南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Towards Unbiased Training in Federated Open-world Semi-supervised Learning

Jie Zhang [1] Xiaosong Ma [1] Song Guo [1] Wenchao Xu [1]

ICML 2023

Introduction

1) **FedSSL:simultaneously exploit both the labeled and unlabeled data to optimize a global model in distributed environments**

Existing FedSSL schemes rely on the **closed-world assumption** that all local training data and global testing data are from the **same set of classes** that are **included in the labeled dataset**,which is often invalid for practical scenarios.

2)In contrast, the **open-world settings** allow **novel class discovery**.

New question arises: **how to collaboratively train models on distributed data to enable classification on both seen and unseen classes under the open-world setting?**

Introduction

More problems:

a) significant **performance degradation** due to the **existence of unseen classes** in unlabeled data during the training.

b) With multiple participants,some unseen classes **in one client may exist in other clients' side from a global view**, and thus requires a **novel fine-grained definition** on unseen classes as well as the **training mechanism for different types of samples/classes.**

c) Due to the heterogeneous distributed classes across different clients,simply aggregating the parameters following traditional FL mechanism can **cause the biased training process** for clients **possessing different unseen classes**.

Solution:FedoSSL——achieve unbiased training procedure among different types of samples

Design:redefined unseen classes / uncertainty-aware suppressed loss / calibration module

Introduction

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- Main contributions

- To the best of our knowledge, we are the first to consider the open-world setting in FedSSL, where unseen classes exist in the unlabeled data, which is challenging due to the heterogeneously distributed unseen classes.
- We design a brand-new FedoSSL framework, that can achieve unbiased learning among different types of classes (i.e., locally unseen and globally unseen classes) and calibrated knowledge aggregation given heterogeneous data distributions.
- We conduct extensive experiments on three typical image classification tasks. The empirical evaluation shows the superior performance of FedoSSL over the state-of-the-art approaches.

Related Works

Federated Open-world Semi-supervised Learning

Table 1: Comparison between our proposed FedoSSL and other SSL-related methods.

Method	Training Dataset		Testing Dataset		FL Environment?
	Seen classes	Unseen classes	Seen classes	Unseen classes	
SSL	Present	Not Present	Classify	-	×
Open-set SSL	Present	Present	Classify	Detect & Reject	×
Novel Class Discovery	Present	Present	-	Discover & Cluster	×
Open-world SSL	Present	Present	Classify	Discover & Cluster	×
FedSSL	Present	Not Present	Classify	-	√
FedoSSL	Present	Present	Classify	Discover & Cluster	√

Open-set SSL: considers that unseen classes in unlabeled samples only exist in training data, while not exist in testing data.

Novel Class Discovery(NCD): aims to classify both seen and unseen classes during the testing phase but assumes all unlabeled instances belonging to unseen classes in training data.

Related Works

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Federated Open-world Semi-supervised Learning

Table 1: Comparison between our proposed FedoSSL and other SSL-related methods.

Method	Training Dataset		Testing Dataset		FL Environment?
	Seen classes	Unseen classes	Seen classes	Unseen classes	
SSL	Present	Not Present	Classify	-	×
Open-set SSL	Present	Present	Classify	Detect & Reject	×
Novel Class Discovery	Present	Present	-	Discover & Cluster	×
Open-world SSL	Present	Present	Classify	Discover & Cluster	×
FedSSL	Present	Not Present	Classify	-	√
FedoSSL	Present	Present	Classify	Discover & Cluster	√

Open-world SSL: each test sample should be either classified into one of existing classes or a new unseen class in the test time

Existing FedSSL: 1) Labels-at Server 2) Labels-at-Client

a) each client contains both labeled and unlabeled data

b) some clients are fully labeled while some clients only contain unlabeled samples

