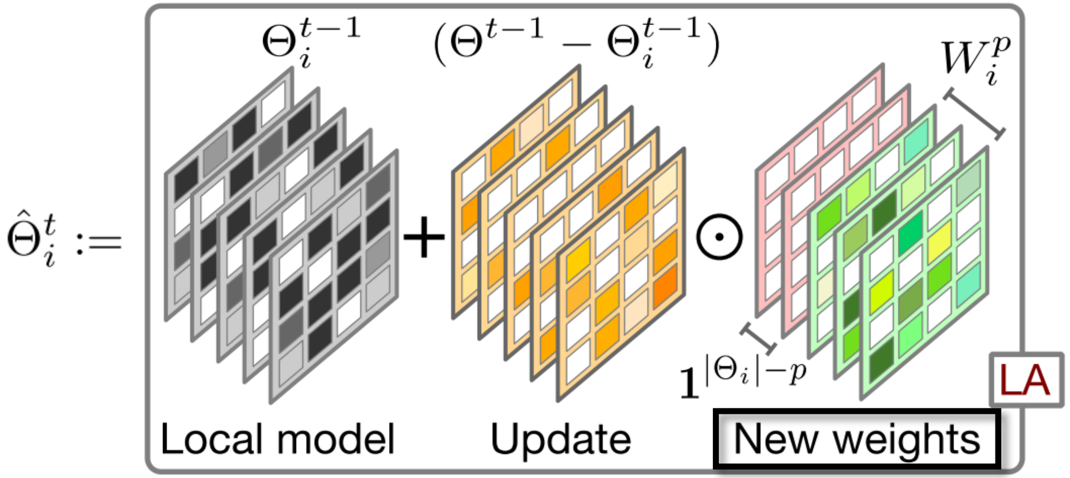


Adaptive local aggregation (ALA)

FedALA: ALA module

- Finally, obtain the initialized local model **with new weights**

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot [\mathbf{1}^{|\Theta_i|-p}; W_i^p]$$

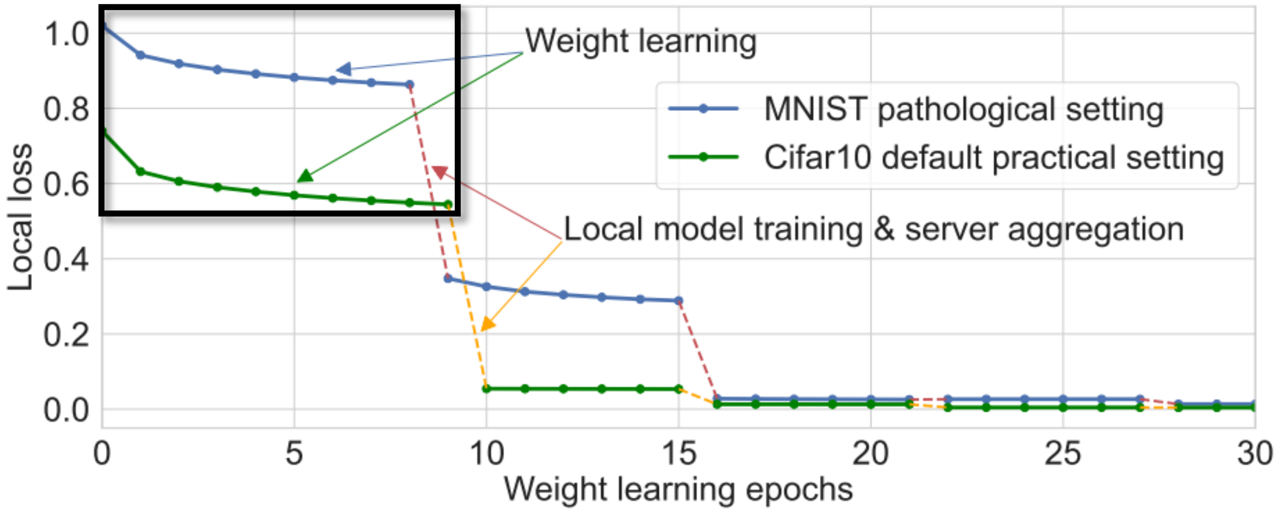


Local aggregation (LA)

