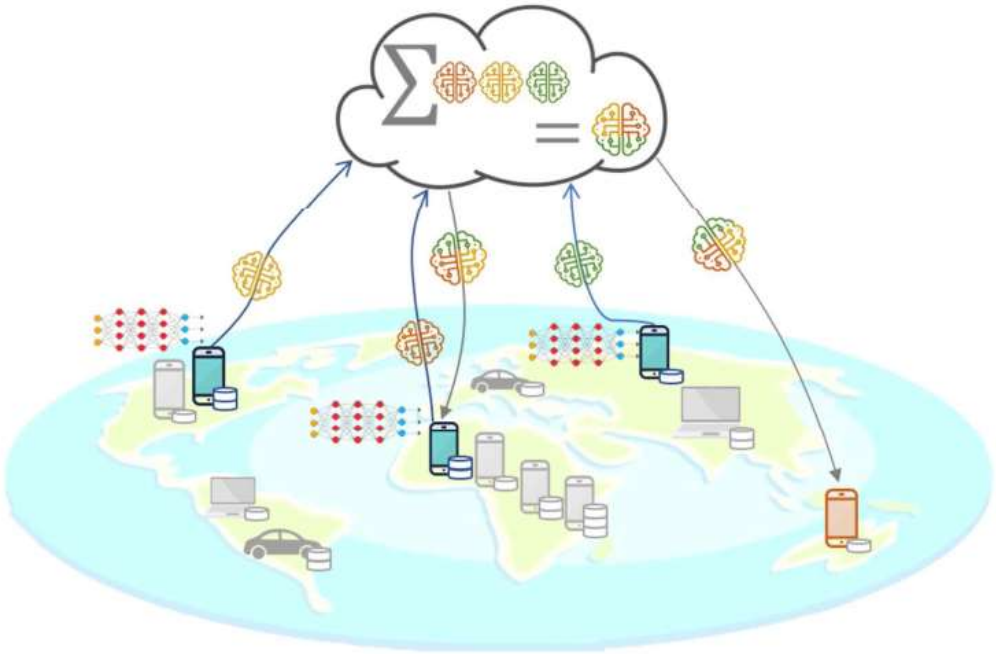
FEDERATED SEMI-SUPERVISED LEARNING WITH INTER-CLIENT CONSISTENCY & DISJOINT LEARNING

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

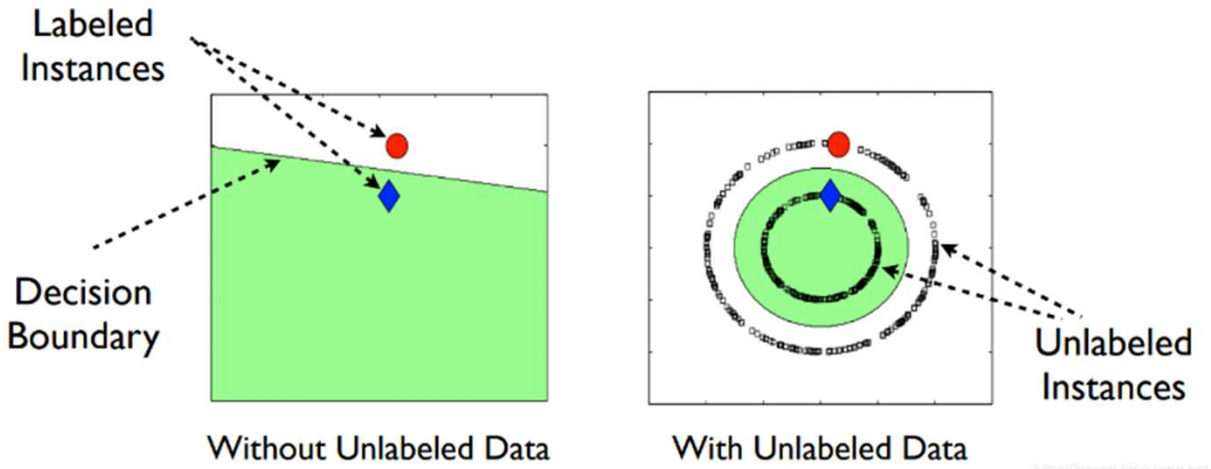
Federated Learning (FL)

FL can improve model's performance while protecting users' privacy.



Semi-supervised learning (SSL)

Semi-supervised learning can learn knowledge from unlabeled data without high labeling cost.



Related work

Federated Learning

FedAvg
(McMahan et al.,2016)

classic FL algorithm requires many communication rounds to train an effective global model.

