FedGCR : Federated Open Set Recognition Using Gaussian Noises in Client's Representation

Ankita Das
Dept. of Computer Science and Engineering
Indian Institute of Technology Hyderabad, Kandi, India
cs23resch01003@iith.ac.in

Mrinmay Sen
Dept. of Artificial Intelligence
Indian Institute of Technology Hyderabad, Kandi, India
senmrinmay@alumni.iith.ac.in

Rahul Biswas
Dept. of Computer Science and Engineering
Indian Institute of Technology Hyderabad, Kandi, India
cs22resch01004@iith.ac.in

C. Krishna Mohan
Dept. of Computer Science and Engineering
Indian Institute of Technology Hyderabad, Kandi, India
ckm@cse.iith.ac.in

***Abstract*—Open Set Recognition (OSR) aims to correctly classify known classes while detecting unknown samples, relaxing the closed-set assumption of traditional classification methods, which presume that both training and testing data contain samples from the same set of classes. However, most existing OSR methods operate in centralized settings, which often require sharing sensitive data. Federated Open Set Recognition (FedOSR) addresses this by enabling collaborative learning without exchanging raw data; however, current FedOSR methods still rely on sharing client-level statistics (e.g., mean and variance), which can leak private information. To address this issue, we propose FedGCR, a privacy-preserving FedOSR framework that trains a binary classifier for unknown-class detection using noise-added representations of local data, alongside the main model trained for known class prediction. Unlike existing methods, FedGCR avoids sharing any data statistics and transmits only the main known-class classifier and the binary unknown-detection model to the server. Experiments on benchmark image classification datasets show that FedGCR achieves superior unknown-class detection while maintaining strong performance on known classes and preserving client data privacy.**

***Index Terms*—Federated learning, open set recognition**

I. Introduction

Deep learning has achieved strong performance across many domains but relies on large centralized datasets, which are often impractical due to privacy, security, and regulatory constraints, particularly in sensitive areas such as healthcare and location-based services. These limitations hinder collaborative model training across organizations.

Federated learning (FL) [1]–[3] addresses this challenge by enabling clients to collaboratively train a global model without sharing raw data, exchanging only local model updates. As a result, FL has been widely adopted in privacy-critical domains including healthcare [4], finance [5], and IoT [6], [7].

However, most FL methods operate under a closed-set assumption, where the model is trained only on predefined classes. When exposed to unseen classes at inference time, such models incorrectly force predictions into known categories, severely limiting their reliability in dynamic real-world settings.

Open Set Recognition (OSR) aims to classify known classes while detecting unknown ones, but existing OSR methods [8]–[14] are primarily designed for centralized training. Recent federated OSR (FedOSR) approaches, such as FedOSS [15], improve unknown detection but require sharing class-level statistics (e.g., means and variances) with the server, which introduces non-trivial privacy leakage risks.

To overcome the limitations of existing OSR and FedOSR methods, we propose FedGCR, a federated open-set recognition framework that achieves effective unknown-class detection and accurate known-class prediction while preserving data privacy. In FedGCR, each client jointly trains a main classifier for known classes and a separate binary classifier for known–unknown discrimination, which is trained on noise-augmented local representations. Specifically, the local model is split into two parts: Base and Head, where the Base extracts low-dimensional representations and the Head performs classification. Unknown detection relies on Base representations, as feature extractors remain more stable under distribution shifts than classification layers [16], [17]. To simulate unknown samples, Gaussian noise with varied labels is added to the representations produced by the Base. These noise-augmented representations, combined with the original ones, are used to train the binary classifier. After local training, both the main models and binary classifiers from all the clients are sent to the server, where they are aggregated separately using FedAvg algorithm [1] for model aggregation to form a global model for known class prediction and a global binary classifier for unknown detection. During inference, a test sample is first processed by the global Base and evaluated by the binary classifier; samples predicted as known are then passed to the global classifier for final prediction. Extensive experiments on benchmark datasets demonstrate that FedGCR outperforms state-of-the-art methods while maintaining strong privacy guarantees.

