# FedLPA: Local Prior Alignment for Heterogeneous Federated Generalized Category Discovery

**Geeho Kim**[1] **Jinu Lee**[3] **Bohyung Han**[1,2]
[1]ECE & [2]IPAI, Seoul National University
[3]Toss (Viva Republica)
{snow1234, bhhan}@snu.ac.kr
jinu.lee@toss.im

## Abstract

Federated Generalized Category Discovery (Fed-GCD) requires a global model to classify seen classes and discover novel classes when data are siloed across heterogeneous clients. Existing GCD work often makes unrealistic assumptions, such as the need for prior knowledge of the number of novel classes or the assumption of uniform class distribution. We present Federated Local Prior Alignment (FedLPA), which eliminates these unrealistic assumptions by grounding learning in client-local structure and aligning predictions to client-local priors. Each client builds a similarity graph refined with reliable seen-class signals and discovers client-specific concepts and prototypes via Infomap. Leveraging the discovered concept structures, we introduce Local Prior Alignment (LPA): a self-distillation loss that matches the batch-mean prediction to an empirical prior computed from current concept assignments. The iterative process of local structure discovery and dynamic prior adaptation enables robust generalized category discovery under severe data heterogeneity. Our framework significantly outperforms existing federated generalized category discovery approaches on fine-grained and standard benchmarks, as demonstrated by extensive experimental results.

## 1 Introduction

Machine learning models increasingly need to operate in open-world settings where not all classes are available at the time of training. Generalized Category Discovery (GCD) [22, 3] aims to categorize unlabeled data that may include instances from both seen and novel classes, by leveraging knowledge transferred from the labeled set, while simultaneously classifying examples from the seen classes. However, existing studies on GCD have focused exclusively on centralized settings, assuming universal access to the training data, which comprises two types of classes—seen and novel—with annotations available for only a subset of the seen classes. In such setups, the number of seen classes is known, and the number of novel classes is typically assumed to be given. Yet, this centralized formulation overlooks a more practical and challenging scenario, where both data and computational resources are distributed across multiple clients.

We address the Federated Generalized Category Discovery (Fed-GCD) problem, which extends Generalized Category Discovery (GCD) to the federated learning (FL) setting. In this scenario, each local client independently manages its own training data without sharing it with others due to privacy constraints [15]. Although the core objective of Fed-GCD aligns with that of centralized GCD, it relies on a model trained in a privacy-preserving federated learning framework, which results in additional challenges for the following reasons. First, each client experiences more severe data heterogeneity and class imbalance [9, 8, 1], as training examples are partitioned across individual clients. Second, and more critically, the sets of classes may differ across clients. In other words,

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

in contrast to the prevalent assumption in centralized GCD [3, 27, 23, 25, 18, 29], each client only observes a *partial* class set from the whole label space and the total number of novel classes, even the count within its own local data, is unknown.

These challenges make most centralized GCD methods ill-suited to Fed-GCD. They rely on assumptions that do not hold in the federated setting. First, most methods require a priori knowledge of the total number of novel classes [3, 27, 23, 25, 18, 29] to configure their classifiers and loss functions. Second, these methods often have a strong assumption that class distribution is uniform. For instance, classifier-based methods like SimGCD [27] and its variants [25, 23] utilize entropy regularization to encourage balanced predictions across all classes (seen and novel), while clustering-based methods [20, 16, 30] employ balanced-cluster constraints to ensure the formation of distinct novel class clusters. Even recent federated methods inherit such assumptions; for example, FedoSSL [28] assumes the total number of novel classes across all clients is known, and AGCL [19] operates under a similar premise of uniform cluster distribution. These unrealistic assumptions directly conflict with the inherent data heterogeneity and class imbalances prevalent across clients in Fed-GCD.

To overcome these limitations, we propose a novel federated learning framework that operates without such unrealistic assumptions by discovering data structure at the client level. Instead of assuming a priori knowledge of the global number of novel classes, our approach empowers each client to construct its own local similarity graph from all its data, leveraging both ground-truth and high-confidence pseudo-labels. By applying Infomap clustering to this graph, the framework reveals client-specific class priors and concept prototypes, and an estimate of each client's novel class count.