FedProx
(Li et al.,2020)

adjusts the local training procedure to pull back local models from global model.

Related work

Semi-supervised learning

UDA
(Xie et al., 2019)

use two sets of augmentations, weak and strong, and enforce consistency between the weakly and strongly augmented examples.

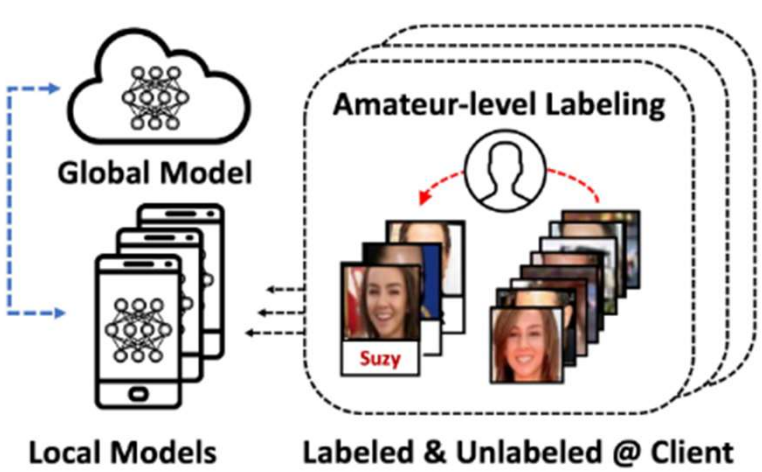
FixMatch
(Sohn et al., 2020)

in addition to enforcing consistency between weak-strong augmented pairs, performs pseudo-label refinement on model predictions via thresholding.

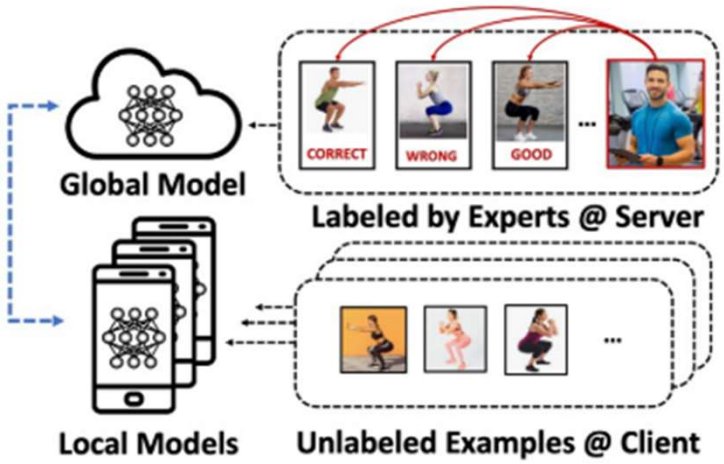
Motivation

Defect in FL

A common limitation is that they only consider supervised learning settings, where the local private data is fully labeled. Yet, the assumption that all of the data examples may include sophisticate annotations is not realistic for real-world applications.



(a) Labels-at-Client Scenario



(b) Labels-at-Server Scenario

Defects in FL + SSL simply

This leads them to practical problems of federated learning with deficiency of labels, namely Federated Semi-Supervised Learning (FSSL).

A naive solution to these scenarios is to simply perform Semi-Supervised Learning (SSL) using any off-the-shelf methods (e.g. FixMatch (Sohn et al., 2020), UDA (Xie et al., 2019)).

This solution has defects:

- does not fully exploit the knowledge of the multiple models trained on heterogeneous data distributions;
- conventional semi-supervised learning approaches are not applicable for scenarios where labeled data is only available at the server;
- even when the labeled data is available at the client, learning from the unlabeled data may lead to forgetting of what the model learned from the labeled data.

INTER-CLIENT CONSISTENCY LOSS

Conventional consistency-regularization methods enforce the predictions from the augmented examples and original (or weakly augmented) instances to output the same class label. $\pi(\mathbf{u})$ performs RandAugment.

$$||p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{u}) - p_{\boldsymbol{\theta}}(\mathbf{y}|\pi(\mathbf{u}))||_2^2$$

They additionally propose a novel consistency loss called inter-client consistency that regularizes the models learned at multiple clients to output the same prediction, defined as follows:

$$\frac{1}{H}\sum_{j=1}^{H}\mathrm{KL}[p^*_{\boldsymbol{\theta}^{\mathrm{h}_j}}(\mathbf{y}|\mathbf{u})||p_{\boldsymbol{\theta}^{l}}(\mathbf{y}|\mathbf{u})]$$