Our contributions can be summarized as follows:

- We propose FedGCR, a novel federated open-set recognition (FedOSR) framework that addresses the privacy

TABLE I: Comparison of Open Set Recognition (OSR), Federated Learning (FL), and Federated Open Set Recognition (FedOSR) methods

Category	Method	Training Setup	Open-Set Capability	Federated	Privacy Risk	Key Limitation
OSR	OpenMax [8]	Centralized	✓ (activation-based)	✗	High	Assumes centralized data; weak under domain shift.
OSR	PROSER [9]	Centralized	✓ (placeholder classes)	✗	High	Limited generalization; not scalable.
OSR	MSCDNet [18]	Centralized	✓ (attention + contrastive)	✗	High	Not designed for federated or privacy-preserving settings.
FL	FedAvg [1]	Federated	✗	✓	Low	Closed-set assumption only.
FedOSR	FedOSS [15]	Federated	✓ (synthetic unknowns)	✓	Medium	Client statistics sharing causes privacy leakage.
Ours	**FedGCR**	Federated	✓(representation-level unknown modeling)	✓	**Low**	**Addresses boundary ambiguity without client data leakage.**

limitations of existing centralized and federated OSR methods. FedGCR trains a separate binary classifier at each client for detecting unknown-class samples, alongside the model for known class prediction.

- To generate samples for unknown-class detection, we introduce a noise-injection strategy. Gaussian noise is added to the local data representations extracted using the base of the client model and combined with the original representations to train the binary classifier. This eliminates the need to share any data statistics with the server, thus preserving client data privacy.
- We evaluate the effectiveness of FedGCR through extensive experiments on multiple benchmark image classification datasets.

II. Related Works

Open Set Recognition (OSR) aims to correctly classify known classes while detecting previously unseen samples, but most existing methods are designed for centralized learning. Representative approaches include OpenMax [8], which models unknowns using extreme value theory on activation distributions, and PROSER [9], which introduces placeholder classes to simulate unknown categories during training. Recent work such as MSCDNet [18], extends centralized OSR to fine-grained visual classification using attention-based feature aggregation and contrastive learning to improve known–unknown separability. Despite strong performance, these methods assume full data access and do not address privacy or data heterogeneity, limiting their applicability to decentralized settings (Table I). Federated Learning (FL) addresses privacy concerns by enabling collaborative training without sharing raw data [1], with prior work focusing on communication efficiency, non-IID data, and privacy preservation [19]–[21]. However, most FL methods operate under a closed-set assumption and fail to detect unknown-classes. Federated Open Set Recognition (FedOSR) extends FL to open-set scenarios; for example, FedOSS [15] synthesizes virtual unknowns to improve unknown detection. Nevertheless, FedOSS relies on sharing client-level statistics, introducing privacy risks and additional communication overhead (Table I).

III. Proposed Method

In this section, we describe the problem formulation and provide a detailed overview of our proposed method.

A. *Problem Formulation*

In traditional federated learning algorithm, the global model is trained on distributed data across clients with the assumption that the test sample contains the samples from same known classes as the training data. Our objective is to relax this assumption in federated learning by separately training a global binary classifier ($\boldsymbol{\phi}'$) for detecting unknown-class along with the main global model $\mathbf{x} = (\boldsymbol{\theta} + \boldsymbol{\phi})$ for predicting known class, where $\boldsymbol{\theta}$ represents the base and $\boldsymbol{\phi}$ represents the classifier of the main global model. To achieve this, we first optimize the local objective F_i of each client i as shown in Eq. 1 to compute the local main model $\mathbf{x}_i = (\boldsymbol{\theta}_i + \boldsymbol{\phi}_i)$, where $\mathbb{D}_i$ is the dataset owned by client i, $\xi \in \mathbb{D}_i$ and $\mathcal{L}_j$ is the loss function for j-th sample.

$$\min_{\mathbf{x}_i} F_i(\mathbb{D}_i; \mathbf{x}_i) = \min_{\mathbf{x}_i} \frac{1}{|\mathbb{D}_i|} \sum_{j=1}^{|\mathbb{D}_i|} \mathcal{L}_j(\xi; \mathbf{x}_i) \tag{1}$$