FedALA: observations

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

- Once we train the weights to converge **in the first time**,

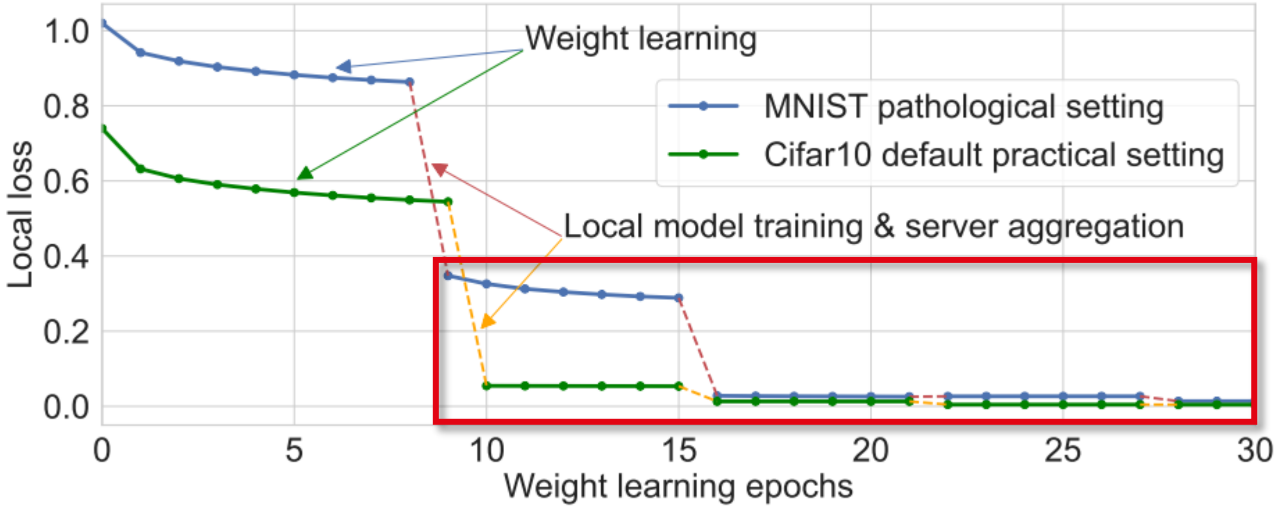


The local loss on client #8 regarding weight learning epochs in ALA on MNIST and Cifar10.

FedALA: observations

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

- Once we train the weights to converge in the first time,
  the weights hardly change in the subsequent iterations

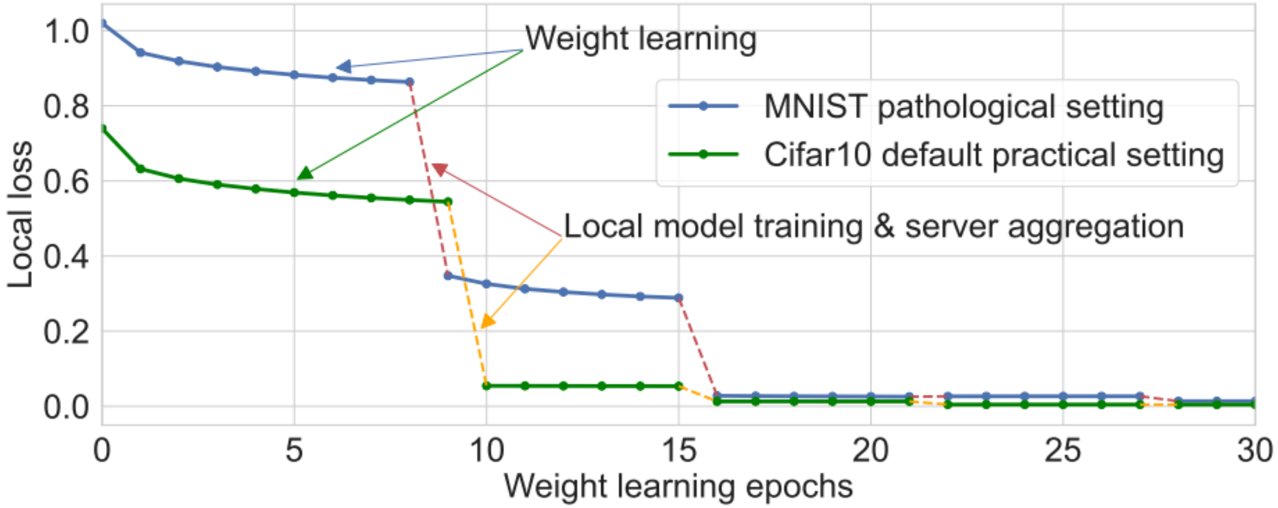


The local loss on client #8 regarding weight learning epochs in ALA on MNIST and Cifar10.

FedALA: observations

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

- The weights can be **reused** or just require few steps of fine-tuning for adaptation



The local loss on client #8 regarding weight learning epochs in ALA on MNIST and Cifar10.