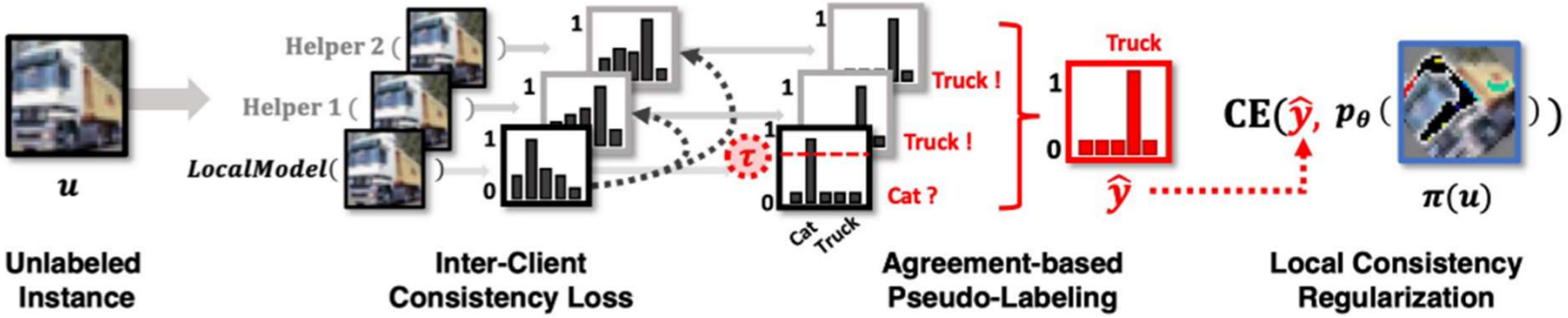
INTER-CLIENT CONSISTENCY LOSS

They also use data-level consistency regularization at each local client similarly to FixMatch. Final consistency regularization term $\Phi(\cdot)$ can be written as follows:

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

$\hat{\mathbf{y}}$ is the agreement-based pseudo label, defined as follows:

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



INTER-CLIENT CONSISTENCY LOSS

They select the H helper agents $p^*_{\boldsymbol{\theta}^{\mathrm{h}_{j:H}}}(\mathbf{y}|\mathbf{u})$ for each client as the most relevant models from other clients:

- represent each model by its prediction **m** on the same arbitrary input **a** located at server (random Gaussian noise). $\mathbf{m}^l = p_{\boldsymbol{\theta}^l}(\mathbf{m}|\mathbf{a})$
- server tries to keep and update all model embeddings $\mathbf{m}^{1:K}$ from clients once each client updates its weights to server.
- server creates K-Dimensional Tree (KD Tree) on m in the current round r for nearest neighbor search to rapidly select the H helper agents for each client in the next rounds.
- server send helper agents for every 10 rounds, and if a certain client has not yet updated its weights to server in the previous step, then server simply skips sending helpers to the client at the round.

PARAMETER DECOMPOSITION FOR DISJOINT LEARNING

Second, they decompose model parameters θ into two variables, σ for supervised learning and ψ for unsupervised learning, such that θ = σ + ψ.

perform standard supervised learning on σ, while keeping ψ fixed during training on labeled data, by minimizing the loss term as follows:

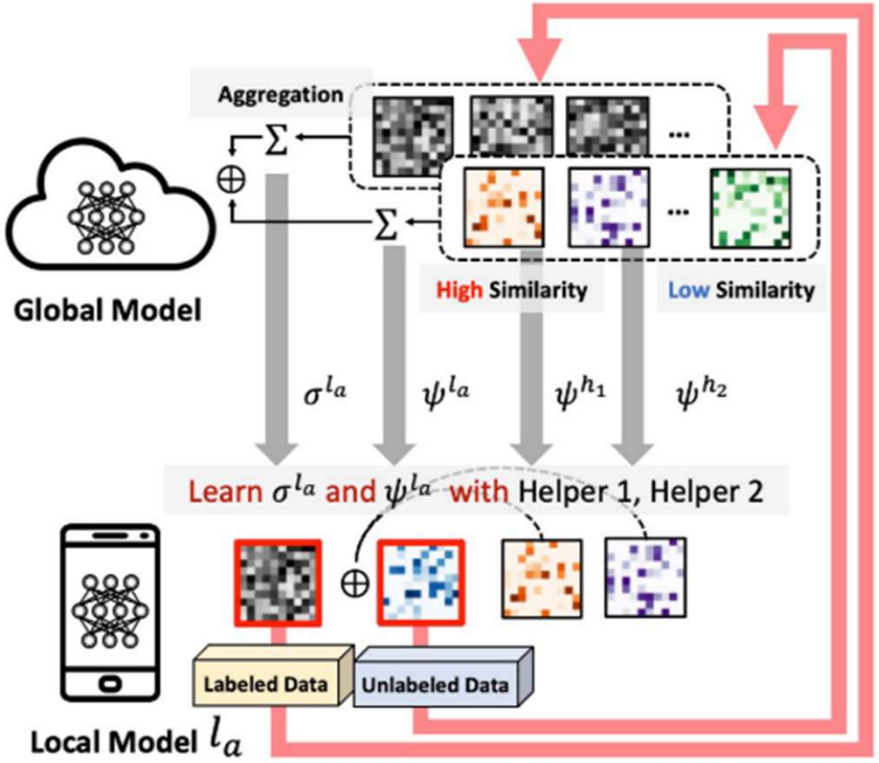
$$\text{minimize } \mathcal{L}_s(\sigma) = \lambda_s \text{CrossEntropy}(\mathbf{y}, p_{\sigma+\psi^*}(\mathbf{y}|\mathbf{x}))$$