Then we train a binary classifier $\boldsymbol{\phi}_i'$ for unknown-class detection by optimizing the following objective:

$$\min_{\boldsymbol{\phi}_i'} {F_i}'(\mathbb{G}; \boldsymbol{\phi}_i') = \min_{\boldsymbol{\phi}_i'} \frac{1}{|\mathbb{G}|} \sum_{k=1}^{|\mathbb{G}|} \mathcal{L}_k'(\xi'; \boldsymbol{\phi}_i') \tag{2}$$

where, $\mathbb{G} = \mathbb{B}_{OR} \cup \mathbb{B}_{NR}$, $\boldsymbol{\theta}_i : \mathbb{D}_i \rightarrow \mathbb{B}_{OR}$ outputs the representations ($\mathbb{B}_{OR}$) of the local data ($\mathbb{D}_i$), $\mathbb{B}_{NR}$ is noise added representation of $\mathbb{B}_{OR}$, $\xi' \in \mathbb{G}$ and $\mathcal{L}_k'$ is the binary classification loss.

Once the local main model and local binary classifier are trained in each client, server collects them from all the clients and follows FedAvg [1] to aggregate them separately to form global main model and the global binary classifier.

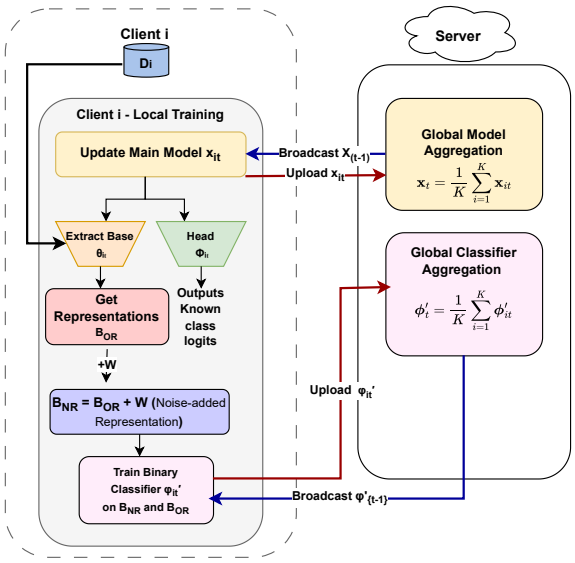

Fig. 1: Overall framework of the proposed FedGCR method. Each client extracts local representations $\mathbb{B}_{OR}$ from its data $\mathbb{D}_i$ using the base encoder θ_{it}. Noise vectors $\mathbf{w} = \{w_1, w_2, \ldots, w_P\}$ generate auxiliary representations $\mathbb{B}_{NR}$, which along with $\mathbb{B}_{OR}$ are used to update the local binary classifier ϕ'_{it}. The server aggregates client models x_{it} and classifiers ϕ'_{it} to form global parameters x_t and ϕ'_t, which are broadcast back for the next round.

B. *Overview of FedGCR*

The proposed FedGCR framework is outlined in Algorithm 1. In each communication round t, each client i receive global main model $\mathbf{x}_{t-1}$ along with the global binary classifier $\boldsymbol{\phi}'_{t-1}$ from the server. Then the main global model is again updated to $\mathbf{x}_{it}$ using local data and local optimizer. Each client then trains a binary classifier $\boldsymbol{\phi}'_{it}$ to distinguish known from unknown samples. This binary classifier is trained using noise-augmented local data representations. The main local model consists of two parts: one is the base ($\boldsymbol{\theta}_{it}$) for the feature extractor, and the second is the head ($\boldsymbol{\phi}_{it}$) for classification. For unknown-class prediction, we use the Base's output since it remains stable across data distributions, while the classification layers are more sensitive. This is motivated by findings in [16] and [17]. Unknown samples are simulated by adding Gaussian noise with varied labels to Base representations, as shown in Fig 1. These, along with original representations, train the binary classifier. After local training, both main and binary models are sent to the server, where they are aggregated via FedAvg [1] into global main model for known classification and unknown detection. During inference, an unseen sample is first processed by the global model's Base to obtain its representation, which is then passed to the binary classifier to determine whether it belongs to a known or unknown-class. If classified as known, it is subsequently passed to the global model's classifier for final prediction.