FedALA: overall algorithm

Algorithm 1: FedALA

Input: N clients, ρ: client joining ratio, $\mathcal{L}$: loss function, Θ^0: initial global model, α: local learning rate, η: the learning rate in ALA, $s\%$: the percent of local data in ALA, p: the range of ALA, $\sigma(\cdot)$: clip function.

Output: Reasonable local models $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

Server sends Θ^0 to all clients to initialize local models.
Clients initialize $W_i^p, \forall i \in [N]$ to ones.
for iteration $t = 1, \ldots, T$ **do**
    Server samples a subset $\mathcal{I}^t$ of clients according to ρ.
    Server sends Θ^{t-1} to $|\mathcal{I}^t|$ clients.
    for Client $i \in \mathcal{I}^t$ in parallel **do**
        Client i samples $s\%$ of local data. ▷ **ALA**
        if $t = 2$ **then** ▷ Start stage
            while W_i^p does not converge **do**
                Client i trains W_i^p by Equation (5).
                Client i clips W_i^p using $\sigma(\cdot)$.
        else if $t > 2$ **then**
            Client i trains W_i^p by Equation (5).
            Client i clips W_i^p using $\sigma(\cdot)$.
        Client i obtains $\hat{\Theta}_i^t$ by Equation (4).
        Client i obtains Θ_i^t by ▷ **Local model training**
            $\Theta_i^t \leftarrow \hat{\Theta}_i^t - \alpha \nabla_{\hat{\Theta}_i} \mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$.
        Client i sends Θ_i^t to the server. ▷ **Uploading**
    Server obtains Θ^t by $\Theta^t \leftarrow \sum_{i \in \mathcal{I}^t} \frac{k_i}{\sum_{j \in \mathcal{I}^t} k_j} \Theta_i^t$.
return $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

Only train weights to converge in the start stage

FedALA: overall algorithm

Algorithm 1: FedALA

Input: N clients, ρ: client joining ratio, $\mathcal{L}$: loss function, Θ^0: initial global model, α: local learning rate, η: the learning rate in ALA, $s\%$: the percent of local data in ALA, p: the range of ALA, $\sigma(\cdot)$: clip function.

Output: Reasonable local models $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

Server sends Θ^0 to all clients to initialize local models.
Clients initialize $W_i^p, \forall i \in [N]$ to ones.
for iteration $t = 1, \ldots, T$ **do**
  Server samples a subset $\mathcal{I}^t$ of clients according to ρ.
  Server sends Θ^{t-1} to $|\mathcal{I}^t|$ clients.
  for Client $i \in \mathcal{I}^t$ in parallel **do**
    Client i samples $s\%$ of local data. ▷ **ALA**
    if $t = 2$ **then** ▷ Start stage
      while W_i^p does not converge **do**
        Client i trains W_i^p by Equation (5).
        Client i clips W_i^p using $\sigma(\cdot)$.
    else if $t > 2$ **then**
      Client i trains W_i^p by Equation (5).
      Client i clips W_i^p using $\sigma(\cdot)$.
    Client i obtains $\hat{\Theta}_i^t$ by Equation (4).
    Client i obtains Θ_i^t by ▷ **Local model training**
      $\Theta_i^t \leftarrow \hat{\Theta}_i^t - \alpha \nabla_{\hat{\Theta}_i} \mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$.
    Client i sends Θ_i^t to the server. ▷ **Uploading**
  Server obtains Θ^t by $\Theta^t \leftarrow \sum_{i \in \mathcal{I}^t} \frac{k_i}{\sum_{j \in \mathcal{I}^t} k_j} \Theta_i^t$.
return $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

Fine-tune weights with only one step for adaptation

FedALA: overall algorithm

Algorithm 1: FedALA

Input: N clients, ρ: client joining ratio, $\mathcal{L}$: loss function, Θ^0: initial global model, α: local learning rate, η: the learning rate in ALA, $s\%$: the percent of local data in ALA, p: the range of ALA, $\sigma(\cdot)$: clip function.

