南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Model-Contrastive Federated Learning

CVPR 2021

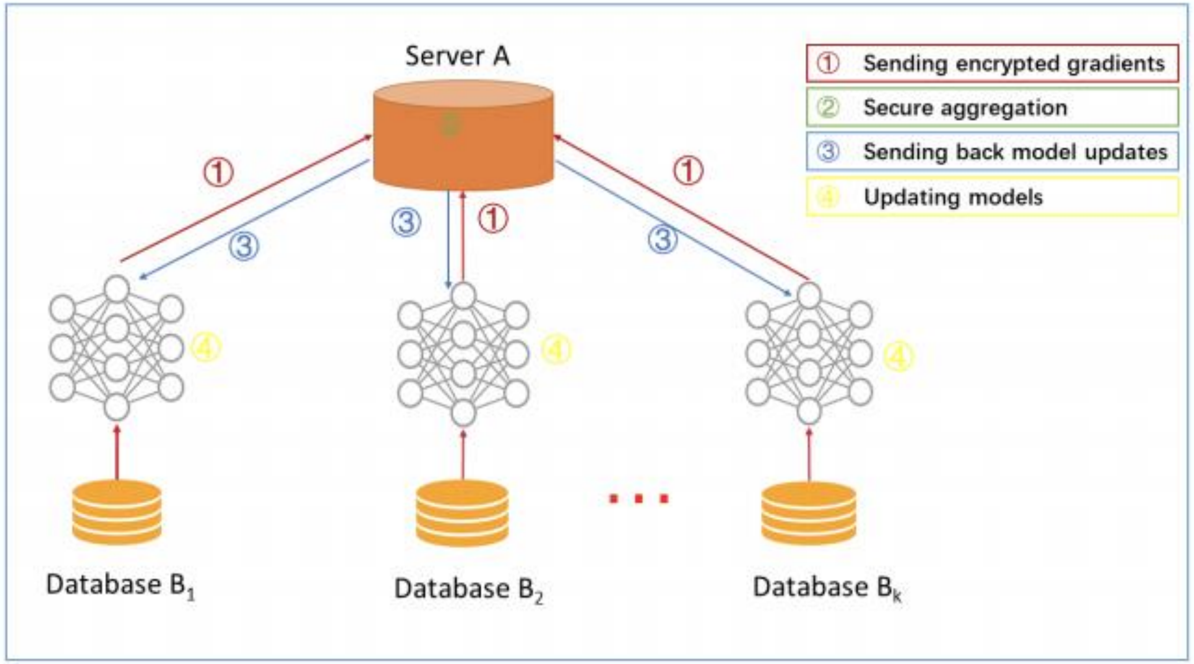
FEDMIX: APPROXIMATION OF MIXUP UNDER MEAN AUGMENTED FEDERATED LEARNING

ICLR 2021

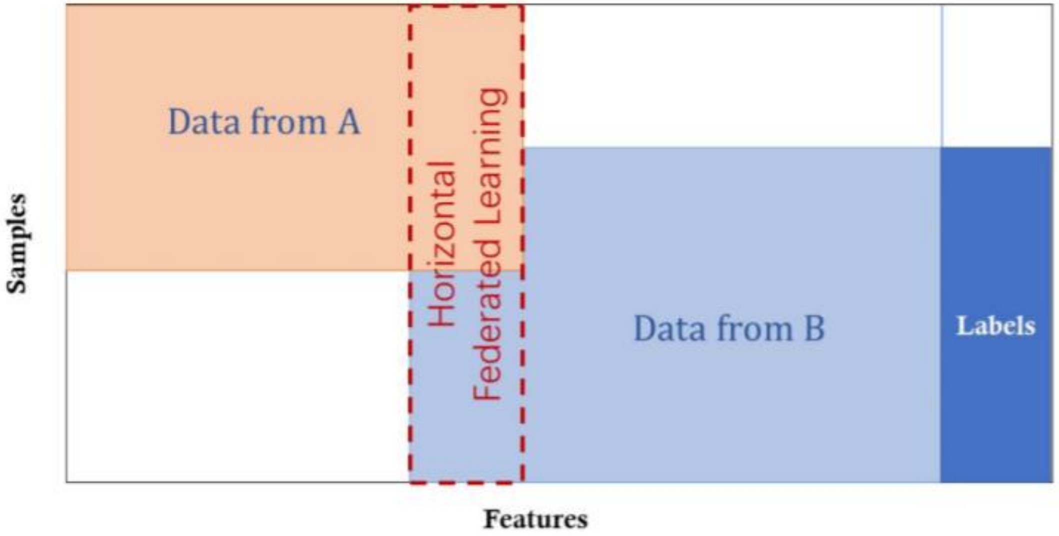
FEDERATED SEMI-SUPERVISED LEARNING WITH INTER-CLIENT CONSISTENCY & DISJOINT LEARNING

ICLR 2021

Federated Learning



Horizontal Federated Learning

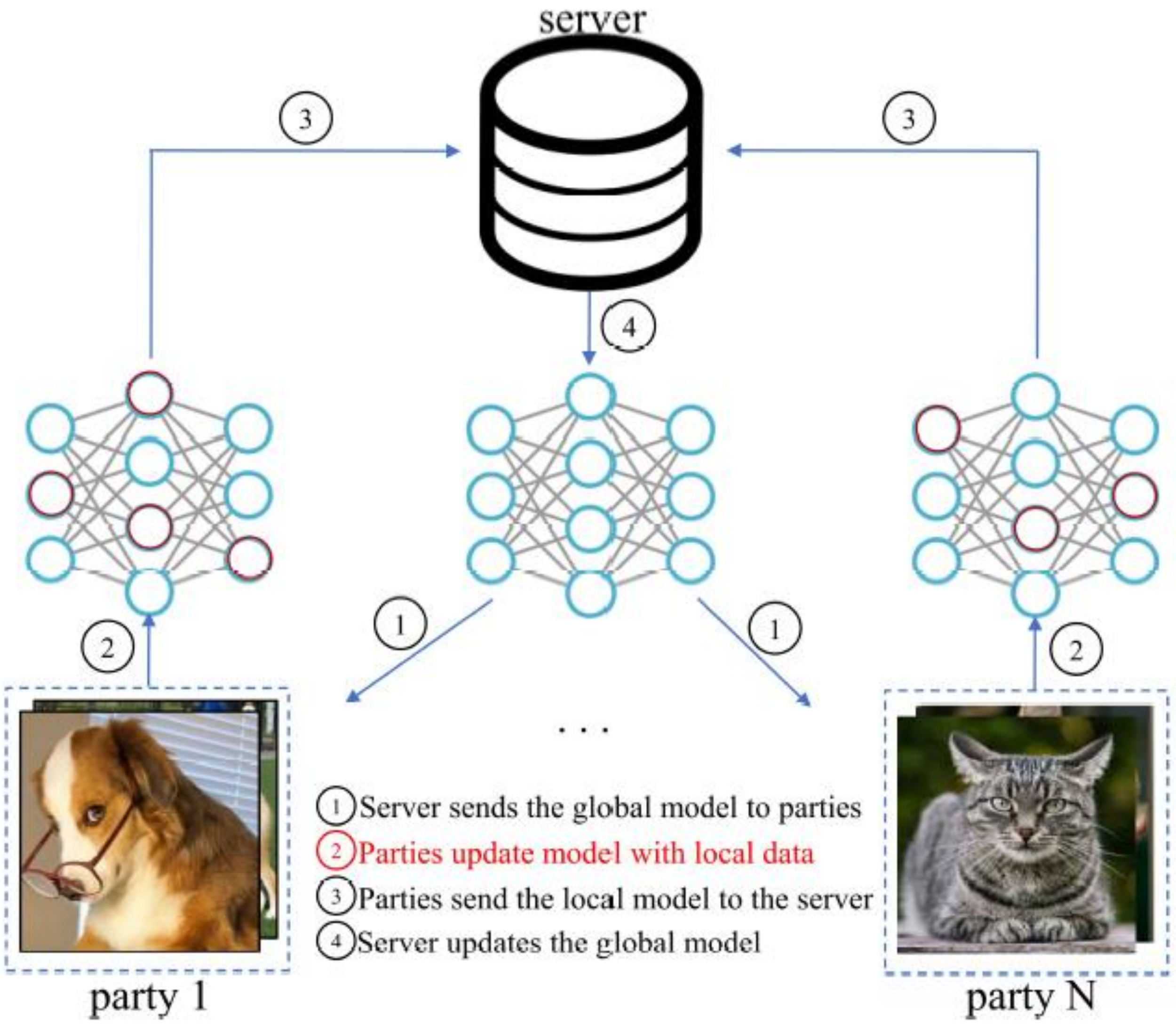


$$\mathcal{X}_i = \mathcal{X}_j,\ \ \mathcal{Y}_i = \mathcal{Y}_j,\ \ I_i \neq I_j,\ \ \forall \mathcal{D}_i, \mathcal{D}_j, i \neq j.$$

Example: Two regional banks may have very different user groups from their respective regions, and the intersection set of their users is very small. However, their business is very similar, so the feature spaces are the same.

http://tangyp.cn/seminar_hsj/2021-spring/20210602-tangyp.pdf

Federated Learning



Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w)**:** *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

FedAvg: One of the standard and most widely used algorithm for federated learning.