Algorithm 1 FedGCR

Input: $\mathbf{x}_0$: Initial global main model; $\boldsymbol{\phi}'_0$: Initial global binary classifier; K: Number of clients; T: Total communication rounds; η_m: Learning rate for main model; η_b: Learning rate for binary classifier; $\mathbf{w}$: Noise vector.
Output: Main global model $\mathbf{x}_T$ and global binary classifier $\boldsymbol{\phi}'_T$.
for $t = 1$ to T **do**
  Local Training:
  for all client $i \in \{1, 2, \ldots, K\}$ **in parallel do**
    Receive $\mathbf{x}_{t-1}$ and $\boldsymbol{\phi}'_{t-1}$ from the server
    Compute updated local main model $\mathbf{x}_{it}$
    Obtain representations $\mathbb{B}_{OR}$ of local data $\mathbb{D}_i$ using base $\boldsymbol{\theta}_{it}$ of $\mathbf{x}_{it}$
    Add noise $\mathbf{w}$ to get $\mathbb{B}_{NR}$ (unknown-sample representations)
    Update binary classifier $\boldsymbol{\phi}'_{it}$ using $\mathbb{B}_{OR}$ and $\mathbb{B}_{NR}$
    Send $\mathbf{x}_{it}$ and $\boldsymbol{\phi}'_{it}$ to the server
  end for
  Server Aggregation:
  $\mathbf{x}_t \leftarrow \frac{1}{K}\sum_{i=1}^{K} \mathbf{x}_{it}$
  $\boldsymbol{\phi}'_t \leftarrow \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{\phi}'_{it}$
end for

Algorithm 2 Get Representations and Add Noise

Input: $\mathbb{D}_i$: Local data; $\mathbf{w} = \{w_1, w_2, \ldots, w_P\}$: Noise vector; $\boldsymbol{\theta}_{it}$: Base of local model; N: Number of mini-batches.
Output: Overall original representations $\mathbb{B}_{OR}$ and overall noise-added representations $\mathbb{B}_{NR}$.
for $n = 1$ to N **do**
  Sample mini-batch $\mathbb{D}_{in} \subseteq \mathbb{D}_i$
  Compute representation $\boldsymbol{\theta}_{it}(\mathbb{D}_{in}) : \mathbb{D}_{in} \rightarrow \mathbb{B}_{ORn}$
  Add Gaussian noise $\mathcal{N}(0, w_j)$ to $\mathbb{B}_{ORn}$ to obtain $\mathbb{B}_{NRn}$
  Select w_j rotationally from $\mathbf{w}$ across mini-batches
end for
  Overall original representations $\mathbb{B}_{OR} = \{\mathbb{B}_{ORn}\}$
Overall noise added representations $\mathbb{B}_{NR} = \{\mathbb{B}_{ORn}\}$

*1) **Training main local model**:* Once a local client receives the global main model $\mathbf{x}_{t-1}$, it updates it to $\mathbf{x}_{it}$ using local data $\mathbb{D}_i$ and the stochastic gradient descent (SGD) optimizer [22] (with learning rate = η_m), by optimizing the objective function in Equation 1, following a procedure similar to local training in FedAvg. The main model $\mathbf{x}_{it}$ comprises two components: the base ($\boldsymbol{\theta}_{it}$), which serves as the feature extractor, and the head ($\boldsymbol{\phi}_{it}$), responsible for classification. The base $\boldsymbol{\theta}_{it}$ is further used to train the binary classifier for unknown-class detection.

*2) **Training local binary classifier**:* Once the training of the local main model $\mathbf{x}_{it}$ is completed, we extract its base $\boldsymbol{\theta}_{it}$ to generate embeddings or representations of the local data. The local dataset $\mathbb{D}_i$ is passed through the base $\boldsymbol{\theta}_{it}$, producing outputs $\mathbb{B}_{OR}$, referred to as the original representations, which correspond to known classes. To simulate samples from unknown-classes, we add a set of P Gaussian noises $\mathbf{w} = \{w_1, w_2, \ldots, w_P\}$ to the original representations $\mathbb{B}_{OR}$. The original representations are divided into N mini-batches, and noise from the set $\mathbf{w}$ is added in a rotating fashion across mini-batches to ensure diversity in the noise-injected samples. This process of generating original and noise-augmented representations is detailed in Algorithm 2. The noise-added representations, denoted as $\mathbb{B}_{NR}$, are considered as representative samples for unknown-classes. The binary classifier $\boldsymbol{\phi}'_{t-1}$ received from the server is then updated to $\boldsymbol{\phi}'_{it}$ using both the original and noise-augmented representations by optimizing the binary classification objective defined in Equation 2, using the SGD optimizer (with learning rate = η_b).