Output: Reasonable local models $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

Server sends Θ^0 to all clients to initialize local models.
Clients initialize $W_i^p, \forall i \in [N]$ to ones.
for iteration $t = 1, \ldots, T$ **do**
    Server samples a subset $\mathcal{I}^t$ of clients according to ρ.
    Server sends Θ^{t-1} to $|\mathcal{I}^t|$ clients.
    for Client $i \in \mathcal{I}^t$ in parallel **do**
        Client i samples $s\%$ of local data. ▷ **ALA**
        if $t = 2$ **then** ▷ Start stage
            while W_i^p does not converge **do**
                Client i trains W_i^p by Equation (5).
                Client i clips W_i^p using $\sigma(\cdot)$.
        else if $t > 2$ **then**
            Client i trains W_i^p by Equation (5).
            Client i clips W_i^p using $\sigma(\cdot)$.
        Client i obtains $\hat{\Theta}_i^t$ by Equation (4).
        Client i obtains Θ_i^t by ▷ **Local model training**
            $\Theta_i^t \leftarrow \hat{\Theta}_i^t - \alpha \nabla_{\hat{\Theta}_i} \mathcal{L}(\hat{\Theta}_i^t, D_i; \Theta^{t-1})$.
        Client i sends Θ_i^t to the server. ▷ **Uploading**
    Server obtains Θ^t by $\Theta^t \leftarrow \sum_{i \in \mathcal{I}^t} \frac{k_i}{\sum_{j \in \mathcal{I}^t} k_j} \Theta_i^t$.
return $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

FedALA: overall algorithm

Algorithm 1: FedALA

Input: N clients, ρ: client joining ratio, $\mathcal{L}$: loss function, Θ^0: initial global model, α: local learning rate, η: the learning rate in ALA, $s\%$: the percent of local data in ALA, p: the range of ALA, $\sigma(\cdot)$: clip function.

Output: Reasonable local models $\hat{\Theta}_1, \ldots, \hat{\Theta}_N$

```
1:  Server sends Θ^0 to all clients to initialize local models.
2:  Clients initialize W_i^p, ∀i ∈ [N] to ones.
3:  for iteration t = 1, ..., T do
4:      Server samples a subset I^t of clients according to ρ.
5:      Server sends Θ^{t-1} to |I^t| clients.
6:      for Client i ∈ I^t in parallel do
7:          Client i samples s% of local data.              ▷ ALA
8:          if t = 2 then                                   ▷ Start stage
9:              while W_i^p does not converge do
10:                 Client i trains W_i^p by Equation (5).
11:                 Client i clips W_i^p using σ(·).
12:         else if t > 2 then
13:             Client i trains W_i^p by Equation (5).
14:             Client i clips W_i^p using σ(·).
15:         Client i obtains Θ̂_i^t by Equation (4).
16:         Client i obtains Θ_i^t by                       ▷ Local model training
                Θ_i^t ← Θ̂_i^t − α ∇_{Θ̂_i} L(Θ̂_i^t, D_i; Θ^{t-1}).
17:         Client i sends Θ_i^t to the server.             ▷ Uploading
18:     Server obtains Θ^t by Θ^t ← Σ_{i∈I^t} (k_i / Σ_{j∈I^t} k_j) Θ_i^t.
19: return Θ̂_1, ..., Θ̂_N
```

* **Capture desired information in global model**
without modifying other FL process

* **Reduce computation overhead with**
reused adaptive weights
small p (applying ALA on p higher layers)
small s (training weights with s% local data)

FedALA: results for computation reduction

- Reduce computation overhead with small $\boldsymbol{p}$ **(applying ALA on $\boldsymbol{p}$ higher layers)**

The test accuracy (%) and the number of trainable parameters (in millions) of FedALA on Tiny-ImageNet using ResNet-18 ($s = 80$)

	$p=6$	$p=5$	$p=4$	$p=3$	$p=2$	$p=1$
Acc.	41.71	41.54	41.62	41.86	**42.47**	41.94
Param.	11.182	11.172	11.024	10.499	8.399	0.005

Accuracy hardly changes with different $\boldsymbol{p}$

FedALA: results for computation reduction

- Reduce computation overhead with small p **(applying ALA on p higher layers)**

The test accuracy (%) and the number of trainable parameters (in millions) of FedALA on Tiny-ImageNet using ResNet-18 ($s = 80$)

	$p = 6$	$p = 5$	$p = 4$	$p = 3$	$p = 2$	$p = 1$
Acc.	41.71	41.54	41.62	41.86	**42.47**	41.94
Param.	11.182	11.172	11.024	10.499	8.399	0.005

Accuracy hardly changes with different p

Parameter amount decreases greatly with small p

FedALA: results for computation reduction

- Reduce computation overhead with small $\boldsymbol{p}$ **(applying ALA on $\boldsymbol{p}$ higher layers)**

The test accuracy (%) and the number of trainable parameters (in millions) of FedALA on Tiny-ImageNet using ResNet-18 ($s = 80$)