Method

- **Problem Definition**

label set: $\mathcal{D}^l = \{\mathcal{D}_i^l\}_{i=1}^K$

unlabel set: $\mathcal{D}^u = \{\mathcal{D}_i^u\}_{i=1}^K$

traditional (close world) FedSSL: $\mathcal{C}^l = \mathcal{C}^u$

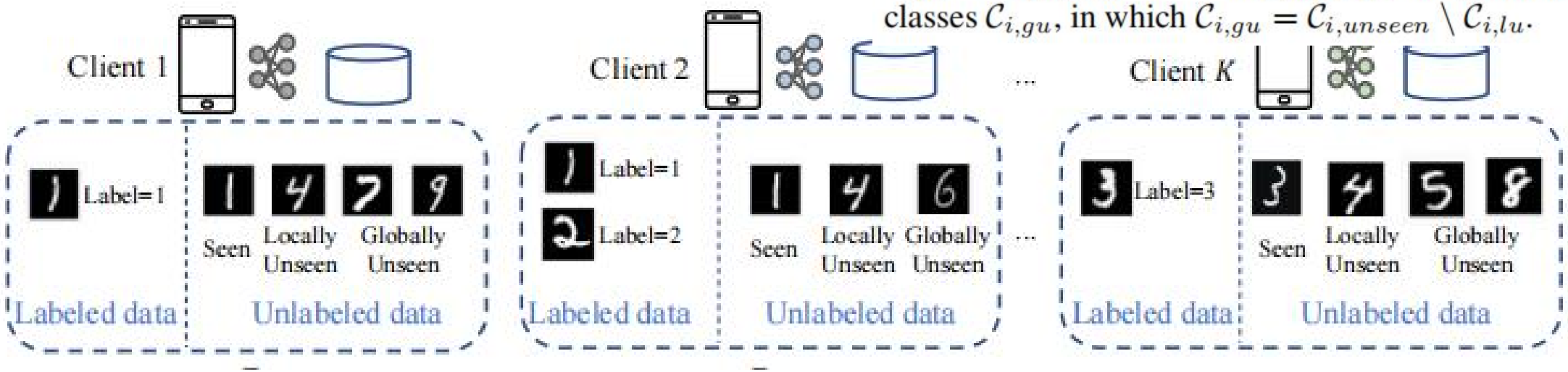
FedoSSL: $\mathcal{C}^l \neq \mathcal{C}^u$

$$\mathcal{C}_{seen} = \mathcal{C}^l \cap \mathcal{C}^u$$

$$\mathcal{C}_{unseen} = \mathcal{C}^u \setminus \mathcal{C}_{seen}$$

- **Inconsistent data distribution on different clients raises another new problem:some unseen classes may exist in more than one client, resulting in biased training among different unseen classes**
- **More fine-grained definition on unseen classes:**

Definition 1 (locally unseen & globally unseen class). In FedoSSL, the unseen classes $\mathcal{C}_{i,unseen}$ on client i can be divided into two types: locally unseen classes $\mathcal{C}_{i,lu}$, in which $\mathcal{C}_{i,lu} = \mathcal{C}_{1,unseen} \cap \cdots \cap \mathcal{C}_{K,unseen}$; and globally unseen classes $\mathcal{C}_{i,gu}$, in which $\mathcal{C}_{i,gu} = \mathcal{C}_{i,unseen} \setminus \mathcal{C}_{i,lu}$.