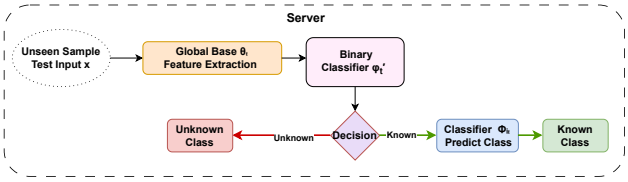

Fig. 2: Inference pipeline: An unseen test sample is processed through the global base for feature extraction, then classified by the binary classifier as either known or unknown before final prediction.

*3) **Server aggregation**:* Once main local models $\{\mathbf{x}_{it}\}$and binary classifiers $\{\phi'_{it}\}$ are trained in all the clients, Server aggregate them separately and compute the global main model $\mathbf{x}_t = \frac{1}{K}\sum_{i=1}^{K}\mathbf{x}_{it}$ and the global binary classifier $\phi'_t = \frac{1}{K}\sum_{i=1}^{K}\phi'_{it}$.

*4) **Inference**:* During inference (Fig. 2), an unseen sample is first processed by the global model's Base to obtain its representation, which is then passed to the global binary classifier to determine whether it belongs to a known or unknown-class. If classified as known, it is subsequently passed to the global model's classifier for final prediction. For unknown-class prediction, we use the representation generated by the Base of the global model trained for known classes, as the Base typically remains relatively stable in neural networks, even when trained on data with diverse distributions. In contrast, the classification layers are more sensitive to such distributional changes. This design choice is inspired by the findings in [16] and [17].

IV. Experiments

This section demonstrates our experimental setup, evaluation metrics, and results, along with the datasets, model, and compared existing methods.

A. Datasets

In our experiments, we use the CIFAR-10, CIFAR-100 and TinyImageNet datasets [23] [24] for training. These datasets consist of RGB images of size $3\times32\times32$, categorized into 10 and 100 classes, respectively, and are widely used benchmarks in computer vision research. Each dataset contains 50,000 training samples and 10,000 test samples. TinyImageNet is a more challenging dataset with 200 classes and RGB images of size $3\times64\times64$, consisting of 100,000 training images and 10,000 validation images.

For heterogeneous data partitioning across $K = 20$ clients, we adopt the widely used Dirichlet distribution-based strategy following [25], with a Dirichlet concentration parameter of $\alpha = 0.5$. We conduct our experiments in two settings for each dataset.

*a) **Setting 1**:* The whole training data is divided into 20 clients. During inference, we use the test data for known prediction, and for unknown prediction, we use the SVHN dataset [26].

*b) **Setting 2**:* The samples from first 50% of classes of the training data is divided across $K = 20$ clients. Evaluation for known class prediction is done using first 50% classes of the test data, and evaluation for unknown-class detection is carried out using the last 50% classes of the test data.

While evaluating for unknown-classes, the label for unknown samples is converted to 1 for binary classification.

B. Model and Loss Function

We use ResNet-9 [27] as the backbone network for all methods in federated multi-class image classification on CIFAR-10 and CIFAR-100, employing the categorical cross-entropy loss. For TinyImageNet, we adopt ResNet-18 as the backbone due to its higher visual complexity and larger number of classes. For the binary classifier, we use the final layer of the respective backbone with two output neurons, trained using the binary cross-entropy loss.

C. Evaluation Metrics

To evaluate the classification performance of the proposed FedGCR framework in open-set scenarios, we use test accuracy, defined as $\frac{\text{Number of true predictions}}{\text{Number of samples}}$, on both known and unknown-class test data.