	$p=6$	$p=5$	$p=4$	$p=3$	$p=2$	$p=1$
Acc.	41.71	41.54	41.62	41.86	**42.47**	41.94
Param.	11.182	11.172	11.024	10.499	8.399	0.005

Accuracy hardly changes with different $\boldsymbol{p}$

Parameter amount decreases greatly with small $\boldsymbol{p}$

Set $\boldsymbol{p} = \mathbf{1}$ to greatly reduce computation overhead

FedALA: results for computation reduction

- Reduce computation overhead with small s **(training weights with s% local data)**

The test accuracy (%) of FedALA on Tiny-ImageNet
using ResNet-18 ($p = 1$)

	$s=5$	$s=10$	$s=20$	$s=40$	$s=60$	$s=80$	$s=100$
Acc.	39.53	40.62	40.02	40.23	41.11	41.94	**42.11**

Accuracy decreases with smaller s

FedALA: results for computation reduction

- Reduce computation overhead with small s **(training weights with s% local data)**

The test accuracy (%) of FedALA on Tiny-ImageNet
using ResNet-18 ($p = 1$)

	$s = 5$	$s = 10$	$s = 20$	$s = 40$	$s = 60$	$s = 80$	$s = 100$
Acc.	39.53	40.62	40.02	40.23	41.11	41.94	**42.11**

Accuracy decreases with smaller s

Set $s = 80$ to reduce computation overhead

FedALA: results for computation reduction

- Reduce computation overhead with small s **(training weights with s% local data)**

The test accuracy (%) of FedALA on Tiny-ImageNet
using ResNet-18 ($p = 1$)

	$s=5$	$s=10$	$s=20$	$s=40$	$s=60$	$s=80$	$s=100$
Acc.	39.53	40.62	40.02	40.23	41.11	41.94	**42.11**

Accuracy decreases with smaller s

Set $s = 80$ to reduce computation overhead

FedALA performs well with only 5% local data for ALA

FedALA: analysis

- Two main equations (omitting $\boldsymbol{p}$):

$$\underbrace{\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i}$$

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

FedALA: analysis

- Two main equations (omitting p):

$$\boxed{\begin{gathered}\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i \\ W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})\end{gathered}}$$

Denote $\mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$ as $\mathcal{L}_i^t$

- Rewrite the **gradient term** as $\nabla_{W_i} \mathcal{L}_i^t = \eta(\Theta^{t-1} - \Theta_i^{t-1}) \odot \nabla_{\hat{\Theta}_i} \mathcal{L}_i^t$

FedALA: analysis

- Two main equations (omitting p):

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

Denote $\mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$ as $\mathcal{L}_i^t$

- Rewrite the gradient term as $\nabla_{W_i} \mathcal{L}_i^t = \eta(\Theta^{t-1} - \Theta_i^{t-1}) \odot \nabla_{\hat{\Theta}_i} \mathcal{L}_i^t$

- We view updating W_i as updating $\hat{\Theta}_i^t$

$$\hat{\Theta}_i^t \leftarrow \hat{\Theta}_i^t - \eta(\Theta^{t-1} - \Theta_i^{t-1}) \odot (\Theta^{t-1} - \Theta_i^{t-1}) \odot \nabla_{\hat{\Theta}_i} \mathcal{L}_i^t$$

FedALA: analysis

- Two main equations (omitting p):

$$\hat{\Theta}_i^t := \Theta_i^{t-1} + (\Theta^{t-1} - \Theta_i^{t-1}) \odot W_i$$

$$W_i^p \leftarrow W_i^p - \eta \nabla_{W_i^p} \mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$$

Denote $\mathcal{L}(\hat{\Theta}_i^t, D_i^{s,t}; \Theta^{t-1})$ as $\mathcal{L}_i^t$

- Rewrite the gradient term as $\nabla_{W_i} \mathcal{L}_i^t = \eta(\Theta^{t-1} - \Theta_i^{t-1}) \odot \nabla_{\hat{\Theta}_i} \mathcal{L}_i^t$