Method

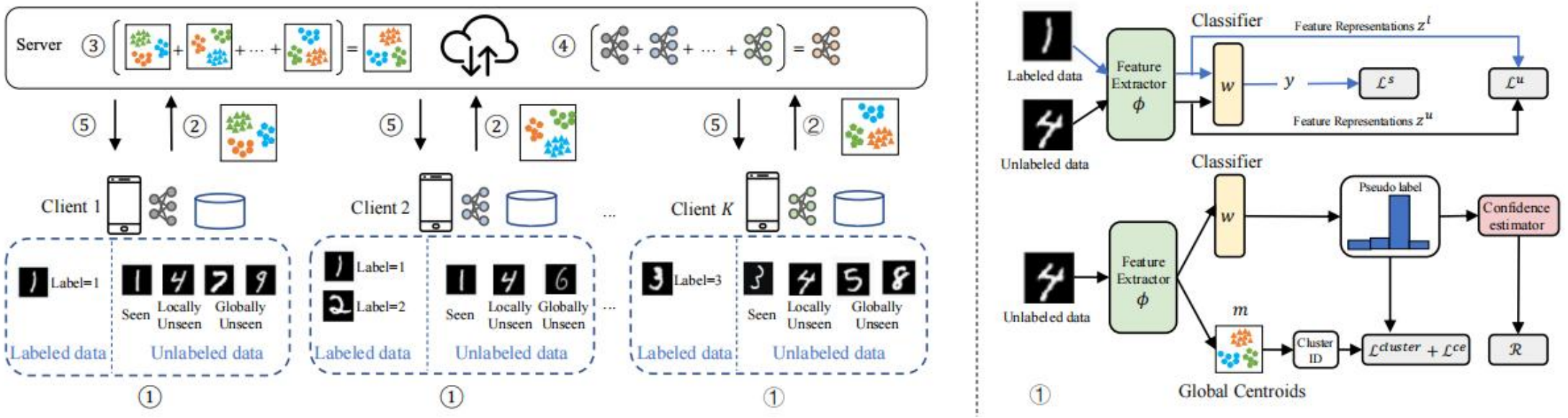
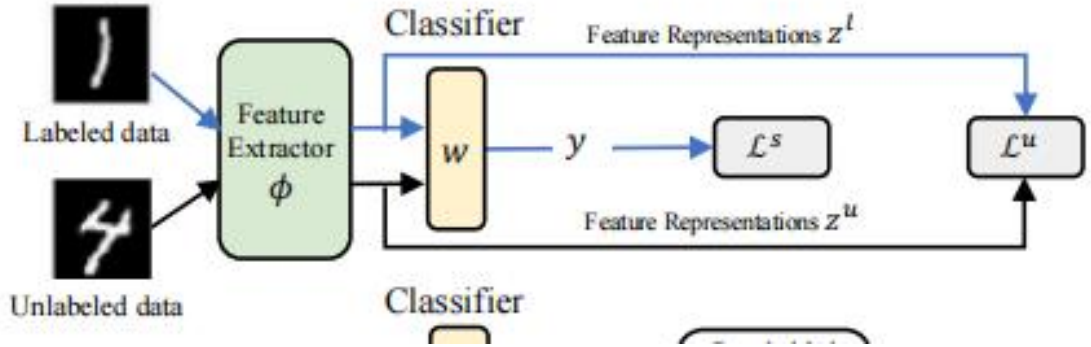


Figure 1: Framework of the proposed FedoSSL algorithm. **Pipeline**: ① **Local Training:** Each client first performs local training on its private dataset for several epochs (i.e., via optimizing loss function in Eq. (1)), and then computes local centroids via a Sinkhorn-Knopp based clustering algorithm (Genevay et al., 2019). ② Upload model parameters and local centroids to the server. ③ The server performs standard model aggregation. ④ The server performs centroids aggregation by again using Sinkhorn-Knopp clustering to obtain global centroids. ⑤ The global model and global centroids are returned to the clients, who use them for local training.

Method



$$\mathcal{L}_i = \mathcal{L}_i^s + \alpha \mathcal{L}_i^u$$

$$\mathcal{L}_i^s = \frac{1}{n_i^l} \sum_{(x_j, y_j) \in \mathcal{D}_i^l} \mathcal{H}(y_j, p(x_j; \theta))$$

Goal of FedSSL: train a generalized global model f with parameter θ from multiple decentralized clients.

$$\min_\theta \mathcal{L}(\theta) := \sum_{i=1}^{K} \frac{n_i}{n} \mathcal{L}_i(\theta)$$

$$n_i = n_i^l + n_i^u$$

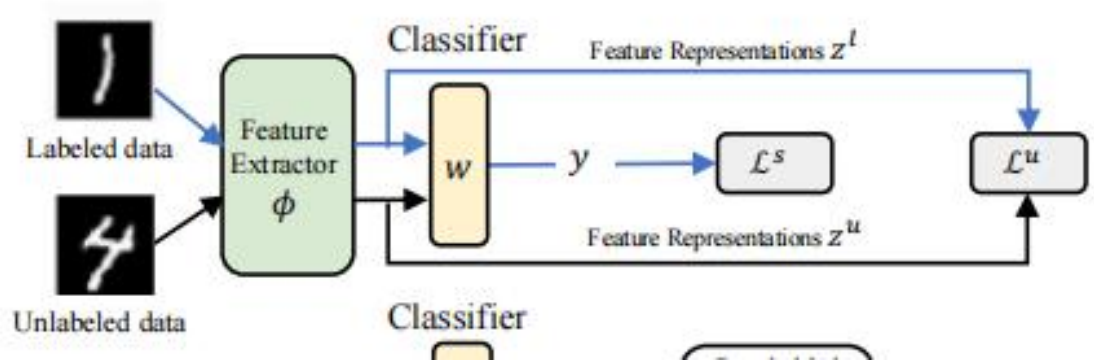
Two typical forms of unsupervisied loss $\mathcal{L}_i^u$:
1) pseudo-labels
2) consistency regularization