D. Compared Methods

To validate the effectiveness of our proposed method, we compare it with existing open-set recognition (OSR) methods, including SoftMax, OpenMax [8], PROSER [9], and FedOSS [15], as well as a standard federated learning method, FedProx [2]. Descriptions of these algorithms are provided in Section I. For fair comparison, we use federated versions of the OpenMax, SoftMax, and PROSER algorithms, where local models are trained using the respective OSR methods, and the global model is obtained by aggregating local models using FedAvg [1].

E. Implementation details

For our proposed method, we conducted experiments on CIFAR-10 and CIFAR-100 datasets with full client participation with 20 clients. To achieve the best performance, we used this set of learning rates $\in \{0.1, 0.01, 0.001, 0.0001\}$ for all the methods. We use two different learning rates for different training stages, like pretraining and fine-tuning stage. The Adam optimizer is used in both stages to update the model parameters. We have experimented on 8 distinct noise levels $\{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 1.1,\ 1.3,\ 1.5\}$ for CIFAR-10, CIFAR-100, and TinyImageNet datasets. For FedProx we use proximal term $\mu = 1$ to help stabilize client optimization. We use OpenMax with a Weibull tail set to 30, Weibull alpha set to 3, and Weibull threshold of 0.9. Additionally, we implement PROSER with γ value of 1. Each client is trained for 1 local epoch on CIFAR-10 and 10 local epochs on CIFAR-100 and TinyImageNet under both Setting 1 and Setting 2 to obtain the reported results. We use 100 federated learning (FL) communication rounds for CIFAR-10, CIFAR-100, and 200 communication rounds for TinyImagenet. We pre-train

TABLE II: Comparison of known-class (K), unknown-class (U), and average (Avg) accuracies across datasets and settings.

Method	CIFAR-10						CIFAR-100						TinyImageNet					
	Setting 1			Setting 2			Setting 1			Setting 2			Setting 1			Setting 2		
	K	U	Avg	K	U	Avg	K	U	Avg	K	U	Avg	K	U	Avg	K	U	Avg
OpenMax	0.8116	0.4290	0.6203	0.8096	0.4290	0.6193	0.4652	0.8056	0.6354	0.4652	0.8145	0.6398	0.4877	0.6120	0.5498	0.3842	0.7650	0.5746
SoftMax	0.3761	0.9899	0.6830	0.8354	0.3598	0.5976	0.2558	0.5464	0.4011	0.4826	0.7712	0.6269	0.3951	0.5298	0.4624	0.2215	0.6120	0.4167
PROSER	0.8245	0.6756	0.7500	0.7620	0.3692	0.5656	0.2783	0.9684	0.6233	0.2580	0.9120	0.5850	0.3100	0.9226	0.6163	0.3577	0.7211	0.5394
FedProx	0.4269	0.0000	0.2134	0.5394	0.0000	0.2697	0.4379	0.0000	0.2189	0.4346	0.0000	0.2173	0.3839	0.0000	0.1919	0.3019	0.0000	0.1509
FedOSS	0.5621	0.7536	0.6578	0.4280	0.7910	0.6095	0.5090	0.9631	0.7360	0.5090	0.9818	**0.7454**	0.4240	0.9513	0.6876	0.4574	0.8313	0.6493
FedGCR (Ours)	0.7792	0.9960	**0.8876**	0.7544	1.0000	**0.8772**	0.5554	1.0000	**0.7777**	0.4850	0.9862	0.7356	0.5066	0.9910	**0.7488**	0.4680	0.8801	**0.6740**

the model for all the methods using five communication rounds with FedAvg. For fair comparisons, we use the same initialization for all the methods by using seed value of 0. Our experiments are conducted using NVIDIA A100-SXM4-40GB and Tesla V100-SXM3-32GB GPUs, with CUDA version 12.2 and PyTorch version 2.4.0.