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Model-Contrastive Federated Learning

Qinbin Li
National University of Singapore
qinbin@comp.nus.edu.sg

Bingsheng He
National University of Singapore
hebs@comp.nus.edu.sg

Dawn Song
UC Berkeley
dawnsong@berkeley.edu

CVPR 2021

Motivation

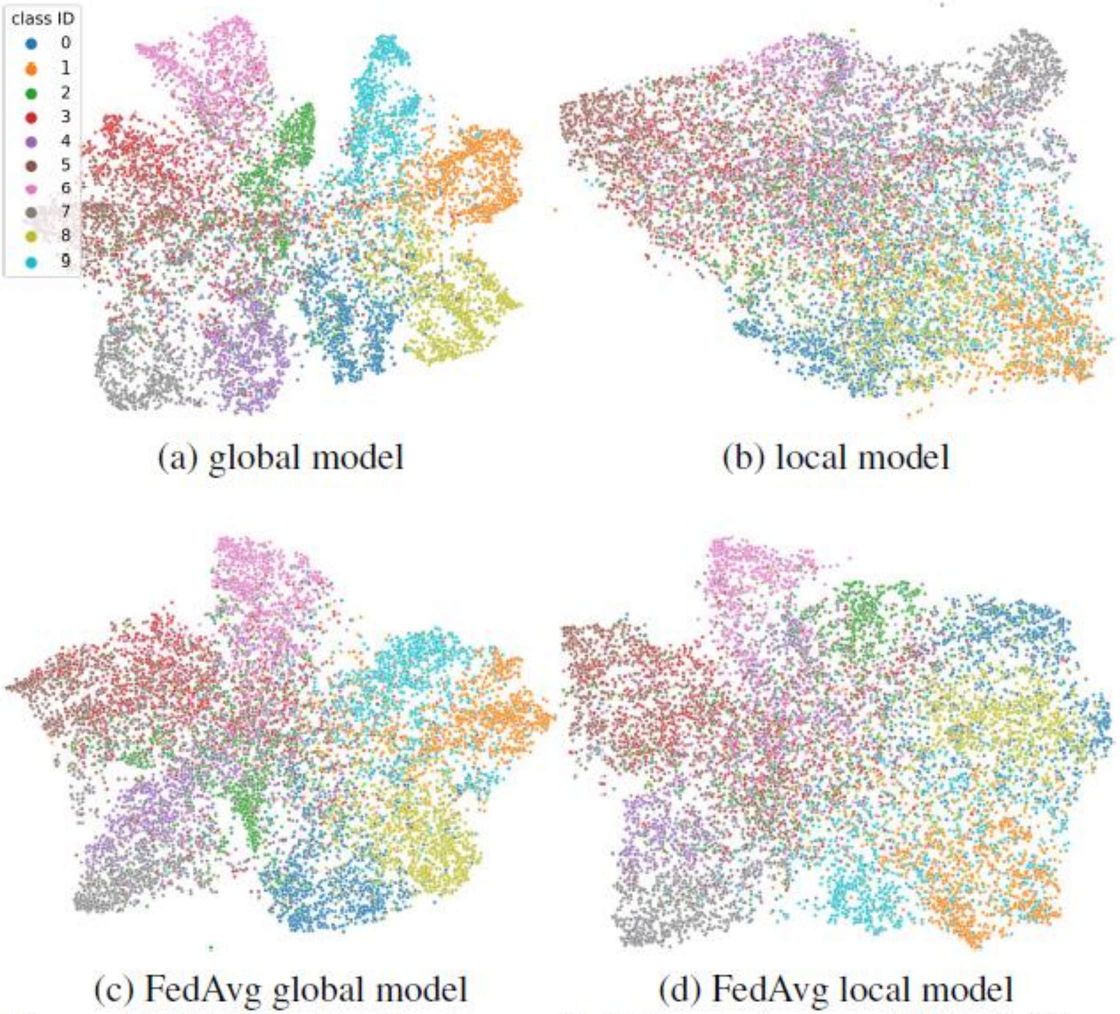
A key challenge in federated learning is to handle the **heterogeneity of local data distribution** across parties.

Existing methods failed to achieve high performance in **image datasets** with **deep learning models**.

⇩

MOON (**mo**del-**con**trastive learning) is proposed.

MOON is based on an intuitive idea: the model trained on the whole dataset is able to extract a better feature representation than the model trained on a skewed subset.



(a) global model (b) local model

(c) FedAvg global model (d) FedAvg local model

Figure 2. T-SNE visualizations of hidden vectors on CIFAR-10.

Method

MOON:

- ✓ **Decreasing the distance between** the representation learned by the **local model and** the representation learned by the **global model**;
- ✓ **Increasing the distance between** the representation learned by the **local model and** the representation learned by the **previous local model**.

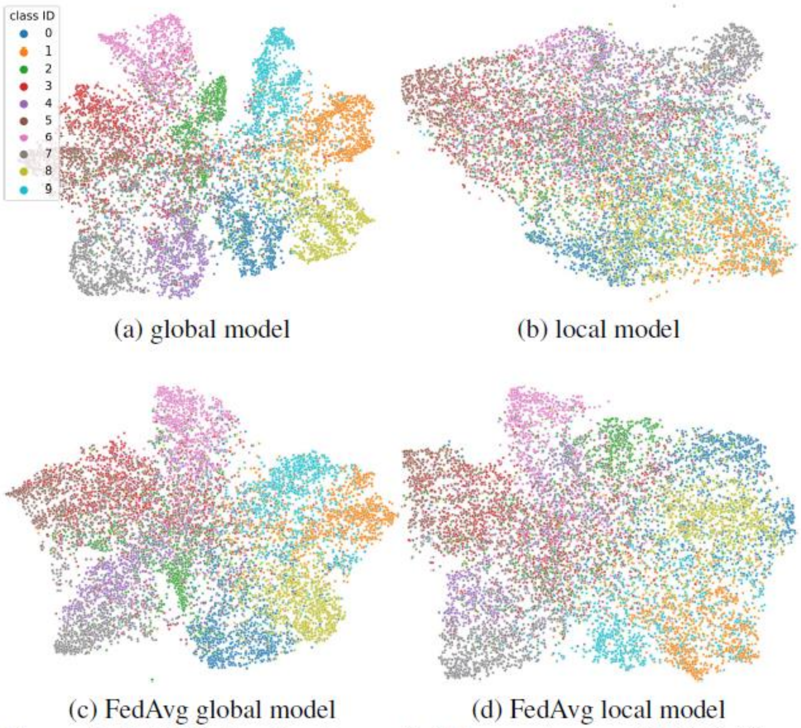


Figure 2. T-SNE visualizations of hidden vectors on CIFAR-10.

$$\Rightarrow \quad l_{i,j} = -\log \frac{\exp(\mathrm{sim}(x_i, x_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\mathrm{sim}(x_i, x_k)/\tau)}$$