fail to classify seen classes and unseen classes

model f
- feature extractor g with parameter ϕ $\mathbb{R}^N \rightarrow \mathbb{R}^d$ to learn a low-dimensional feature z
- a classifier h with parameter w $\mathbb{R}^d \rightarrow \mathbb{R}^{|\mathcal{C}_{seen} \cup \mathcal{C}_{unseen}|}$

The dimension of the classifier corresponds to the number of classes in each dataset

Method



Based on **ORCA and NACH**, use **pairwise objective** as unsupervised loss on unlabeled data:

$$\mathcal{L}_i^u = -\frac{1}{n_i^l + n_i^u} \sum_{\substack{z_j, \bar{z}_j \in \\ (Z_i^l \cup Z_i^u, \bar{Z}_i^l \cup \bar{Z}_i^u)}} \mathcal{H}(p(w^\top \cdot z_j), p(w^\top \cdot \bar{z}_j))$$

two main challenges:

a) locally unseen classes may be learned faster than globally unseen classes

existing unsupervised pairwise loss treats each class equally → a big bias on pseudo-label generation

b) both labeled data and unlabeled data are required to feed into the same model classifier → generated cluster/class id heterogeneous among different clients

Method

The overall objective consists of three parts:
1) fundamental semi-supervised loss for all data;
2) an uncertainty-aware regularization loss to reduce the training gap among locally unseen and globally unseen classes;
3) a calibration loss to achieve efficient model aggregation

$$\mathcal{L}_i^* = \mathcal{L}_i + \beta \mathcal{R}_i + \gamma \mathcal{L}_i^{cal}$$

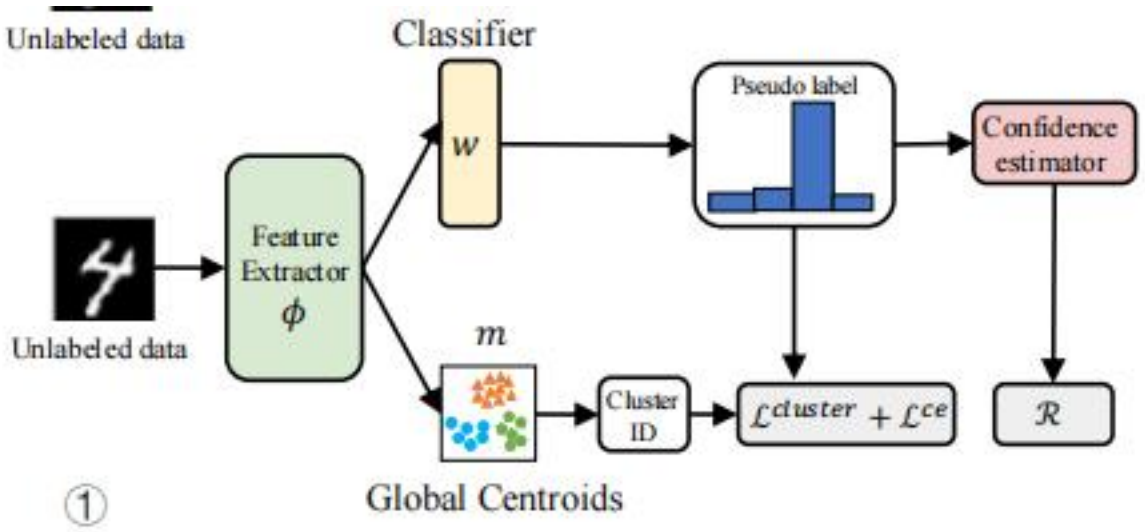
Method

UNCERTAINTY-AWARE LOSS

$$\mathcal{R}_i = \frac{1}{n_i^u} \sum_{x_j^u \in \mathcal{D}_i^u} |\pi(x_j^u)|$$