perform unsupervised learning conversely on ψ, while keeping σ fixed for the learning phase on unlabeled data, by minimizing the consistency loss terms as follows:

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

Method

The Overall Algorithm

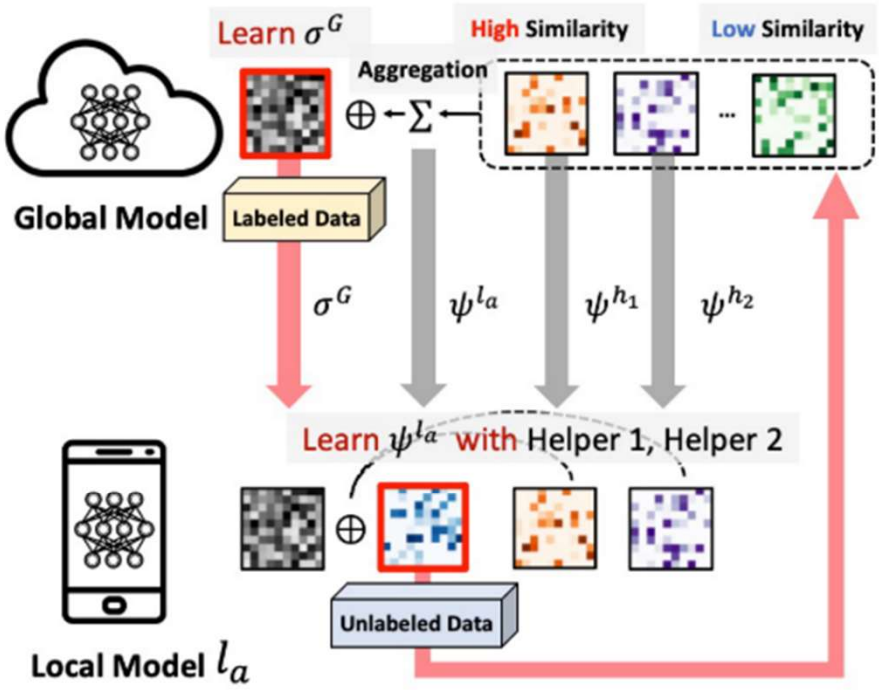


Algorithm 1 Labels-at-Client Scenario

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

Method

The Overall Algorithm



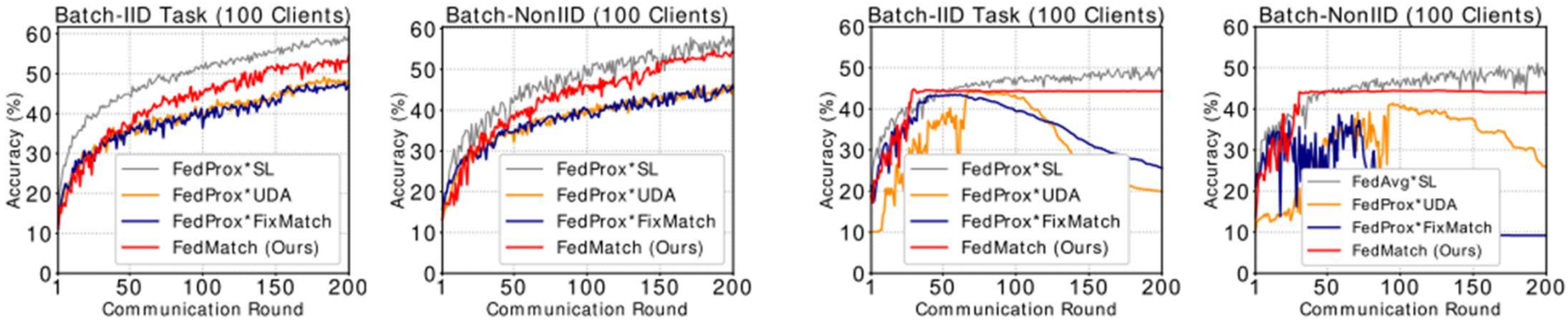
Algorithm 2 Labels-at-Server Scenario

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**
    $\psi_{1:H}^r \leftarrow \text{GetNearestNeighbors}(\psi^r)$
    $\psi_a^r \leftarrow \text{RunClient}(\sigma^{r+1}, \psi^r, \psi_{1:H}^r)$
    $\text{EmbedLocalModel}(\sigma^{r+1}, \psi_a^r)$
  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

Performance Comparison on Batch-IID & NonIID Tasks

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

Performance Comparison on Batch-IID & NonIID Tasks



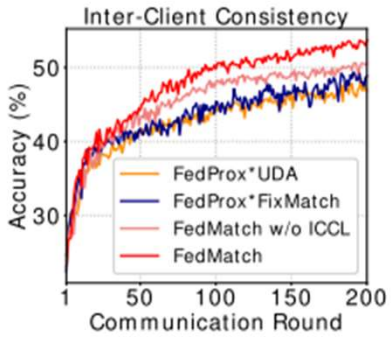
(a) Labels-at-Client Scenario