Method

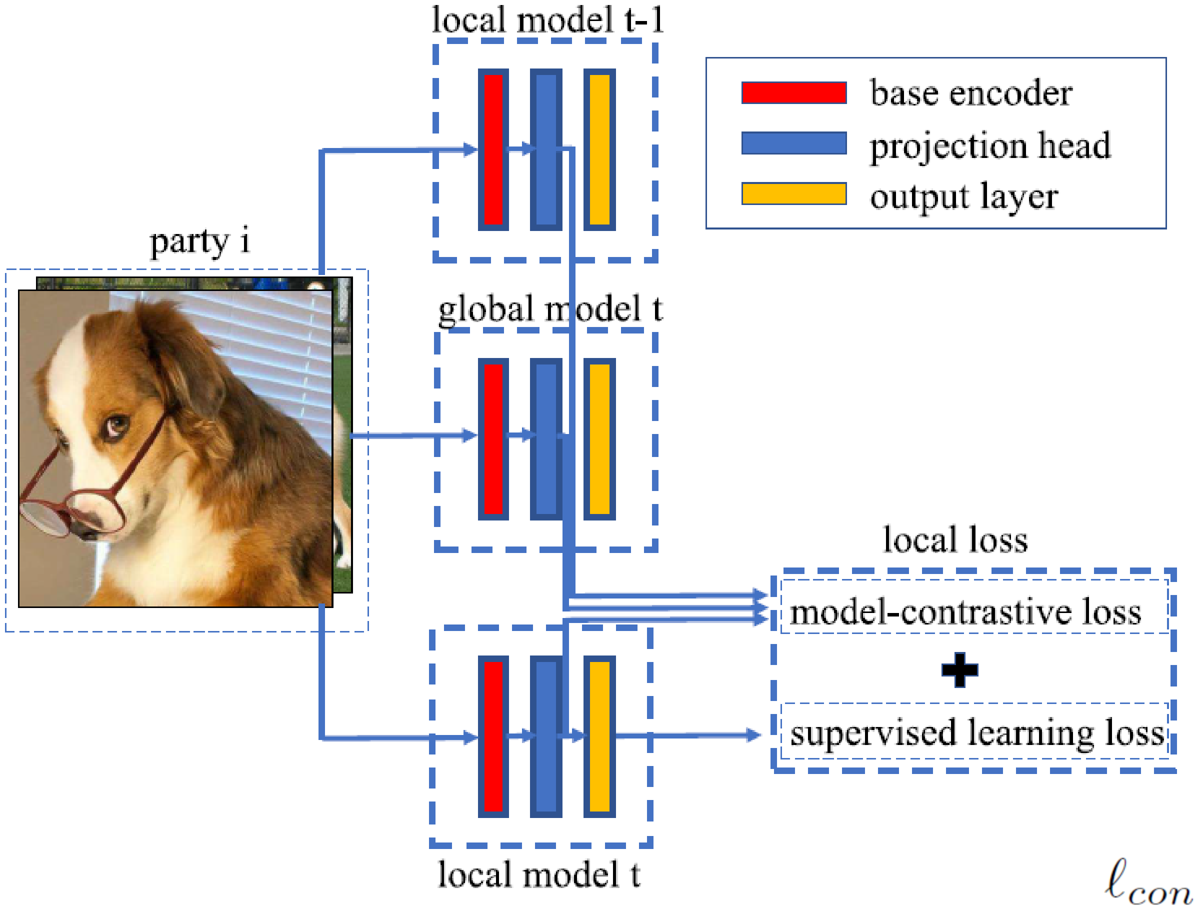
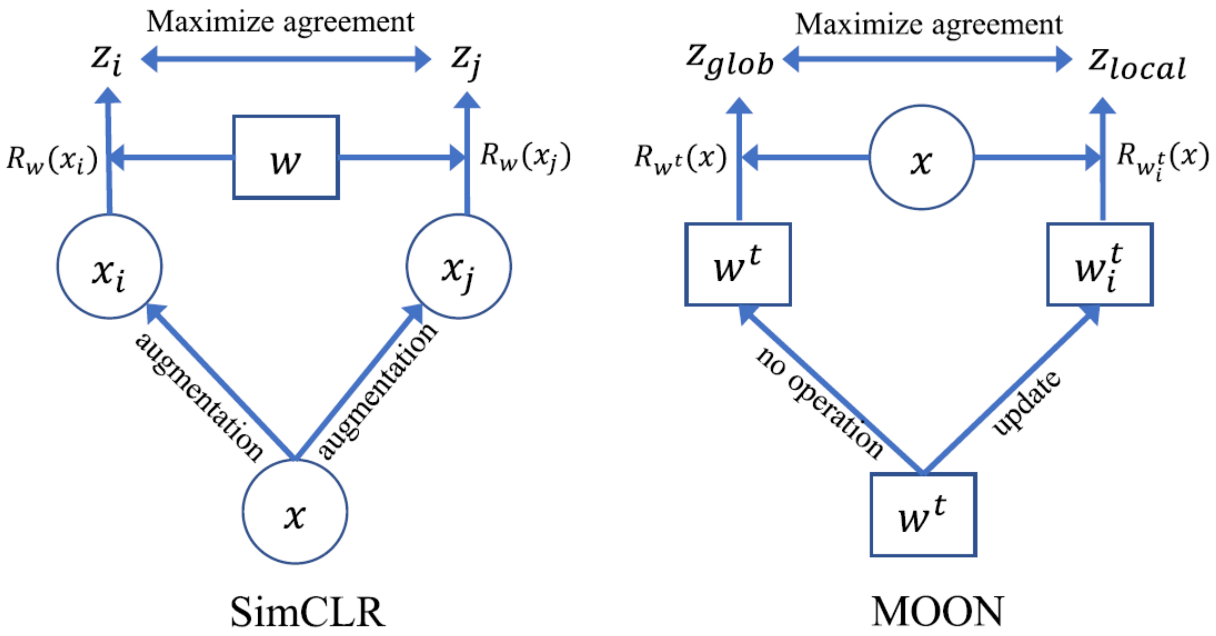


Figure 3. The local loss in MOON.



$$\ell_{con} = -\log \frac{\exp(\mathrm{sim}(z, z_{glob})/\tau)}{\exp(\mathrm{sim}(z, z_{glob})/\tau) + \exp(\mathrm{sim}(z, z_{prev})/\tau)}$$

$$\ell = \ell_{sup}(w_i^t; (x, y)) + \mu \ell_{con}(w_i^t; w_i^{t-1}; w^t; x)$$

Method

Algorithm 1: The MOON framework

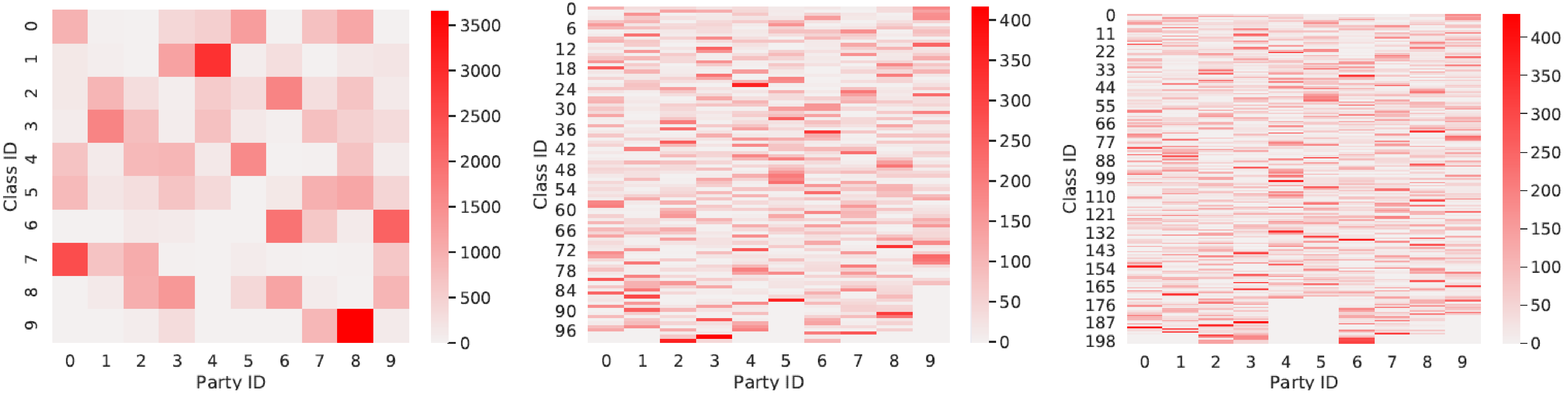
Input: number of communication rounds T, number of parties N, number of local epochs E, temperature τ, learning rate η, hyper-parameter μ

Output: The final model w^T

Server executes:
initialize w^0
for $t = 0, 1, ..., T-1$ **do**
    for $i = 1, 2, ..., N$ **in parallel do**
        send the global model w^t to P_i
        $w_i^t \leftarrow$ **PartyLocalTraining**(i, w^t)
    $w^{t+1} \leftarrow \sum_{k=1}^{N} \frac{|\mathcal{D}^i|}{|\mathcal{D}|} w_k^t$
return w^T

PartyLocalTraining(i, w^t):
$w_i^t \leftarrow w^t$
for epoch $i = 1, 2, ..., E$ **do**
    for each batch $\mathbf{b} = \{x, y\}$ of $\mathcal{D}^i$ **do**
        $\ell_{sup} \leftarrow CrossEntropyLoss(F_{w_i^t}(x), y)$
        $z \leftarrow R_{w_i^t}(x)$
        $z_{glob} \leftarrow R_{w^t}(x)$
        $z_{prev} \leftarrow R_{w_i^{t-1}}(x)$
        $\ell_{con} \leftarrow -\log \frac{\exp(\mathrm{sim}(z, z_{glob})/\tau)}{\exp(\mathrm{sim}(z, z_{glob})/\tau) + \exp(\mathrm{sim}(z, z_{prev})/\tau)}$
        $\ell \leftarrow \ell_{sup} + \mu \ell_{con}$
        $w_i^t \leftarrow w_i^t - \eta \nabla \ell$
return w_i^t to server

Experiment



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-Imagenet

Figure 5. The data distribution of each party using non-IID data partition. The color bar denotes the number of data samples. Each rectangle represents the number of data samples of a specific class in a party.