$$\pi(x_j^u) = \rho(n^c | \arg\max_c p(x_j^u; \theta))[1 - \max_c p(x_j^u; \theta)]$$

$$\rho(n^c) = -\tau^{1-\frac{n^c}{n_{\max}}}$$



CALIBRATION MODULE

$$\mathcal{L}_i^{cal} = \mathcal{L}_i^{ce} + \mathcal{L}_i^{cluster}$$

$$\mathcal{L}_i^{ce} = \frac{1}{n_i^u} \sum_{z_j \in Z_i^u} \mathcal{H}(q(z_j; m), p(w^\top \cdot z_j))$$

corresponding cluster assignments computed by matching representations with global centroids

$$\mathcal{L}_i^{cluster} = \frac{1}{n_i^u} \sum_{z_j, \bar{z}_j \in Z_i^u, \bar{Z}_i^u} \mathcal{H}(q(z_j; m), q(\bar{z}_j; m))$$

Method

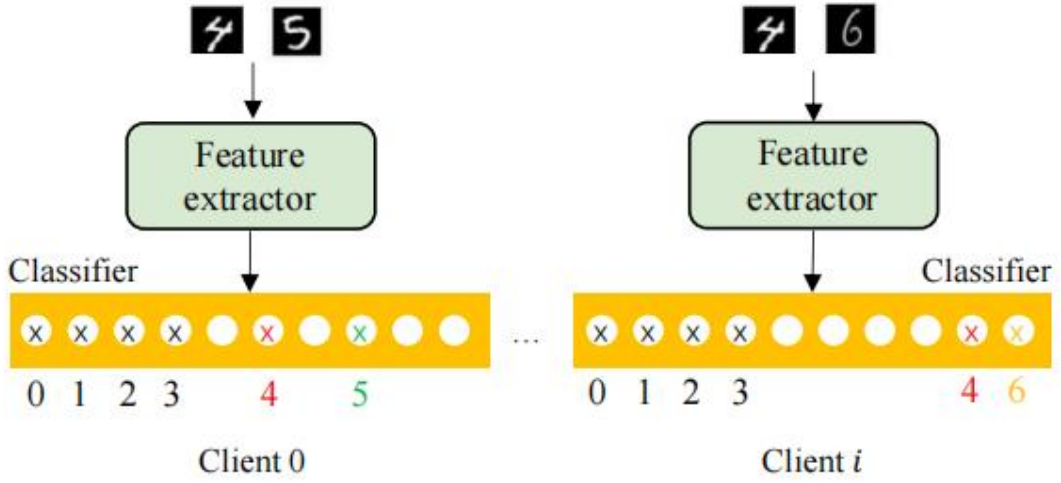


Figure 2: Illustration of label heterogeneity in FedoSSL. In a 10-class classification example, classes $\{0, 1, 2, 3\}$ are seen classes, while classes $\{4, 5, 6\}$ are unseen classes. Due to the feature-level pair-wise unsupervised loss (i.e., $\mathcal{L}_i^u$) on unlabeled data, same unseen class would be classified with different label id on different clients, e.g., unseen class 4 would be classified into the sixth position of the client 0's classifier, while in client i class 4 would be classified into the eighth position of the classifier.

Method

Algorithm 1 FedoSSL Algorithm

Input: Number of clients K, learning rate η, local epochs E, labeled data $\{\mathcal{D}_1^l, \mathcal{D}_2^l, \ldots, \mathcal{D}_K^l\}$, unlabeled data $\{\mathcal{D}_1^u, \mathcal{D}_2^u, \ldots, \mathcal{D}_K^u\}$, hyperparameter α, β, γ

Output: Final model θ

Initialize the model parameter θ
repeat
  Sample a set of clients $\mathcal{S}$.
  for each client $i \in \mathcal{S}$ in parallel **do**
    for $j = 1$ **to** E **do**
      Update local model: $\theta_i \leftarrow \theta_i - \eta \nabla_\theta \mathcal{L}_i^*(\theta_i)$
    end for
    Calculate local centroids m_i
  end for
  Update global model: $\theta \leftarrow \frac{n_i}{\sum_{i \in \mathcal{S}} n_i n} \sum_{i \in \mathcal{S}} \theta_i$
  Update global centroids m
  Distribute θ and m to all clients
until Model converges

Experiments

Table 2: Classification accuracy of compared methods on seen, unseen and all classes with 10 clients over three benchmark datasets. Asterisk (∗) in *SemiFL denotes that the original methods cannot classify unseen classes (and we had to extend it). On unseen classes, *LU.* denotes locally unseen classes, while *GU.* denotes globally unseen classes. *AU.* represents the overall accuracy of all unseen classes. Gray rows indicate the upper bound of the model performance of FedoSSL.