F. Results

*1) **Comparisons with existing FL methods on CIFAR-10 dataset**:* In this section, we present the performance of various methods on the CIFAR-10 dataset under two settings, named Setting 1 and Setting 2, to assess the performance of FedGCR. As shown in Table II, traditional methods like OpenMax have shown a higher known accuracy (81.16%) and less unknown accuracy (42.90%) for Setting 1, therefore, OpenMax shows limitations in balancing known and unknown accuracy. Although SoftMax achieves high unknown accuracy in Setting 1 (98.99%), its poor known accuracy leads to average performance. Similarly, FedProx shows inconsistent performance where its unknown accuracy drops to zero. While PROSER performs well under Setting 1, however its unknown accuracy reduces to 36.92% in Setting 2. FedOSS shows inconsistent performance, particularly in Setting 2. In contrast, FedGCR outperforms all other methods across both settings. FedGCR can achieve highest average accuracies of 88.76% in Setting 1 and 87.72% in Setting 2.

*2) **Comparisons with existing FL methods on CIFAR-100 dataset**:* In Table II, we present a performance comparison between the proposed FedGCR framework and existing methods on CIFAR-100 dataset under two different settings with 10 local epochs. In Setting 1, FedGCR achieves the highest known accuracy of 55.54% and unknown accuracy of 100%, resulting in an average accuracy of 77.77%. This greatly outperforms the previous state-of-the-art FedOSS, which achieves 73.60% average accuracy. PROSER performs better in detecting unknown accuracy (96.84%) but struggles in known class detection (27.83%), resulting in average accuracy of 62.33%. SoftMax and FedProx gives weaker results overall, with FedProx failing to detect unknown-class. In Setting 2, FedGCR maintains unknown accuracy of 98.62% but shows a slight drop in known class accuracy 48.50% gives, average accuracy of 73.56%, which surpasses all compared baselines except for FedOSS. Although FedOSS outperforms FedGCR in this setting with a higher average accuracy of 74.54%, FedGCR's average performance remains competitive and close to FedOSS while maintaining privacy. Moreover, FedGCR's perfect unknown recognition in Setting 1 and consistent high performance for both settings highlight its robustness. Compared with PROSER, OpenMax, and SoftMax, FedGCR shows considerable improvement in average accuracy, such as 15.44% over PROSER, 14.23% over OpenMax, and 37.66% over SoftMax in Setting 1.

TABLE III: FedGCR performance under different noise gaps on CIFAR-10 (Setting 1). Reported values are Known (K), Unknown (U), and Avg accuracies.

Noise Gap	First / Last 4			All 8		
	K	U	Avg	K	U	Avg
0.8	0.7824	1.0000	0.8912	0.7786	0.0120	0.3953
0.4	0.7824	1.0000	0.8912	0.7786	0.0206	0.3996
0.2	0.7824	1.0000	0.8912	0.7792	0.9860	0.8826
0.05	0.7824	1.0000	0.8912	0.7786	0.0475	0.4130

*3) **Comparisons with existing FL methods on TinyImageNet dataset**:* We further evaluate FedGCR on the challenging TinyImageNet dataset using a ResNet-18 backbone with 10 local epochs and 200 global rounds. Due to its larger class count and higher visual diversity, most baselines show degraded performance. OpenMax and PROSER exhibit imbalanced behavior with high unknown but poor known accuracy (e.g., PROSER: K = 31.00%, U = 92.26% in Setting 1), while SoftMax is inconsistent, and FedProx completely fails at unknown detection. FedOSS improves unknown recognition via virtual unknown synthesis (U = 95.13% in Setting 1) but suffers from reduced known accuracy, limiting its average performance (68.76%). In contrast, FedGCR achieves the best trade-off, obtaining the highest average accuracy of 74.88% in Setting 1 and 67.40% in Setting 2, with near-perfect unknown detection (99.10%) and superior known-class performance.

*4) **Parameter Sensitivity Analyses**:* In our experiments, we evaluate the impact of varying noise levels on the performance of proposed FedGCR method using the CIFAR-10 dataset under Setting 1. Specifically, we vary the noise gap from 0.05 to 0.8 and generate sequences of noise values accordingly. The values for each noise gap were generated using the rule:

$$x_i = x_{i-1} + \text{noise gap} \quad \text{for } i = 1, 2, \ldots, 7$$

starting from an initial value of $x_0 = 0.1$. The index i increases sequentially from 1 to 7 to produce a total of 8 values, including the initial one. As a result, each sequence contains progressively increasing noise levels. For example, with a noise gap of 0.2, the resulting noise sequence

TABLE IV: Performance of FedGCR on CIFAR-10 under Setting 1 with varying fractions of client participation.