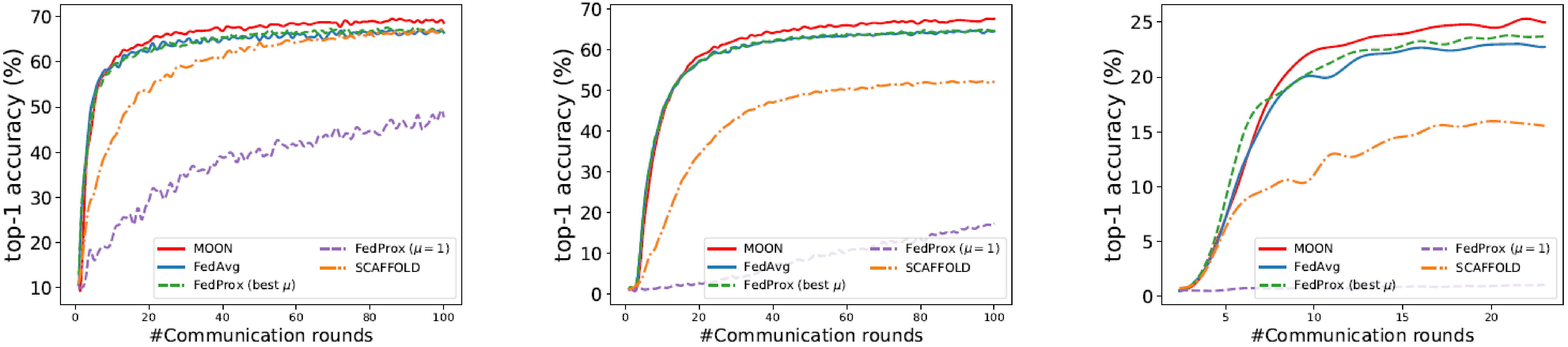
Experiment

南京航空航天大学
Nanjing University of Aeronautics and Astronautics

Table 1. The top-1 accuracy of MOON and the other baselines on test datasets. For MOON, FedAvg, FedProx, and SCAFFOLD, we run three trials and report the mean and standard derivation. For SOLO, we report the mean and standard derivation among all parties.

Method	CIFAR-10	CIFAR-100	Tiny-Imagenet
MOON	**69.1%**±0.4%	**67.5%**±0.4%	**25.1%**±0.1%
FedAvg	66.3%±0.5%	64.5% ±0.4%	23.0%±0.1%
FedProx	66.9%±0.2%	64.6%±0.2%	23.2%±0.2%
SCAFFOLD	66.6%±0.2%	52.5% ±0.3%	16.0%±0.2%
SOLO	46.3% ±5.1%	22.3%±1.0%	8.6%±0.4%

Experiment



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-Imagenet

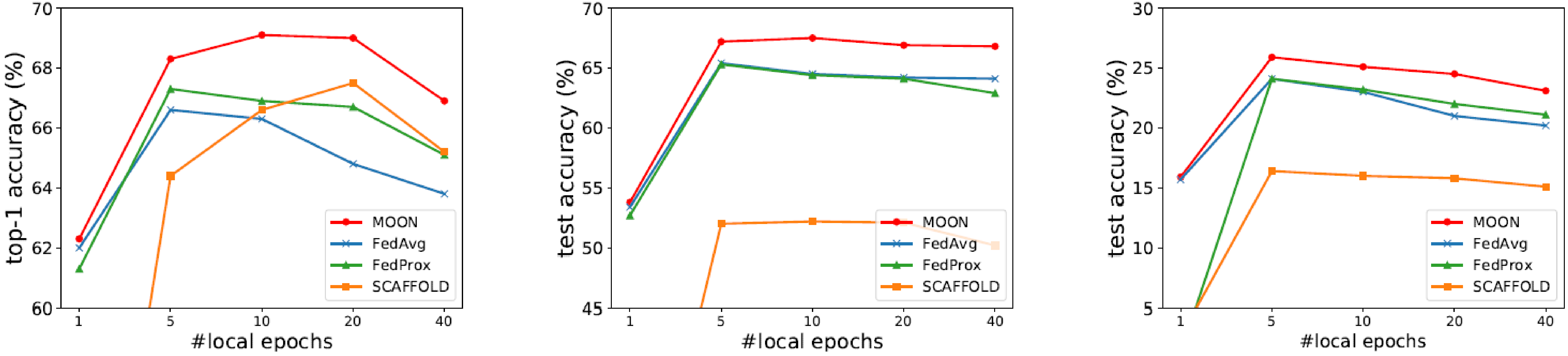
Figure 6. The top-1 test accuracy in different number of communication rounds. For FedProx, we report both the accuracy with best μ and the accuracy with $\mu = 1$.

Experiment

Table 2. The number of rounds of different approaches to achieve the same accuracy as running FedAvg for 100 rounds (CIFAR-10/100) or 20 rounds (Tiny-Imagenet). The speedup of an approach is computed against FedAvg.

Method	CIFAR-10		CIFAR-100		Tiny-Imagenet	
	#rounds	speedup	#rounds	speedup	#rounds	speedup
FedAvg	100	1×	100	1×	20	1×
FedProx	52	1.9×	75	1.3×	17	1.2×
SCAFFOLD	80	1.3×	\	<1×	\	<1×
MOON	**27**	**3.7**×	**43**	**2.3**×	**11**	**1.8**×

Experiment



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-Imagenet

Figure 7. The top-1 test accuracy with different number of local epochs. For MOON and FedProx, μ is set to the best μ from Section 4.2 for all numbers of local epochs. The accuracy of SCAFFOLD is quite bad when number of local epochs is set to 1 (45.3% on CIFAR10, 20.4% on CIFAR-100, 2.6% on Tiny-Imagenet). The accuracy of FedProx on Tiny-Imagenet with one local epoch is 1.2%.

Experiment

Table 3. The accuracy with 50 parties and 100 parties (sample fraction=0.2) on CIFAR-100.

Method	#parties=50		#parties=100	
	100 rounds	200 rounds	250 rounds	500 rounds
MOON (μ=1)	54.7%	58.8%	54.5%	58.2%
MOON (μ=10)	**58.2%**	**63.2%**	**56.9%**	**61.8%**
FedAvg	51.9%	56.4%	51.0%	55.0%
FedProx	52.7%	56.6%	51.3%	54.6%
SCAFFOLD	35.8%	44.9%	37.4%	44.5%
SOLO	10%±0.9%		7.3%±0.6%	

Experiment

Table 4. The test accuracy with β from $\{0.1, 0.5, 5\}$.

Method	$\beta = 0.1$	$\beta = 0.5$	$\beta = 5$
MOON	**64.0%**	**67.5%**	**68.0%**
FedAvg	62.5%	64.5%	65.7%
FedProx	62.9%	64.6%	64.9%
SCAFFOLD	47.3%	52.5%	55.0%
SOLO	15.9%±1.5%	22.3%±1%	26.6%±1.4%

Experiment

Table 5. The top-1 accuracy with different kinds of loss for the second term of local objective. We tune μ from $\{0.001, 0.01, 0.1, 1, 5, 10\}$ for the ℓ_2 norm approach and report the best accuracy.

second term	CIFAR-10	CIFAR-100	Tiny-Imagenet
none (FedAvg)	66.3%	64.5%	23.0%
ℓ_2 norm	65.8%	66.9%	24.0%
MOON	**69.1%**	**67.5%**	**25.1%**

$$\ell = \ell_{sup} + \mu \left\| z - z_{glob} \right\|_2$$

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

FedMix: Approximation of Mixup under Mean Augmented Federated Learning

Tehrim Yoon & Sumin Shin & Sung Ju Hwang & Eunho Yang
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, South Korea
{tryoon93,sym807,sjhwang82,eunhoy}@kaist.ac.kr

ICLR 2021

Method

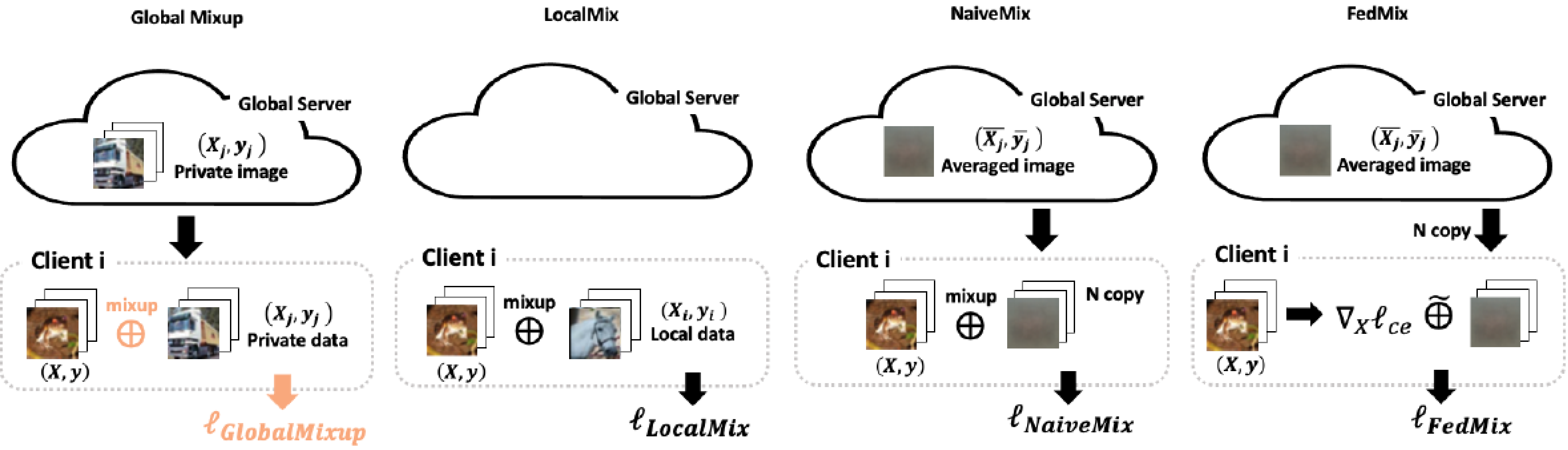
Mix-Up:

- Given two natural images x_i and x_j , mix-up generates multiple synthetic images by a convex combination of the two with different coefficients,

$$\hat{x}_{ij}(\lambda) = \lambda x_i + (1 - \lambda)x_j,$$

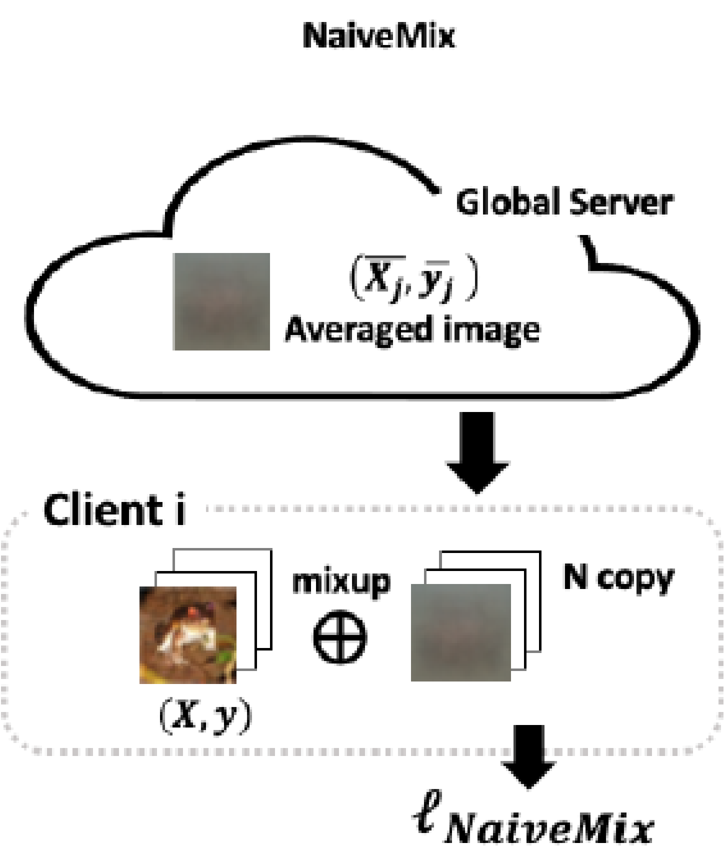
- where the coefficient $\lambda \in [0, 1]$. Note that this notation also includes the original unlabeled data x_i and x_j when $\lambda = 1$ and $\lambda = 0$, respectively.

Method



(a) Global Mixup (b) Local Mixup (c) NaiveMix (d) FedMix

Method



(c) NaiveMix

$$\ell_{\mathtt{NaiveMix}} = (1-\lambda)\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i + \lambda\bar{\boldsymbol{x}}_j\big), y_i\Big) + \lambda\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i + \lambda\bar{\boldsymbol{x}}_j\big), \bar{y}_j\Big)$$

Algorithm 1: Mean Augmented Federated Learning (MAFL)

Input: $\mathbb{D}_k = \{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ for $k = 1, \ldots, N$

M_k: number of data instances used for computing average $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$

Initialize w_0 for global server

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

  for *client k with updated local data* **do**

    Split local data into M_k sized batches

    Compute $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ for each batch

    Send all $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ to server

  end

  $\mathbb{S}_t \leftarrow K$clients selected at random

  Send $\boldsymbol{w}_t$ to clients $k \in \mathbb{S}_t$

  if *updated* **then**

    Aggregate all $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ to $\boldsymbol{X}_g, \boldsymbol{Y}_g$

    Send $\boldsymbol{X}_g, \boldsymbol{Y}_g$ to clients $k \in \mathbb{S}_t$

  end

  for $k \in \mathbb{S}_t$ **do**

    $\boldsymbol{w}_{t+1}^k \leftarrow LocalUpdate(k, \boldsymbol{w}_t; \boldsymbol{X}_g, \boldsymbol{Y}_g)$

  end

  $\boldsymbol{w}_{t+1} \leftarrow \frac{1}{K}\sum_{k\in\mathbb{S}_t} p_k \boldsymbol{w}_{t+1}^k$

end

Method

Proposition 1 *Consider the loss function of the global Mixup modulo the privacy issues,*

$$\ell_{\texttt{GlobalMixup}}(f(\tilde{\boldsymbol{x}}), \tilde{y}) = \ell\Big(f\big((1-\lambda)\boldsymbol{x}_i + \lambda \boldsymbol{x}_j\big), (1-\lambda)y_i + \lambda y_j\Big) \tag{3}$$

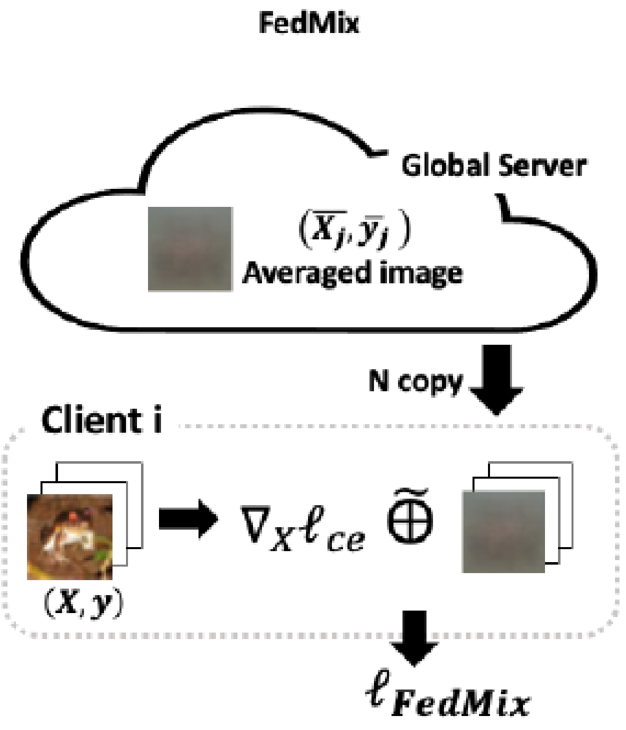
for cross-entropy loss ℓ[1]*. Suppose that Eq. (3) is approximated by applying Taylor series around the place where* $\lambda \ll 1$*. Then, if we ignore the second order term (i.e.,* $\mathcal{O}(\lambda^2)$*), we obtain the following approximated loss:*

$$(1-\lambda)\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_i\Big) + \lambda\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_j\Big) + \lambda\frac{\partial \ell}{\partial \boldsymbol{x}} \cdot \boldsymbol{x}_j \tag{4}$$

where the derivative $\frac{\partial \ell}{\partial \boldsymbol{x}}$ *is evaluated at* $\boldsymbol{x} = (1-\lambda)\boldsymbol{x}_i$ *and* $y = y_i$*.*

$$(1-\lambda)\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_i\Big) + (1-\lambda) \times \frac{\partial \ell}{\partial \boldsymbol{x}}\Big|_{(1-\lambda)\boldsymbol{x}_i, y_i} \cdot (\lambda \boldsymbol{x}_j)$$
$$+\lambda\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_j\Big) + \lambda \times \frac{\partial \ell}{\partial \boldsymbol{x}}\Big|_{(1-\lambda)\boldsymbol{x}_i, y_j} \cdot (\lambda \boldsymbol{x}_j).$$

Method



(d) FedMix

$$\ell_{\texttt{FedMix}} = \frac{1}{|J|}\sum_{j\in J}(1-\lambda)\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_i\Big) + \lambda\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_j\Big) + \lambda\frac{\partial \ell}{\partial \boldsymbol{x}}\cdot \boldsymbol{x}_j$$

$$= (1-\lambda)\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), y_i\Big) + \lambda\ell\Big(f\big((1-\lambda)\boldsymbol{x}_i\big), \bar{y}_j\Big) + \lambda\frac{\partial \ell}{\partial \boldsymbol{x}}\cdot \bar{\boldsymbol{x}}_j$$