#Method	CIFAR-10 (%)					CIFAR-100 (%)					CINIC-10 (%)				
	All	Seen	Unseen			All	Seen	Unseen			All	Seen	Unseen		
			LU.	GU.	AU.			LU.	GU.	AU.			LU.	GU.	AU.
Cen-O	78.26	86.63	-	-	71.95	56.92	73.68	-	-	44.28	69.32	83.18	-	-	58.86
Cen-N	81.02	89.47	-	-	74.64	58.98	75.10	-	-	46.82	71.89	83.82	-	-	62.89
Local-O	65.98	79.57	-	-	45.60	43.10	54.33	-	-	26.25	55.33	65.23	-	-	40.48
Local-N	67.67	83.95	-	-	43.26	45.28	57.24	-	-	27.34	57.31	65.70	-	-	44.73
Fed-AO	69.46	81.01	89.38	42.03	52.15	47.91	59.67	38.07	29.12	30.26	54.85	63.22	71.31	37.88	42.29
Fed-RO	71.72	82.22	89.84	53.43	55.96	47.72	59.79	44.13	28.86	29.62	57.16	62.26	72.24	42.09	49.50
Fed-AN	66.58	84.18	78.76	37.58	40.15	47.25	58.24	42.11	30.44	30.77	53.49	63.61	66.78	36.06	38.32
Fed-RN	68.83	**85.52**	79.84	41.79	43.81	48.02	59.4	**48.77**	30.36	30.96	58.11	65.97	68.81	39.01	46.33
*SemiFL	64.91	81.57	86.33	31.16	39.92	42.28	54.94	31.68	21.46	23.29	52.27	62.72	64.53	37.21	37.34
FedoSSL	**76.26**	84.29	**90.68**	**59.69**	**64.22**	**51.58**	**61.12**	45.76	**33.82**	**31.13**	**63.82**	**68.40**	**79.79**	**47.78**	**56.96**

O:ORCA
N:NACH
Fed-A:FedAvg
Fed-R:FedRep

seen:unseen = 60% : 40% **select 50% of the seen classes as the label data**
CIFAR-10/CINIC-10 = 6 seen classes : 3 local unseen classes : 1 global unseen class
each client = all 6 seen classes, 1 local unseen class , 1 global unseen class
CIFAR-100 = 60 seen classes : 30 local unseen classes : 10 global unseen classes

Ablation Study

Table 3: Analysis of Loss function: classification accuracy on CIFAR-10 (the number of clients: 10).

METHOD	SEEN	UNSEEN	ALL
FED-AO	81.01	52.15	69.46
FEDOSSL-$\mathcal{R}_i$-$\mathcal{L}_i^{ce}$	83.53	52.24	71.01
FEDOSSL-$\mathcal{R}_i$	83.13	62.98	75.07
FEDOSSL	84.29	64.22	76.26