Leveraging the discovered concept structures, we also introduce a novel self-distillation strategy, termed Local Prior Alignment (LPA), which enhances generalized category discovery on skewed local data by regularizing the model to align its batch-wise predictions for unlabeled examples with these discovered structures. This simple regularization effectively guides the model toward the true structure of each client's local data, enabling robust representation learning across heterogeneous clients. The proposed approach demonstrates remarkable performance improvements in all datasets and settings consistently, surpassing existing Fed-GCD baselines by significant margins.

Our main contributions are summarized as follows.

- We propose FedLPA, a client-level framework for federated generalized category discovery that requires no prior knowledge of the number of novel classes and tackles non-*i.i.d.* data.
- We discover client-specific categories by constructing a similarity graph from high-confidence seen-class labels and applying a graph-based clustering algorithm to obtain a local class prior and prototypes.
- We introduce Local Prior Alignment (LPA), a self-distillation strategy with a batch-level regularizer that aligns predictions with an empirical client prior, improving robustness under severe data heterogeneity.
- FedLPA demonstrates its outstanding performance in terms of robustness to client heterogeneity on fine-grained and generic benchmarks under various settings.

In the rest of this paper, we first review related works in Section 2 and discuss our main algorithm in Section 3. Section 4 presents our experimental results and Section 5 concludes this paper.

## 2 Related Works

### 2.1 Centralized generalized category discovery

The objective of Generalized Category Discovery (GCD), formulated by [22, 3], is to classify samples from seen categories and, at the same time, discover novel classes by leveraging knowledge from a labeled dataset. Unlike Novel Class Discovery (NCD) [7], which assumes unlabeled data contains only novel classes, GCD presents a more realistic and challenging setting by considering unlabeled data with both known and novel classes. Existing studies on GCD typically follow two main paradigms: parametric classifier learning and non-parametric representation learning.

Parametric methods [3, 27, 23, 25, 26, 14] build a learnable classifier and optimize it with the backbone using labelled data and pseudo-labelled data derived from model predictions. Specifically,

they incorporate adaptive margin [3] or entropy regularization [27, 23, 25, 14] for balanced pseudo-labeling, while mean teacher framework [23] or prompt-tuning [25] strategies for improved pseudo-labels. Besides, non-parametric methods employ combined contrastive losses [22], multiple projection heads [6], hierarchical [20] or concept-level contrastive loss [18], or Gaussian Mixture Models (GMMs) [30] to improve the generalization ability of features to novel categories. However, these approaches have focused on centralized settings, relying on assumptions that are ill-suited for the realistic federated learning setting. First, they often rely on an assumption that the ground-truth number of novel classes [3, 27, 23, 25, 20, 18] is given, or they require labeled validation data to estimate the class counts [22, 18, 20, 6]. Second, they often make the stronger assumption that class distribution is balanced. For instance, a mean entropy maximization (ME-MAX [2]) regularizer commonly adopted in parametric methods [27, 16, 23, 25] achieves novel class discovery by forcing uniform predictions over all classes (seen and novel). Some non-parametric approaches [20, 16, 30] employ balanced-cluster constraints to ensure the formation of distinct novel class clusters. Such assumptions about novel class counts and data uniformity are unrealistic in real-world distributed settings where data is partitioned heterogeneously across multiple clients.

### 2.2 Federated generalized category discovery

To address the limitations of centralized approaches, there has been increasing interest in Federated Generalized Category Discovery (Fed-GCD). Fed-GCD addresses the problem of GCD within the decentralized paradigm of Federated Learning (FL) [15], where clients collaboratively train a global model without sharing their raw data, thereby preserving privacy. The primary objective of Fed-GCD is to enable this global model to discover novel categories and accurately classify known categories present across all participating client datasets. This task is more challenging than centralized GCD due to severe data heterogeneity, so each client only observes a *partial* class set from the whole label space. An initial work, FedoSSL [28], addresses data heterogeneity in novel classes by introducing *locally unseen* (novel in some clients' unlabeled data) and *globally unseen* (novel in all clients' unlabeled data) classes. However, this approach relies on unrealistic assumptions that the total number of novel classes is known a priori and each client has *i.i.d.* and balanced seen class data. Recently, AGCL [19] tackles a more challenging Fed-GCD setting where both seen and novel classes exhibit highly skewed and non-*i.i.d.* distributions across clients, and employs GMM-based contrastive learning for robust representation learning of both seen and novel classes. However, this approach samples cluster instances uniformly based on the assumption that each cluster has an equal prior probability. Additionally, this method necessitates the communication of local class representations to the server, potentially introducing privacy leakage and increasing communication overhead. In contrast, our framework addresses these Fed-GCD challenges by robustly handling severe data heterogeneity, requiring no prior knowledge of the novel class count, no assumption of balanced distributions, and no communication of privacy-sensitive local representations.