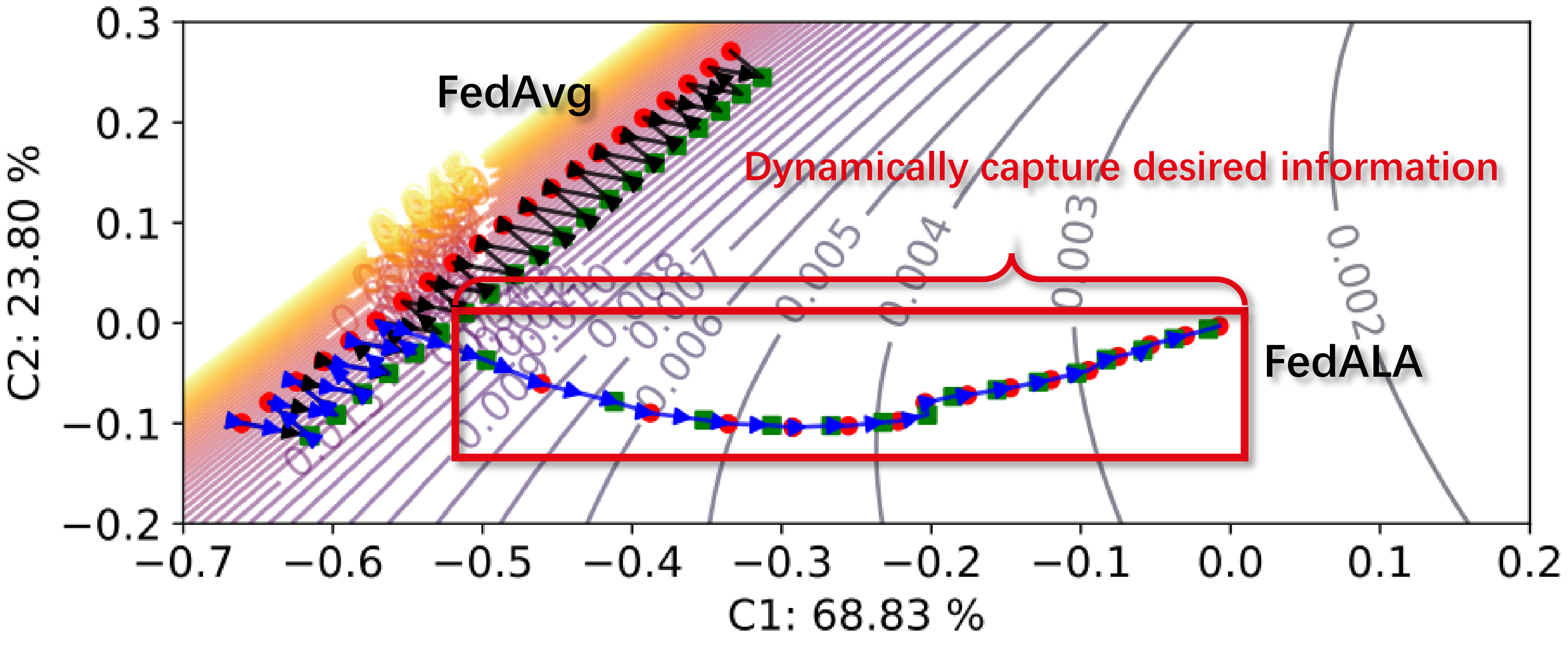
- We view updating W_i as updating $\hat{\Theta}_i^t$

$$\hat{\Theta}_i^t \leftarrow \hat{\Theta}_i^t - \eta \underbrace{(\Theta^{t-1} - \Theta_i^{t-1}) \odot (\Theta^{t-1} - \Theta_i^{t-1})} \odot \nabla_{\hat{\Theta}_i} \mathcal{L}_i^t$$

*** Dynamic generic information**

FedALA: overview (recall)

- Learning trajectory on one client: **FedAvg** vs. **FedALA**
- Activate **ALA** in the subsequent iterations



FedALA: applicability of ALA module

- Applying **ALA** to other FL methods

FedALA: applicability of ALA module

- Applying **ALA** to other FL methods
  - only modifies the local initialization process

FedALA: applicability of ALA module

- Applying **ALA** to other FL methods
  - only modifies the local initialization process

The test accuracy (%) and improvement (%)

	Datasets	Tiny-ImageNet		Cifar100	
	Methods	Acc.	Imps.	Acc.	Imps.
Traditional FL	FedAvg+**ALA**	40.54±0.17	21.08	55.92±0.15	24.03
	FedProx+**ALA**	40.53±0.26	21.16	56.18±0.65	24.19
Personalized FL	Per-FedAvg+**ALA**	30.90±0.28	5.83	48.68±0.36	4.40
	FedRep+**ALA**	37.89±0.31	0.62	53.02±0.11	0.63
	pFedMe+**ALA**	27.30±0.24	0.37	47.91±0.21	0.57
	Ditto+**ALA**	40.75±0.06	8.60	56.33±0.07	3.46
	FedAMP+**ALA**	28.18±0.20	0.19	48.03±0.23	0.34
	FedPHP+**ALA**	40.16±0.24	4.47	54.28±0.21	3.76
	PartialFed+**ALA**	35.40±0.02	0.14	48.99±0.05	0.18

Performance comparison

- FedALA requires **less computation** than most FL methods

The computation and communication overhead, $M = 20$.