Algorithm 2: FedMix

LocalUpdate$(k, \boldsymbol{w}_t; \boldsymbol{X}_g, \boldsymbol{Y}_g)$ under MAFL (Algorithm 1):

$\boldsymbol{w} \leftarrow \boldsymbol{w}_t$

for $e = 0, \ldots, E-1$ **do**

  Split $\mathbb{D}_k$ into batches of size B

  for *batch*$(\boldsymbol{X}, \boldsymbol{Y})$ **do**

    Select an entry $\boldsymbol{x}_g, \boldsymbol{y}_g$ from $\boldsymbol{X}_g, \boldsymbol{Y}_g$

    $\ell_1 = (1-\lambda)\ell\big(f((1-\lambda)\boldsymbol{X}; \boldsymbol{w}), \boldsymbol{Y}\big)$

    $\ell_2 = \lambda\ell\big(f((1-\lambda)\boldsymbol{X}; \boldsymbol{w}), \boldsymbol{y}_g\big)$

    $\ell_3 = \lambda\frac{\partial \ell_1}{\partial \boldsymbol{x}} \cdot \boldsymbol{x}_g$

    (derivative calculated at $\boldsymbol{x} = (1-\lambda)\boldsymbol{x}_i$ and $y = y_i$ for each of $\boldsymbol{x}_i, y_i$ in $\boldsymbol{X}, \boldsymbol{Y}$)

    $\ell = \ell_1 + \ell_2 + \ell_3$

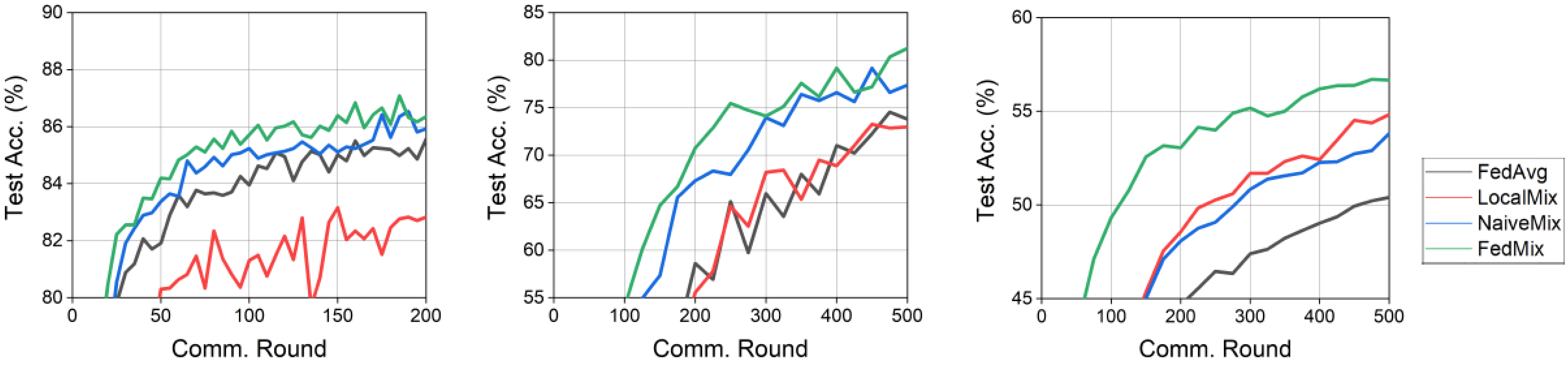
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta_{t+1}\nabla\ell$

  end

end

return $\boldsymbol{w}$

Experiment



(a) FEMNIST (b) CIFAR10 (c) CIFAR100

Figure 2: Learning curves for various algorithms on benchmark datasets. Learning curves correspond to results in Table 1. (For simplicity, we only show key algorithms to compare.)

Experiment

Table 1: **Test accuracy after (target rounds)** and **number of rounds to reach (target test accuracy)** on various datasets. Algorithms in conjunction with FedProx are compared separately (bottom). MAFL-based algorithms are marked in bold.

Algorithm	FEMNIST		CIFAR10		CIFAR100	
	test acc. (200)	rounds (80%)	test acc. (500)	rounds (70%)	test acc. (500)	rounds (40%)
Global Mixup	88.2	8	88.2	85	61.4	54
FedAvg	85.3	26	73.8	283	50.4	101
LocalMix	82.8	28	73.0	267	54.8	91
NaiveMix	85.9	23	77.4	198	53.8	85
FedMix	**86.5**	**18**	**81.2**	**162**	**56.7**	**34**
FedProx	84.6	**29**	77.3	266	51.2	79
FedProx + LocalMix	84.1	39	74.1	314	54.0	90
FedProx + **NaiveMix**	85.7	37	76.7	230	53.1	74
FedProx + **FedMix**	**86.0**	32	**78.9**	**223**	**54.5**	**63**

Experiment

Table 2: Test accuracy after 50 rounds on Shakespeare dataset.

Algorithm	Global Mixup	FedAvg	FedProx	LocalMix	NaiveMix	FedMix
Test Acc. (%)	54.4	54.7	54.4	53.7	**56.9**	**56.9**

Experiment

Table 3: Test accuracy on CIFAR10, under varying number of clients (N). Number of samples per client is kept constant.

# of Clients(N)	**20**	**40**	**60**
Global Mixup	86.3	89.2	88.2
FedAvg	65.8	73.4	73.8
LocalMix	46.9	71.4	73.0
NaiveMix	62.2	75.1	77.4
FedMix	**68.5**	**76.4**	**81.2**

Experiment

Table 5: Test accuracy on CIFAR10, under varying mixup ratio λ.

λ	**0.05**	**0.1**	**0.2**	**0.5**
Global Mixup	79.4	80.4	81.1	63.6
FedMix	81.2	80.5	77.7	67.1

	M_k	**5**	**10**	**20**	**50**	**All**
FEMNIST	NaiveMix	85.7	**86.3**	86.2	86.1	85.9
	FedMix	**86.0**	85.7	**86.4**	**86.2**	**86.5**
CIFAR10	NaiveMix	79.6	77.9	79.1	77.1	77.4
	FedMix	**81.4**	**79.9**	**80.4**	**79.5**	**81.2**

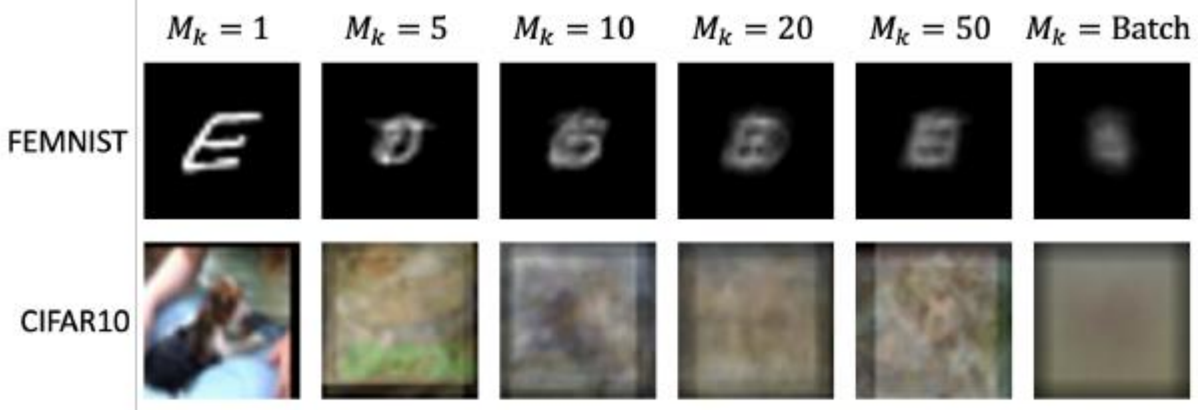


Figure 3: Performance of MAFL-based algorithms for various M_k values (left), and samples of averaged images from EMNIST/CIFAR10 for various M_k values (right).

Table 6: Test accuracy after 500 rounds on CIFAR10, under varying number of classes per client.