## 3 Proposed Algorithm: FedLPA

This section presents our approach for federated generalized category discovery, referred to as FedLPA, which combines graph-based local category discovery and adaptive prior regularization.

### 3.1 Problem setup

We consider a federated learning (FL) setting with $N$ clients $\mathcal{C} = \{C_n\}_{n=1}^{N}$. Each client $C_n$ holds a local dataset $\mathcal{D}_n = \mathcal{D}_n^l \cup \mathcal{D}_n^u$, where $\mathcal{D}_n^l = \{(x_i, y_i)\}_{i=1}^{|\mathcal{D}_n^l|}$ contains labeled data with $y_i \in \mathcal{Y}_n^l$, and $\mathcal{D}_n^u = \{x_i\}_{i=1}^{|\mathcal{D}_n^u|}$ contains unlabeled data whose true (unknown) labels reside in $\mathcal{Y}_n^u$. The global set of known labels is $\mathcal{Y}^l = \bigcup_{n=1}^{N} \mathcal{Y}_n^l$, while the true global label space is $\mathcal{Y}^u = \bigcup_{n=1}^{N} \mathcal{Y}_n^u$. The global set of known labels $\mathcal{Y}^l$ is a subset of the true global label space $\mathcal{Y}^u$ (i.e., $\mathcal{Y}^l \subseteq \mathcal{Y}^u$). The classes in $\mathcal{Y}^u \setminus \mathcal{Y}^l$ constitute the set of novel classes, which are, by definition, present in the aggregated unlabeled data $\mathcal{D}^u = \bigcup_n \mathcal{D}_n^u$. The objective of Fed-GCD is to collaboratively train a global model $f : \mathcal{X} \to \mathcal{Y}^u$ using $\{\mathcal{D}_n\}_{n=1}^{N}$, enabling accurate classification of all instances in $\mathcal{D}^u$ into their true classes within $\mathcal{Y}^u$. In Fed-GCD, client data distributions may be heterogeneous, meaning local label sets $\mathcal{Y}_n^l$ and $\mathcal{Y}_n^u$ can vary across clients, and the cardinalities $|\mathcal{Y}^u|$ and $|\mathcal{Y}_n^u|$ are unknown. Due to privacy constraints in FL, transferring raw training data between clients is strictly prohibited.