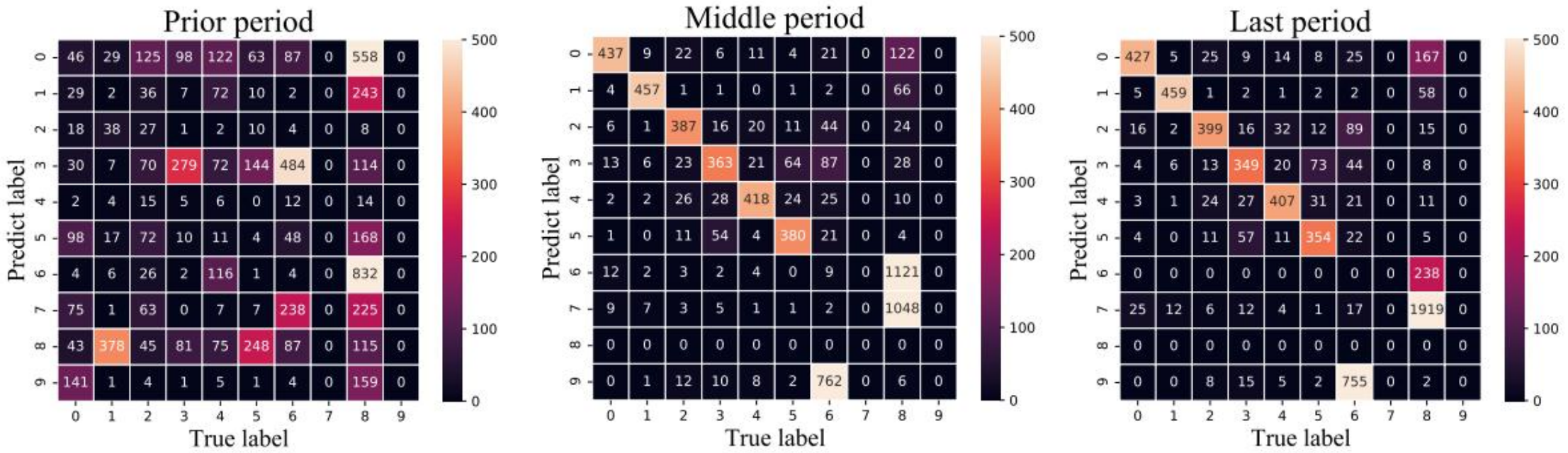
Table 4: Analysis of Loss function: classification accuracy on CINIC-10 (the number of clients: 10).

METHOD	SEEN	UNSEEN	ALL
FED-AO	63.22	42.29	54.85
FEDOSSL-$\mathcal{R}_i$-$\mathcal{L}_i^{ce}$	69.10	40.31	57.58
FEDOSSL-$\mathcal{R}_i$	67.59	47.73	59.65
FEDOSSL	68.40	56.69	63.82

$$\mathcal{L}_i^* = \mathcal{L}_i + \beta\mathcal{R}_i + \gamma\mathcal{L}_i^{cal}$$

$$\mathcal{L}_i^{cal} = \mathcal{L}_i^{ce} + \mathcal{L}_i^{cluster}$$

Experiments



(a) In initial stage. (b) In intermediate stage. (c) In final stage.

Figure 3: Visualization of the predicted clustering assignments in different training stages.

Experiments

Table 5: Classification accuracy of compared methods on seen, unseen and all classes with 50 clients over three benchmark datasets.

	CIFAR-10 (%)			CIFAR-100 (%)			CINIC-10 (%)		
#Method	All	Seen	Unseen	All	Seen	Unseen	All	Seen	Unseen
Fed-AO	70.22	83.34	50.54	45.63	56.25	29.69	53.81	60.49	43.80
Fed-RO	71.36	84.31	51.93	45.18	56.78	27.79	57.26	61.70	50.61
Fed-AN	69.89	85.36	46.68	45.22	56.30	28.59	53.42	63.62	38.13
Fed-RN	71.49	**86.28**	49.30	45.57	56.79	28.73	57.81	65.29	46.60
FedoSSL	**76.41**	85.71	**62.46**	**47.01**	**58.34**	**30.17**	**64.02**	**69.56**	**55.71**