Algorithm	**class/client**			
	2	**3**	**5**	**10 (iid)**
Global Mixup	88.2	90.7	90.9	91.4
FedAvg	73.8	84.2	86.8	89.3
Localmix	73	83.3	86.4	89.1
NaiveMix	77.4	84.5	87.7	**89.4**
FedMix	**81.2**	**85.1**	**87.9**	89.1

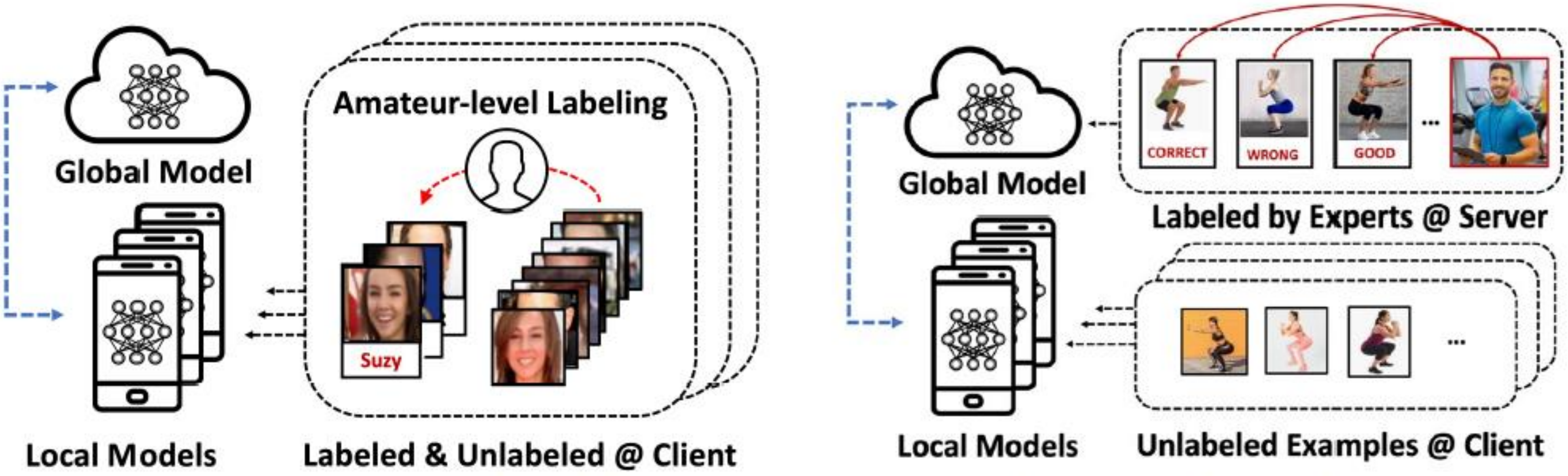
南京航空航天大學
Nanjing University of Aeronautics and Astronautics

FEDERATED SEMI-SUPERVISED LEARNING WITH INTER-CLIENT CONSISTENCY & DISJOINT LEARNING

Wonyong Jeong[1], Jaehong Yoon[2], Eunho Yang[1,3], and Sung Ju Hwang[1,3]
Graduate School of AI[1], KAIST, Seoul, South Korea
School of Computing[2], KAIST, Daejeon, South Korea
AITRICS [3], Seoul, South Korea
{wyjeong, jaehong.yoon, eunhoy, sjhwang82}@kaist.ac.kr

ICLR 2021

Method



(a) Labels-at-Client Scenario (b) Labels-at-Server Scenario

Method

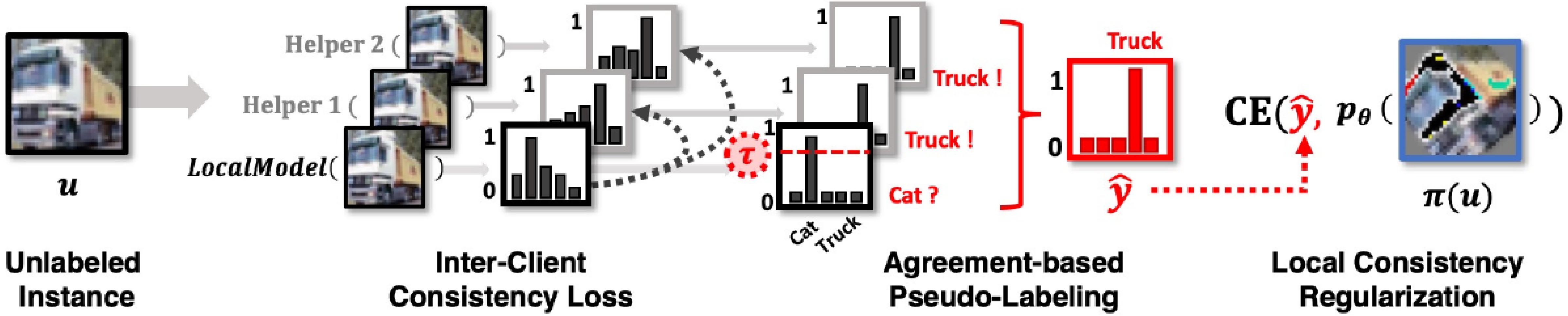


Figure 2: **Illustration of Inter-Client Consistency Loss.** We illustrate each step of our inter-client consistency regularization process performed at local client. We provide the detailed explanations in Section 3.1.

$$\Phi(\cdot) = \text{CrossEntropy}(\hat{\mathbf{y}}, p_{\boldsymbol{\theta}^l}(\mathbf{y}|\pi(\mathbf{u}))) + \frac{1}{H}\sum_{j=1}^{H} \text{KL}[p^*_{\boldsymbol{\theta}^{h_j}}(\mathbf{y}|\mathbf{u})||p_{\boldsymbol{\theta}^l}(\mathbf{y}|\mathbf{u})]$$

$$\hat{\mathbf{y}} = \text{Max}(\mathbb{1}(p^*_{\boldsymbol{\theta}^l}(\mathbf{y}|\mathbf{u})) + \sum_{j=1}^{H} \mathbb{1}(p^*_{\boldsymbol{\theta}^{h_j}}(\mathbf{y}|\mathbf{u})))$$

Method

Decomposing model parameters θ into two variables, σ for supervised learning and φ for unsupervised learning, such that $\theta = \sigma + \varphi$.

Stage 1: Performing standard supervised learning on σ, while keeping φ fixed.

$$\text{minimize } \mathcal{L}_s(\sigma) = \lambda_s \text{CrossEntropy}(\mathbf{y}, p_{\sigma+\psi^*}(\mathbf{y}|\mathbf{x}))$$

Stage 2: Performing unsupervised learning conversely on φ, while keeping σ fixed.

$$\text{minimize } \mathcal{L}_u(\psi) = \lambda_{\text{ICCS}} \Phi_{\sigma^*+\psi}(\cdot) + \lambda_{L_2} ||\sigma^* - \psi||_2^2 + \lambda_{L_1} ||\psi||_1$$

Method

Algorithm 1 Labels-at-Client Scenario

RunServer()
initialize σ^0 and ψ^0
for each round $r = 1, 2, ..., R$ **do**
  $\mathcal{L}^r \leftarrow$ (select random A clients from $\mathcal{L}$)
  for each client $l_a^r \in \mathcal{L}^r$ **in parallel do**
    $\psi_{1:H}^r \leftarrow \text{GetNearestNeighbors}(\psi^r)$
    $\sigma_a^r, \psi_a^r \leftarrow \text{RunClient}(\sigma^r, \psi^r, \psi_{1:H}^r)$
    $\text{EmbedLocalModel}(\sigma_a^r, \psi_a^r)$
  end for
  $\sigma^{r+1} \leftarrow \frac{1}{A}\sum_{a=1}^{A}(\sigma_{l_a}^r)$
  $\psi^{r+1} \leftarrow \frac{1}{A}\sum_{a=1}^{A}(\psi_{l_a}^r)$
end for
RunClient$(\sigma, \psi, \psi_{1:H})$
$\theta_{l_a} \leftarrow \sigma + \psi$, $\theta_{h_{1:H}} \leftarrow \sigma + \psi_{1:H}$
for each local epoch e from 1 to E_L **do**
  for minibatch $s \in \mathcal{S}_{l_a}$ and $u \in \mathcal{U}_{l_a}$ **do**
    $\theta_{\sigma+\psi^*} \leftarrow \theta_{\sigma+\psi^*} - \eta \nabla \ell_s(\theta_{\sigma+\psi^*}; \theta_{h_{1:H}}, s)$
    $\theta_{\sigma^*+\psi} \leftarrow \theta_{\sigma^*+\psi} - \eta \nabla \ell_u(\theta_{\sigma^*+\psi}; \theta_{h_{1:H}}, u)$
  end for
end for