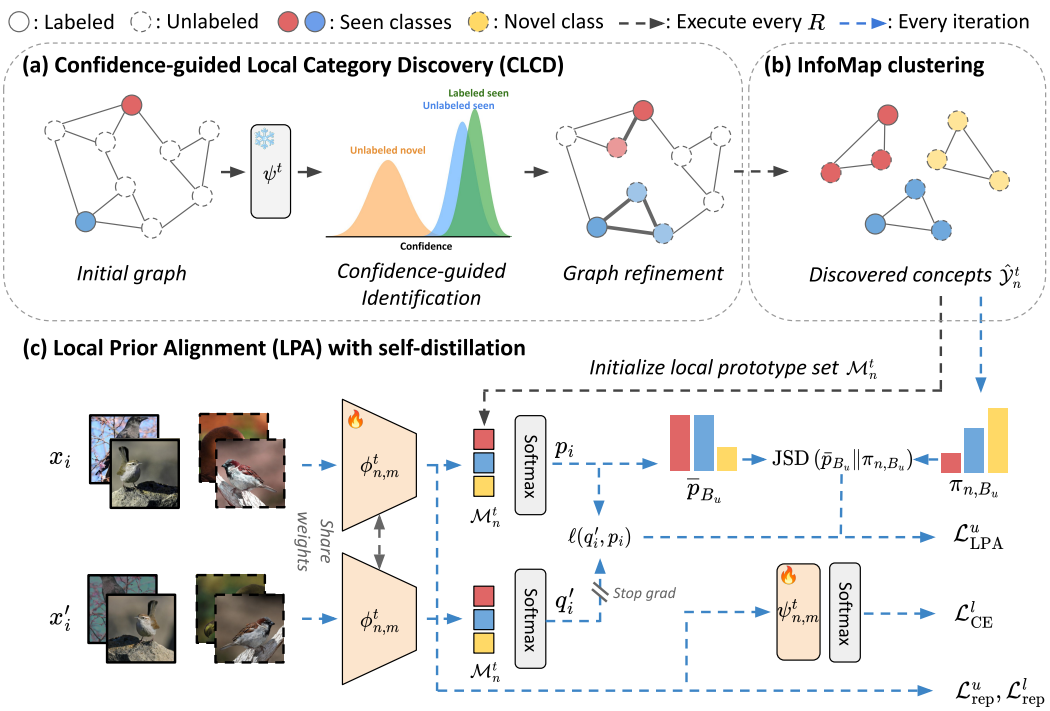



Figure 1: Overview of FedLPA's local training process. (a) Every $R$ rounds, each client builds a local similarity graph (via current global backbone $\phi^t$) and refines it using (pseudo-)labels from seen classes. (b) Each client applies graph-based clustering (Infomap) on the refined graph, and obtains a marginal local category prior $\hat{\mathcal{Y}}_n^t$ and corresponding concept prototypes $\mathcal{M}_n^t$. (c) During local training, for unlabeled data, a standard cross-entropy loss is applied with predictions $p_i$ and soft-targets $q_i'$, which are derived from similarities with the prototypes $\mathcal{M}_n^t$. Simultaneously, the average of the batch predictions $\bar{p}_{\mathcal{B}_u}$ is aligned with a dynamic batch prior $\pi_{n,\mathcal{B}_u}$ via Jensen-Shannon Divergence (JSD). Additionally, unsupervised contrastive loss is applied to unlabeled data, while labeled data utilizes standard cross-entropy for seen classes and supervised contrastive loss.

### 3.2 Overview

The main objective of FedLPA is robust generalized category discovery under severe data heterogeneity and class imbalance, without requiring knowledge of the number of novel classes or relying on balanced-class assumptions common in prior methods. As illustrated in Figure 1, we achieve this goal through three synergistic local stages: (1) Confidence-guided Local Category Discovery (CLCD) to build a richly supervised local similarity graph, (2) graph-based clustering algorithm (Infomap) to derive a local category prior and concept prototypes, and (3) Local Prior Alignment (LPA) to adaptively align model predictions using these discovered local priors. These stages collaboratively adapt to local data heterogeneity, fostering robust category discovery across heterogeneous clients.

FedLPA operates within a standard federated learning framework, FedAvg [15]. Specifically, a central server initializes a global model parameterized by $\theta = \{\phi, \psi\}$, corresponding to a feature extractor $f(\cdot;\phi)$, and a classifier $g(\cdot;\psi)$ dedicated to seen classes $\mathcal{Y}^l$. In communication round $t \in \{1, \ldots, T\}$, a central server sends a global model $\theta^t$ to the active client set $\mathcal{C}_t \subseteq \mathcal{C}$. Each client $C_n \in \mathcal{C}_t$ initializes its parameters $\theta_{n,0}^t$ to $\theta^t$, and performs $M$ iterations for optimization using its local data. The server collects the resulting local models $\theta_{n,M}^t$ and updates the global model $\theta^{t+1}$ for the next round of training by simply averaging the local model parameters. This training process is repeated until the global model $\theta^t$ converges.

### 3.3 Confidence-guided local category discovery

**Initial similarity graph construction** Each client $n$ constructs an initial similarity graph $G_n = \{\mathcal{I}_n, \mathcal{E}_n\}$ to capture pairwise feature-based relationships among its local samples $\mathcal{D}_n$. The nodes $\mathcal{I}_n$ represent all local samples $x_i \in \mathcal{D}_n$. Edge weights $e_{ij} \in \mathcal{E}_n$ are defined by the cosine similarity

between $\ell_2$-normalized features $v_i$ and $v_j$, which are the $\ell_2$-normalized outputs from the global backbone $f(\cdot;\phi^t)$ for the corresponding samples $x_i$ and $x_j$. This initial graph $G_n$ provides a foundational structure of pairwise relationships among the local samples.