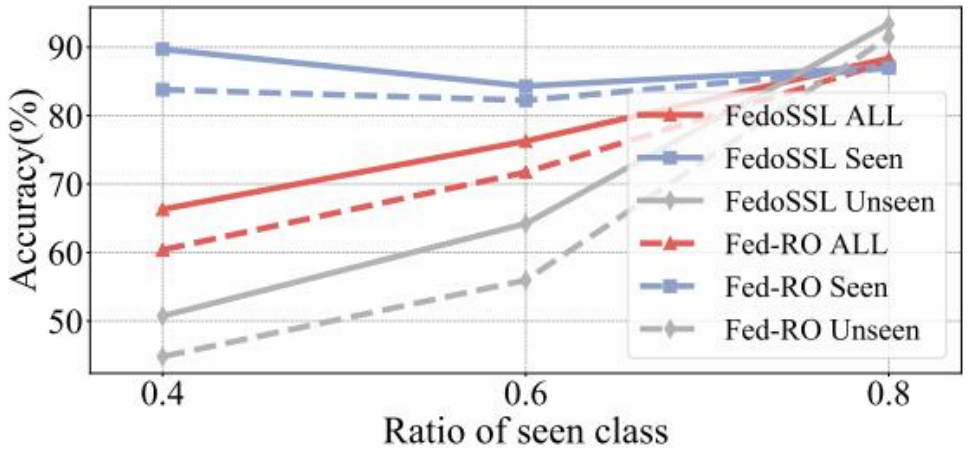


Figure 4: Performance of Fed-RO and FedoSSL with different numbers of seen classes on CIFAR-10.

Table 6: Sensitivity to number of local clusters on CIFAR-10. The number of global centroids is 10.

L	All	Seen	Unseen		
			LU.	GU.	AU.
8	74.28	84.26	88.90	54.09	59.29
16	75.76	84.17	89.28	58.36	63.15
32	76.26	84.29	90.68	59.69	64.22

Experiments

Table 7: Accuracy obtained using different privacy-guarnteed version of FedoSSL on CIFAR-10. 'No Privacy' represents the idealized setting when local representations are shared with the server. The number of global centroids is 10.

	All	Seen	Unseen		
			LU.	GU.	AU.
No Privacy	77.19	85.95	89.76	58.77	64.05
K-anonymity	76.26	84.29	90.68	59.69	64.22

- FedoSSL uses **Sinkhorn-Knopp based clustering algorithm** to compute L equally-sized local clusters. This operation enables a **n/L anonymity privacy guarantee** across all n samples present on a client (Lubana et al., 2022).

Experiments

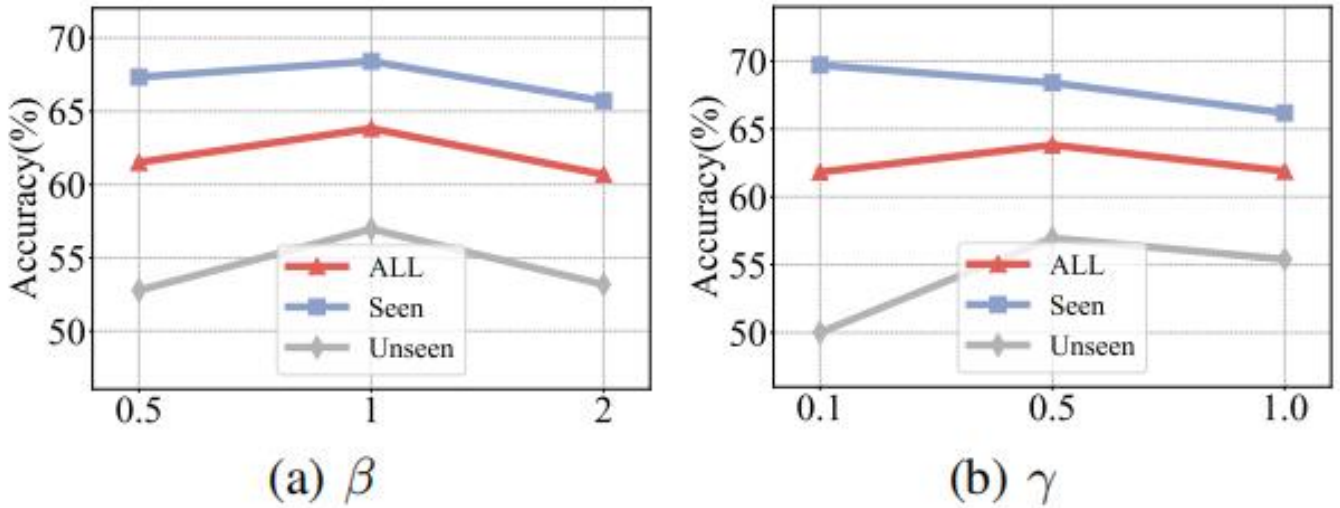


Figure 5: Classification accuracy on different setting of β (a) and γ (b).

$$\mathcal{L}_i^* = \mathcal{L}_i + \beta\mathcal{R}_i + \gamma\mathcal{L}_i^{cal}$$

Thanks