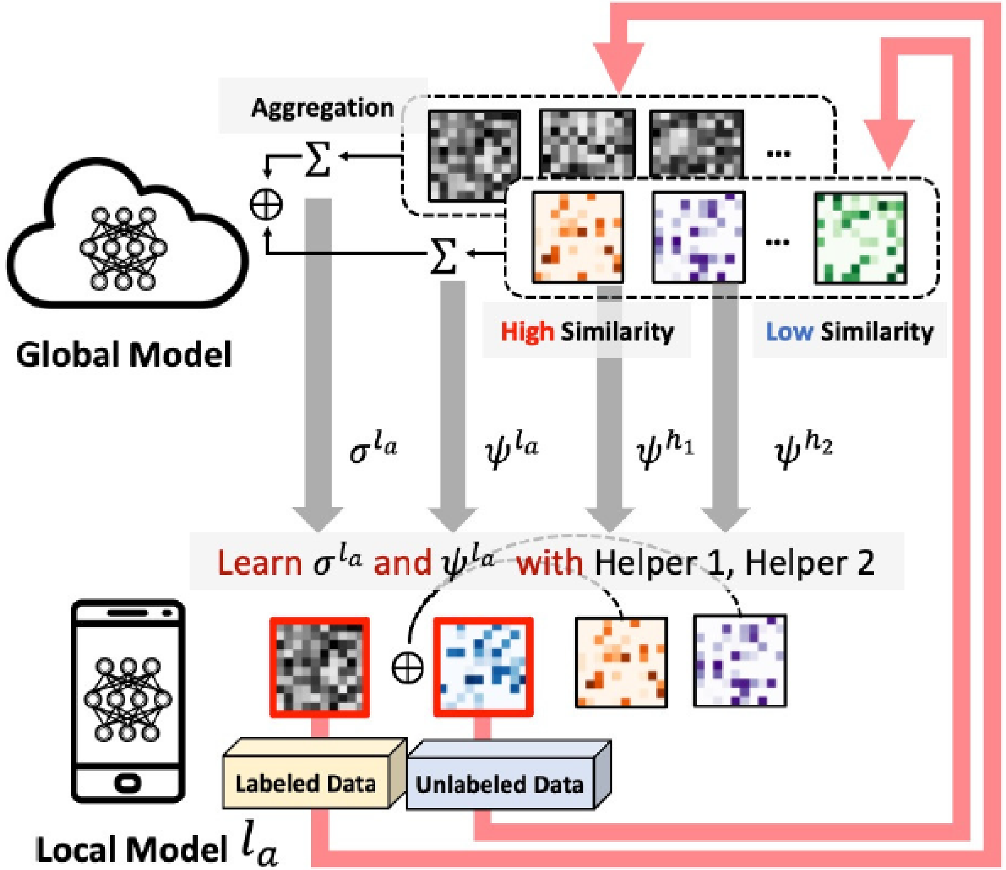


Figure 3: **Illustrative Running Example of Labels-at-Client Scenario** We describe training and communication procedure between local and global model under Labels-at-Client scenario corresponding to the Algorithm 1. More details are described in Section 4.

Method

Algorithm 2 Labels-at-Server Scenario

RunServer()
initialize σ^0, ψ^0
for each round $r = 1, 2, ..., R$ **do**
  for each server epoch e from 1 to E_G **do**
    for minibatch $s \in \mathcal{S}_G$ **do**
      $\theta_{\sigma+\psi^*} \leftarrow \theta_{\sigma+\psi^*} - \eta \nabla \ell_s(\theta_{\sigma+\psi^*}; s)$
    end for
  end for
  $\mathcal{L}^r \leftarrow$ (select random A clients from $\mathcal{L}$)
  for each client $l_a^r \in \mathcal{L}^r$ **in parallel do**
    $\psi_{1:H}^r \leftarrow$ GetNearestNeighbors(ψ^r)
    $\psi_a^r \leftarrow$ RunClient$(\sigma^{r+1}, \psi^r, \psi_{1:H}^r)$
    EmbedLocalModel(σ^{r+1}, ψ_a^r)
  end for
  $\psi^{r+1} \leftarrow \frac{1}{A} \sum_{a=1}^{A} (\psi_{l_a}^r)$
end for
RunClient$(\sigma, \psi, \psi_{1:H})$
$\theta_l \leftarrow \sigma^* + \psi$, $\theta_{h_{1:H}} \leftarrow \sigma^* + \psi_{1:H}$
for each local epoch e from 1 to E_L **do**
  for minibatch $u \in \mathcal{U}_{l_a}$ **do**
    $\theta_{\sigma^*+\psi} \leftarrow \theta_{\sigma^*+\psi} - \eta \nabla \ell_u(\theta_{\sigma^*+\psi}; \theta_{h_{1:H}}, u)$
  end for
end for

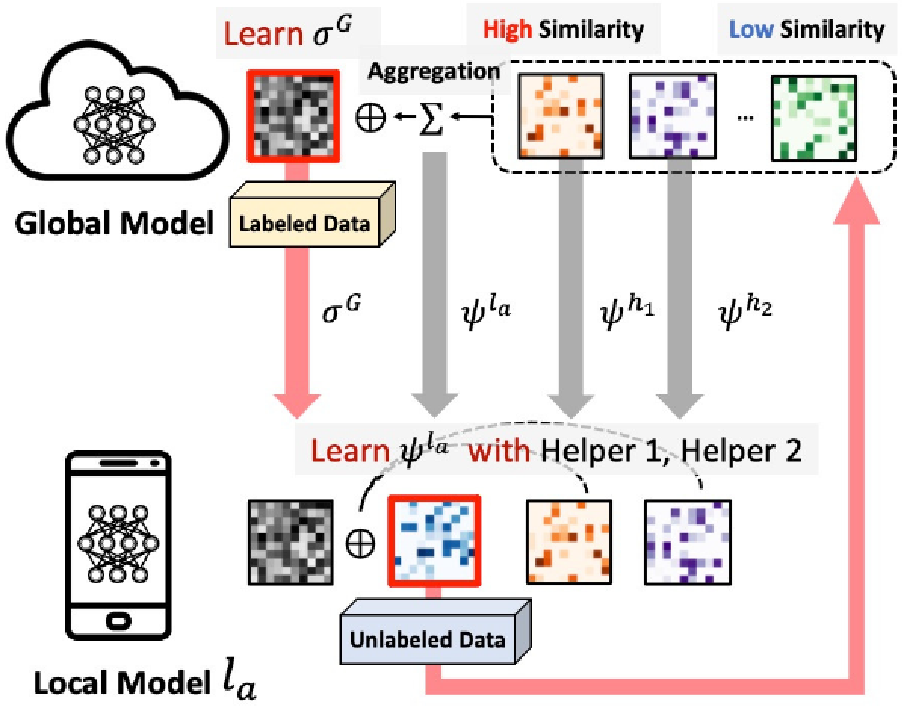


Figure 4: **Illustrative Running Example of Labels-at-Server Scenario** We depict learning and transmitting procedure between a client and the global server under Labels-at-Server scenario corresponding to the Algorithm 2. Note that, in labels-at-server scenario, the labeled data is only available at the server, and thus global model at the server learns on labeled data, while local models at clients learn on only unlabeled data. Further details are explained in Section 5.

Experiment

Table 1: **Performance Comparison on Batch-IID & NonIID Tasks** We use 100 clients (F=0.05) for 200 rounds. We measure global model accuracy and averaged communication costs. Note that the SL (Supervised Learning) models learn on both $\mathcal{S}$ and $\mathcal{U}$ with full labels, and are utilized as the upper bounds for each experiment.

CIFAR-10, Batch-IID Task with 100 Clients (K=100, F=0.05, H=2)						
	Labels-at-Client Scenario			*Labels-at-Server* Scenario		
Methods	**Acc.(%)**	**S2C Cost**	**C2S Cost**	**Acc.(%)**	**S2C Cost**	**C2S Cost**
FedAvg-SL	58.60 $\pm$ 0.42	100 %	100 %	52.45 $\pm$ 0.23	100 %	100 %
FedProx-SL	59.30 $\pm$ 0.31	100 %	100 %	49.11 $\pm$ 0.38	100 %	100 %
FedAvg-UDA	46.35 $\pm$ 0.29	100 %	100 %	24.81 $\pm$ 0.73	100 %	100 %
FedProx-UDA	47.45 $\pm$ 0.21	100 %	100 %	19.91 $\pm$ 0.31	100 %	100 %
FedAvg-FixMatch	47.01 $\pm$ 0.43	100 %	100 %	11.95 $\pm$ 0.60	100 %	100 %
FedProx-FixMatch	47.20 $\pm$ 0.12	100 %	100 %	25.61 $\pm$ 0.32	100 %	100 %
FedMatch (Ours)	**52.13** $\pm$ 0.34	**79 %**	**46 %**	**44.95** $\pm$ 0.49	**45 %**	**22 %**
CIFAR-10, Batch-NonIID Task with 100 Clients (K=100, F=0.05, H=2)						
FedAvg-SL	55.15 $\pm$ 0.21	100 %	100 %	51.50 $\pm$ 0.51	100 %	100 %
FedProx-SL	57.75 $\pm$ 0.15	100 %	100 %	49.31 $\pm$ 0.18	100 %	100 %
FedAvg-UDA	44.35 $\pm$ 0.39	100 %	100 %	27.61 $\pm$ 0.71	100 %	100 %
FedProx-UDA	46.31 $\pm$ 0.63	100 %	100 %	26.01 $\pm$ 0.78	100 %	100 %
FedAvg-FixMatch	46.20 $\pm$ 0.52	100 %	100 %	09.45 $\pm$ 0.34	100 %	100 %
FedProx-FixMatch	45.55 $\pm$ 0.63	100 %	100 %	09.21 $\pm$ 0.24	100 %	100 %
FedMatch (Ours)	**52.25** $\pm$ 0.81	**85 %**	**49 %**	**44.17** $\pm$ 0.19	**42 %**	**20 %**

THANKS