	Computation		Communication
	Total time	Time/iter.	Param./iter.
FedAvg	365 min	1.59 min	$2 * \Sigma$
FedProx	325 min	1.99 min	$2 * \Sigma$
FedAvg-C	607 min	24.28 min	$2 * \Sigma$
FedProx-C	711 min	28.44 min	$2 * \Sigma$
Per-FedAvg	121 min	3.56 min	$2 * \Sigma$
FedRep	471 min	4.09 min	$2 * \alpha_f * \Sigma$
pFedMe	1157 min	10.24 min	$2 * \Sigma$
Ditto	318 min	11.78 min	$2 * \Sigma$
FedAMP	92 min	1.53 min	$2 * \Sigma$
FedPHP	264 min	4.06 min	$2 * \Sigma$
FedFomo	193 min	2.72 min	$(1 + M) * \Sigma$
APPLE	132 min	2.93 min	$(1 + M) * \Sigma$
PartialFed	693 min	2.13 min	$2 * \Sigma$
FedALA	7+116 min	1.93 min	$2 * \Sigma$

Performance comparison

- Compared to FedAvg, FedALA **does not introduce additional communication** per iteration

The computation and communication overhead, $M = 20$.

	Computation		Communication
	Total time	Time/iter.	Param./iter.
FedAvg	365 min	1.59 min	$2 * \Sigma$
FedProx	325 min	1.99 min	$2 * \Sigma$
FedAvg-C	607 min	24.28 min	$2 * \Sigma$
FedProx-C	711 min	28.44 min	$2 * \Sigma$
Per-FedAvg	121 min	3.56 min	$2 * \Sigma$
FedRep	471 min	4.09 min	$2 * \alpha_f * \Sigma$
pFedMe	1157 min	10.24 min	$2 * \Sigma$
Ditto	318 min	11.78 min	$2 * \Sigma$
FedAMP	92 min	1.53 min	$2 * \Sigma$
FedPHP	264 min	4.06 min	$2 * \Sigma$
FedFomo	193 min	2.72 min	$(1 + M) * \Sigma$
APPLE	132 min	2.93 min	$(1 + M) * \Sigma$
PartialFed	693 min	2.13 min	$2 * \Sigma$
FedALA	7+116 min	1.93 min	$2 * \Sigma$

Performance comparison

- Compared to FedAvg, FedALA **does not introduce additional communication** per iteration
  - but **costs fewer iterations** to converge

The computation and communication overhead, $M = 20$.

	Computation		Communication
	Total time	Time/iter.	Param./iter.
FedAvg	365 min	1.59 min	$2 * \Sigma$
FedProx	325 min	1.99 min	$2 * \Sigma$
FedAvg-C	607 min	24.28 min	$2 * \Sigma$
FedProx-C	711 min	28.44 min	$2 * \Sigma$
Per-FedAvg	121 min	3.56 min	$2 * \Sigma$
FedRep	471 min	4.09 min	$2 * \alpha_f * \Sigma$
pFedMe	1157 min	10.24 min	$2 * \Sigma$
Ditto	318 min	11.78 min	$2 * \Sigma$
FedAMP	92 min	1.53 min	$2 * \Sigma$
FedPHP	264 min	4.06 min	$2 * \Sigma$
FedFomo	193 min	2.72 min	$(1 + M) * \Sigma$
APPLE	132 min	2.93 min	$(1 + M) * \Sigma$
PartialFed	693 min	2.13 min	$2 * \Sigma$
FedALA	7+116 min	1.93 min	$2 * \Sigma$

Performance comparison

- Compared to FedFomo and APPLE, FedALA **requires less communication** per iteration

The computation and communication overhead, $M = 20$.