**Confidence-guided identification of known samples** The initial graph, based solely on feature similarity, can be noisy and may not accurately reflect true semantic relationships. To mitigate this and establish a more reliable structure, each client refines the graph $G_n$ using supervisory signals from the known categories $\mathcal{Y}^l$ by identifying high-confidence pseudo-labels for unlabeled samples $x_i \in \mathcal{D}_n^u$. This is achieved by leveraging the current global model $\theta^t = \{\phi^t, \psi^t\}$ from the previous round (specifically, $\theta_{n,0}^t$ before local updates) to predict logits $h(x;\theta^t) = g(f(x;\phi^t);\psi^t)$ for seen classes $\mathcal{Y}^l$. The confidence score $s(x_i)$ for each unlabeled sample $x_i$ is the maximum softmax probability over these logits. If $s(x_i)$ exceeds a client-adaptive threshold $\xi_n$ (the $P$-th percentile of confidences from local labeled data $\mathcal{D}_n^l$), $x_i$ is deemed a reliably identified known sample. This sample is assigned a pseudo-label $\hat{y}_i \in \mathcal{Y}^l$ corresponding to the class with the highest confidence, forming a set:

$$\hat{\mathcal{D}}_n^{u,\text{seen}} = \{(x_i, \hat{y}_i) \mid x_i \in \mathcal{D}_n^u,\ s(x_i) > \xi_n,\ \text{and } \hat{y}_i = \arg\max_{k \in \mathcal{Y}^l} \sigma(h(x_i;\theta^t))_k\}, \tag{1}$$

where $\sigma(\cdot)$ denotes the softmax function applied to the logits $h(x_i;\theta^t)$ to obtain class probabilities. To ensure the classifier provides meaningful confidences, especially in early training stages, we employ initial warm-up training rounds. Further details on this warm-up procedure and the determination of $\xi_n$ are provided in the supplementary document.

**Label-informed graph refinement** With the full set of (pseudo-)labeled samples, each client now updates the edge weights $\mathcal{E}_n$ of its local similarity graph $G_n$ to reflect this supervisory information. Let $\mathcal{D}_n^{\text{sup}} = \mathcal{D}_n^l \cup \hat{\mathcal{D}}_n^{u,\text{seen}}$ be this set of samples with (pseudo-)labels $\tilde{y}_i \in \mathcal{Y}^l$. We update the initial edge weights $e_{ij} \in \mathcal{E}_n$ as follows:

$$e'_{ij} \leftarrow \begin{cases} 1, & \text{if } x_i, x_j \in \mathcal{D}_n^{\text{sup}}, \tilde{y}_i = \tilde{y}_j, \text{ and } i \neq j \\ 0, & \text{if } x_i, x_j \in \mathcal{D}_n^{\text{sup}}, \tilde{y}_i \neq \tilde{y}_j, \text{ and } i \neq j \\ e_{ij}, & \text{otherwise} \end{cases} \tag{2}$$

We also take an edge-pruning step to enhance the graph's robustness against noisy feature representations. This pruning step forms the final edge set $\mathcal{E}'_n$ by discarding the edges if their value does not exceed a predefined threshold $\tau_f$. The resulting refined graph is thus $G'_n = \{\mathcal{I}_n, \mathcal{E}'_n\}$, which provides a cleaner and more reliable structure for the subsequent local category discovery.

### 3.4 Infomap clustering

Each client then discovers its local concept structure from the refined graph $G'_n$. To achieve this, the client employs the Infomap algorithm [21], which partitions the graph into communities by minimizing the description length of a random walk. This process yields two key outputs for the client's local data $\mathcal{D}_n$: (1) a set of concept assignments $\{c_i\}$, effectively grouping the data into discovered concepts $\hat{\mathcal{Y}}_n^t$, and (2) an estimate of the number of these concepts, $K_n = |\hat{\mathcal{Y}}_n^t|$.

With these concept assignments, the client initializes a set of $K_n$ local prototypes, $\mathcal{M}_n^t = \{\mu_{n,k}^t\}_{k=1}^{K_n}$. Each prototype $\mu_{n,k}^t$ is the mean of the $\ell_2$-normalized feature vectors of all instances assigned to the corresponding concept $c_k$. These prototypes are pivotal, serving as anchors for the self-distillation mechanism described in Section 3.5. To ensure the prototypes remain aligned with the evolving feature space, this entire discovery and initialization process is repeated every $R$ communication rounds at the start of local training.