(b) Labels-at-Server Scenario

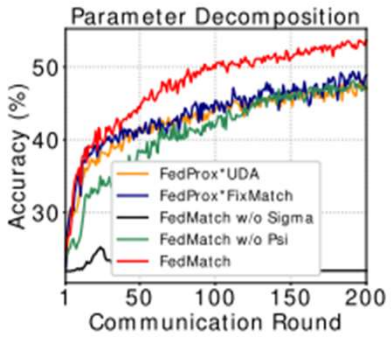
Averaged Local Performance on Streaming-NonIID Task

Fashion-MNIST, Streaming-NonIID Task with 10 Clients (K=10, F=1.0, H=2)						
	Labels-at-Cleint Scenario			*Labels-at-Server* Scenario		
Methods	**Acc.(%)**	**S2C Cost**	**C2S Cost**	**Acc.(%)**	**S2C Cost**	**C2S Cost**
Local-SL	87.19 $\pm$ 0.36	N/A	N/A	N/A	N/A	N/A
Local-UDA	70.70 $\pm$ 0.28	N/A	N/A	N/A	N/A	N/A
Local-FixMatch	62.62 $\pm$ 0.32	N/A	N/A	N/A	N/A	N/A
FedProx-SL	82.06 $\pm$ 0.26	100 %	100 %	77.43 $\pm$ 0.42	100 %	100 %
FedProx-UDA	73.71 $\pm$ 0.17	100 %	100 %	83.34 $\pm$ 0.21	100 %	100 %
FedProx-FixMatch	62.40 $\pm$ 0.43	100 %	100 %	73.71 $\pm$ 0.32	100 %	100 %
FedMatch (Ours)	**77.95** $\pm$ **0.14**	**37 %**	**48 %**	**84.15** $\pm$ **0.31**	**14 %**	**63 %**

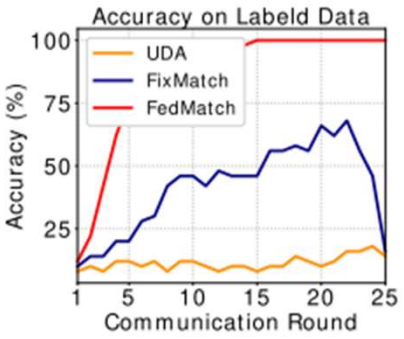
Ablation Study



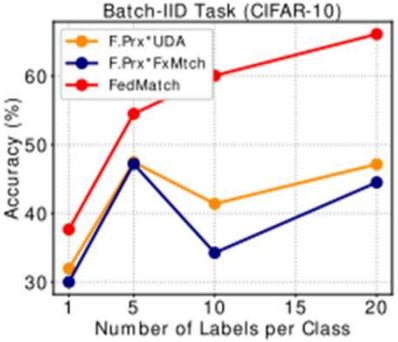
(a) Inter-Client Consistency



(b) Sigma & Psi



(c) Inter-Task Interference



(d) Number of Labels

Thanks