Fraction of Clients	Known Acc.	Unknown Acc.	Avg. Acc.
1.0 (All clients)	0.7792	0.9960	0.8876
0.5 (Half clients)	0.7407	1.0000	0.8704

is: $[0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5]$, and similarly for other gaps such as 0.05, 0.4, and 0.8. Eight noise values were generated for each configuration. For each noise gap, we evaluate model performance under three scenarios: using only the first four noise levels (lower noise), using all eight noise levels, and using only the last four noise levels (higher noise). The detailed results across different noise gaps and configurations are summarized in Table III, which reports the known, unknown, and average classification accuracies on the CIFAR-10 dataset under Setting 1.

*5) **Effect of Client Participation**:* Since our goal is accurate unknown-class detection, we primarily use full client participation, following FedOSS, and additionally evaluate robustness under 50% participation. As shown in Table IV on CIFAR-10 with 20 clients, FedGCR achieves its best performance with full participation (K = 77.92%, U = 99.60%). When participation is reduced to 50%, unknown accuracy slightly improves to 100%, likely due to reduced influence of noisy or heterogeneous unknown samples during aggregation.

V. Conclusions and Future Work

In this work, we introduce a novel framework called FedGCR, designed to address unknown-class detection as well as preserve privacy. Unlike prior methods, which rely on statistics of local data thereby risking privacy. FedGCR eliminates this problem by using noise added representation of the input data and by training a binary classifier for unknown detection. Extensive experiments on multiple benchmark image classification datasets confirm that FedGCR achieves superior performance in open-set recognition compared to existing state-of-the-art methods. In future, we plan to incorporate open-set recognition in vertical federated learning.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[4] D. C. Nguyen, Q. Pham, P. N. Pathirana, M. Ding, A. Seneviratne, Z. Lin, O. A. Dobre, and W. Hwang, "Federated learning for smart healthcare: A survey," *ACM Comput. Surv.*, vol. 55, no. 3, pp. 60:1–60:37, 2023.

[5] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," *CoRR*, vol. abs/2108.10749, 2021.

[6] Y. Jiang, B. Ma, X. Wang, G. Yu, P. Yu, Z. Wang, W. Ni, and R. P. Liu, "Blockchained federated learning for internet of things: A comprehensive survey," *ACM Comput. Surv.*, vol. 56, no. 10, p. 258, 2024.

[7] H. N. C. Neto, J. Hribar, I. Dusparic, D. M. F. Mattos, and N. C. Fernandes, "A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends," *IEEE Access*, vol. 11, pp. 41 928–41 953, 2023.

[8] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1563–1572.

[9] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Learning placeholders for open-set recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4401–4410.

[10] C. Geng, S.-j. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.

[11] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[12] H.-M. Yang, X.-Y. Zhang, F. Yin, Q. Yang, and C.-L. Liu, "Convolutional prototype network for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2358–2370, 2020.

[13] D. Miller, N. Sunderhauf, M. Milford, and F. Dayoub, "Class anchor clustering: A loss for distance-based open set recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3570–3578.

[14] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Open-set recognition: A good closed-set classifier is all you need?" 2021.

[15] M. Zhu, J. Liao, J. Liu, and Y. Yuan, "Fedoss: Federated open set recognition via inter-client discrepancy and collaboration," *IEEE Transactions on Medical Imaging*, vol. 43, no. 1, pp. 190–202, 2023.

[16] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 2089–2099.

[17] A. Ahmad, W. Luo, and A. Robles-Kelly, "Robust federated learning under statistical heterogeneity via hessian-weighted aggregation," *Machine Learning*, vol. 112, no. 2, pp. 633–654, 2023.

[18] S. Zhong and D. Wan, "Multi-stream collective deliberation network for open-set fine-grained visual classification," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–8.

[19] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[20] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[21] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[22] N. Ketkar, "Stochastic gradient descent," in *Deep learning with Python: A hands-on introduction*. Springer, 2017, pp. 113–132.

[23] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[24] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[25] M. Yurochkin, M. Agarwal, S. Ghosh, K. H. Greenewald, T. N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 7252–7261.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.