	Computation		Communication
	Total time	Time/iter.	Param./iter.
FedAvg	365 min	1.59 min	$2 * \Sigma$
FedProx	325 min	1.99 min	$2 * \Sigma$
FedAvg-C	607 min	24.28 min	$2 * \Sigma$
FedProx-C	711 min	28.44 min	$2 * \Sigma$
Per-FedAvg	121 min	3.56 min	$2 * \Sigma$
FedRep	471 min	4.09 min	$2 * \alpha_f * \Sigma$
pFedMe	1157 min	10.24 min	$2 * \Sigma$
Ditto	318 min	11.78 min	$2 * \Sigma$
FedAMP	92 min	1.53 min	$2 * \Sigma$
FedPHP	264 min	4.06 min	$2 * \Sigma$
FedFomo	193 min	2.72 min	$(1 + M) * \Sigma$
APPLE	132 min	2.93 min	$(1 + M) * \Sigma$
PartialFed	693 min	2.13 min	$2 * \Sigma$
FedALA	7+116 min	1.93 min	$2 * \Sigma$

Performance comparison

- FedALA outperforms **11** SOTA traditional FL and pFL methods

The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Performance comparison

- FedALA outperforms **2** fine-tuning-based pFL methods

The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Performance comparison

- FedALA outperforms 13 traditional FL and pFL methods
  - in various settings

The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Performance comparison

- FedALA outperforms 13 traditional FL and pFL methods
  - in various settings and various datasets (CV and NLP domains)

The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Performance comparison

- FedALA outperforms 13 traditional FL and pFL methods by up to **3.27%**

The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Performance comparison

- FedALA outperforms 13 traditional FL and pFL methods by up to **3.27%**
  - For more results, please refer to our paper

The test accuracy (%) in the pathological heterogeneous setting and practical heterogeneous setting.

Settings	Pathological heterogeneous setting			Practical heterogeneous setting				
Methods	MNIST	Cifar10	Cifar100	Cifar10	Cifar100	TINY	TINY*	AG News
FedAvg	97.93±0.05	55.09±0.83	25.98±0.13	59.16±0.47	31.89±0.47	19.46±0.20	19.45±0.13	79.57±0.17
FedProx	98.01±0.09	55.06±0.75	25.94±0.16	59.21±0.40	31.99±0.41	19.37±0.22	19.27±0.23	79.35±0.23
FedAvg-C	99.79±0.00	92.13±0.03	66.17±0.03	90.34±0.01	51.80±0.02	30.67±0.08	36.94±0.10	95.89±0.25
FedProx-C	99.80±0.04	92.12±0.03	66.07±0.08	90.33±0.01	51.84±0.07	30.77±0.13	38.78±0.52	96.10±0.22
Per-FedAvg	99.63±0.02	89.63±0.23	56.80±0.26	87.74±0.19	44.28±0.33	25.07±0.07	21.81±0.54	93.27±0.25
FedRep	99.77±0.03	91.93±0.14	67.56±0.31	90.40±0.24	52.39±0.35	37.27±0.20	39.95±0.61	96.28±0.14
pFedMe	99.75±0.02	90.11±0.10	58.20±0.14	88.09±0.32	47.34±0.46	26.93±0.19	33.44±0.33	91.41±0.22
Ditto	99.81±0.00	92.39±0.06	67.23±0.07	90.59±0.01	52.87±0.64	32.15±0.04	35.92±0.43	95.45±0.17
FedAMP	99.76±0.02	90.79±0.16	64.34±0.37	88.70±0.18	47.69±0.49	27.99±0.11	29.11±0.15	94.18±0.09
FedPHP	99.73±0.00	90.01±0.00	63.09±0.04	88.92±0.02	50.52±0.16	35.69±3.26	29.90±0.51	94.38±0.12
FedFomo	99.83±0.00	91.85±0.02	62.49±0.22	88.06±0.02	45.39±0.45	26.33±0.22	26.84±0.11	95.84±0.15
APPLE	99.75±0.01	90.97±0.05	65.80±0.08	89.37±0.11	53.22±0.20	35.04±0.47	39.93±0.52	95.63±0.21
PartialFed	99.86±0.01	89.60±0.13	61.39±0.12	87.38±0.08	48.81±0.20	35.26±0.18	37.50±0.16	85.20±0.16
FedALA	**99.88±0.01**	**92.44±0.02**	**67.83±0.06**	**90.67±0.03**	**55.92±0.03**	**40.54±0.02**	**41.94±0.05**	**96.52±0.08**

Summary

- **Contributions** of FedALA:
  - **Adaptively aggregates** the global model and local model towards the local objective to **capture the desired information** from the global model
  - Outperforms **11 SOTA** methods by up to 3.27% in test accuracy **without additional communication overhead** in each iteration
  - The ALA module in FedALA **can be directly applied to existing FL methods to enhance their performance** by up to 24.19%

- **Resources**:
  - Full paper: https://arxiv.org/abs/2212.01197
  - Code: https://github.com/TsingZ0/FedALA

FedALA: Adaptive Local Aggregation for Personalized Federated Learning

Full paper: https://arxiv.org/abs/2212.01197

Code: https://github.com/TsingZ0/FedALA

Thanks!