### 3.5 Local prior alignment (LPA) with self-distillation

Building upon the local category discovery, we introduce a novel self-distillation strategy incorporating a principled regularizer, termed Local Prior Alignment (LPA). During local training, the unsupervised objective for client $n$ on an unlabeled mini-batch $B_u \subset B$ is formulated as:

$$\mathcal{L}_{\text{LPA}}^u = \frac{1}{|B^u|} \sum_{x_i \in B^u} \ell(q'_i, p_i) + \varepsilon\, \text{JSD}(\bar{p}_{B^u} \,||\, \pi_{n,B^u}). \tag{3}$$

The objective consists of a self-distillation loss, based on the cross-entropy function $\ell(\cdot,\cdot)$, and the proposed LPA regularizer, with their relative importance balanced by the hyperparameter $\varepsilon$.

The self-distillation component refines feature representations by enforcing predictive consistency across augmented views of each unlabeled image. To achieve this, we generate two random augmentations, $x_i$ and $x_i'$. For the first view $x_i$, the model computes a soft probability distribution $p_i$ over the $K_n$ local concepts. Specifically, $p_i$ is the softmax distribution (with temperature $\tau_s$) of cosine similarities between the feature vector $v_i = f(x_i; \phi^t_{n,m})$ and the local prototypes in $\mathcal{M}^t_n$. Concurrently, a sharper pseudo-label distribution $q_i'$ (with lower temperature $\tau_t < \tau_s$) is generated for the second view $x_i'$, which serves as the soft target for $p_i$.

The LPA regularizer aligns the model's collective predictions with the underlying data structure of each batch. It achieves this by minimizing the Jensen-Shannon Divergence (JSD) between two distributions: the model's average prediction and a batch-specific empirical prior. The empirical prior, $\pi_{n,B_u}$, is calculated as the distribution of the pre-assigned concepts $\{c_j\}$ within the unlabeled mini-batch $B_u$:

$$\pi_{n,B^u}[k] = \frac{1}{|B^u|} \sum_{x_j \in B^u} \mathbb{I}(c_j = c'_k), \quad k \in [K_n], \tag{4}$$

where $c'_k$ is the $k$-th unique concept in the client's estimated concept set $\hat{\mathcal{Y}}^t_n$. The model's average prediction for the unlabeled batch, $\bar{p}_{B_u}$, is then computed by averaging the soft predictions from both augmented views:

$$\bar{p}_{B^u} = \frac{1}{|B^u|} \sum_{x_i \in B^u} \frac{1}{2}(p_i + p'_i), \tag{5}$$

where $p'_i$ is the prediction for the second view $x'_i$, also computed using temperature $\tau_s$. By minimizing the JSD between these two distributions, LPA adaptively steers the model toward the client's true local data structure, enhancing robustness against severe data skew.

### 3.6 Joint optimization

We also employ supervised [10] and self-supervised [4] contrastive losses for robust representation learning as

$$\mathcal{L}^l_{\text{rep}} = \frac{1}{|B^l|} \sum_{i \in B^l} \frac{1}{|\mathcal{N}_i|} \sum_{q \in \mathcal{N}_i} -\log \frac{\exp\left(v_i^\top v'_q / \tau_c\right)}{\sum_i^{i \neq j} \exp\left(v_i^\top v'_j / \tau_c\right)}, \tag{6}$$

$$\mathcal{L}^u_{\text{rep}} = \frac{1}{|B|} \sum_{i \in B} -\log \frac{\exp(v_i^\top v'_i / \tau_u)}{\sum_i^{i \neq j} \exp(v_i^\top v'_j / \tau_u)}. \tag{7}$$

where $\mathcal{N}_i$ indexes all other images in the same batch that hold the same label as $x_i$, while $\tau_c$ and $\tau_u$ are temperature parameters. Additionally, we employ a standard cross-entropy loss $\mathcal{L}^l_{\text{CE}}$ on $B_l$ with the ground-truth labels to train the local classifier $g(\cdot; \psi^t_{n,m})$ for seen classes. Thus, the overall objective function for each client $n$ is:

$$\mathcal{L}_n = \lambda(\mathcal{L}^u_{\text{LPA}} + \mathcal{L}^u_{\text{rep}}) + (1-\lambda)(\mathcal{L}^l_{\text{rep}} + \mathcal{L}^l_{\text{CE}}), \tag{8}$$

where $\lambda$ is a hyperparameter balancing the loss terms. The detailed learning procedure of FedLPA is described in Algorithm 1 in the supplementary document.

## 4 Experiment

### 4.1 Experimental setup

**Dataset** We evaluate our proposed method on six image classification benchmarks: three fine-grained datasets, CUB-200 [24], Stanford-Cars [11], and Oxford-IIIT Pet [17], three generic object recognition datasets, CIFAR-10 [12], CIFAR-100 [12], and ImageNet-100 [5]. For each dataset, we designate half of the classes as known and the other half as novel. From the known classes, 50% of instances form the labeled training subset, while the remaining instances, along with all instances from novel classes, constitute the unlabeled training subset. To simulate non-*i.i.d.* data distributions,

we sample label ratios from a Dirichlet distribution with a symmetric parameter $\alpha \in \{0.2, 0.05\}$, following [8, 19]. This partitioning results in $|\mathcal{C}| = 5$ subsets, each of which is considered a local dataset stored individually on each client.

**Baselines** We compare our method, dubbed as *FedLPA*, with the state-of-the-art Fed-GCD methods, which include GCL [19], and AGCL [19]. We also establish federated baselines by adapting prominent centralized GCD methods, GCD [22], SimGCD [27], and GPC [30], and an unsupervised learning method, PCL [13]. These are integrated with FedAvg [15], following the strategy in [19], and are denoted as *FedAvg + GCD*, *FedAvg + SimGCD*, *FedAvg + GPC*, and *FedAvg + PCL*, respectively. For *FedAvg + SimGCD*, the number of novel classes is assumed known a priori for classifier initialization.

**Evaluation protocol** We evaluate the model performance with clustering accuracy (ACC) on an unlabeled test set held by the server, following a standard practice in [19]. This test set is partitioned from a global evaluation set alongside a labeled validation set, mirroring the partitioning scheme of the training data. Note that the baselines [19, 30, 22, 13] utilize the labeled validation data for either category number estimation or semi-supervised clustering. To ensure a direct and fair comparison with these baselines, we also report the performance of a variant named FedLPA+, which utilizes this validation set by applying our CLCD algorithm to guide semi-supervised clustering.

Given predicted labels $\hat{y}_i$ and ground-truth labels $y_i$, ACC is defined as follows:

$$ACC = \max_{\Pi \in S_k} \frac{1}{N_u} \sum_{i=1}^{N_u} \mathbf{1}\{\hat{y}_i = \Pi(y_i)\}, \tag{9}$$

where $S_k$ is the set of all possible permutations of $k$ cluster assignments, $N_u$ is the total number of unlabeled test samples, and $\Pi(\cdot)$ is the optimal mapping found using the Hungarian algorithm. We report ACC for all unlabeled test samples ("All"), as well as separately for samples belonging to "Old" classes ($y_i \in \mathcal{Y}^l$) and "New" classes ($y_i \in \mathcal{Y}^u \setminus \mathcal{Y}^l$).

**Implementation details** We use a ViT-B/16 pretrained with DINO as the backbone. We use the output of the `[CLS]` token with a dimension of 768 as the feature for an image, and only fine-tune the last block of the backbone, following [19, 22]. The model undergoes a warmup stage of 20 rounds, followed by 50 rounds of Fed-GCD training. Both stages use SGD with a batch size of 128 and an initial learning rate of 0.1. For Fed-GCD training, the learning rate is decayed via a cosine schedule. Following [19], the number of local training epochs is set to 1 with full client participation. The balancing factor $\lambda$ is set to 0.35, the temperature values $\tau_s, \tau_c, \tau_u$ are set to 0.1, 0.07, 1.0, respectively. Following [27, 22], $\tau_t$ starts at 0.07 and anneals to 0.04 over the first 30 rounds using a cosine schedule. For FedLPA, we set the percentile $P$ to 80, the LPA regularization weight $\varepsilon$ to 0.5, and the CLCD update frequency $R$ as 1. We set $\tau_f$ to 0.6 and 0.4 for fine-grained datasets and standard datasets, respectively. All experiments were conducted on a single NVIDIA RTX A6000 or A5000 GPU.

### 4.2 Results

We compare the proposed methods, FedLPA and FedLPA+, on six benchmarks: three fine-grained datasets and three standard object recognition datasets, under two different data heterogeneity settings. Table 1 and Table 2 show that both FedLPA and FedLPA+ consistently outperform all existing Fed-GCD baselines across all datasets at every data heterogeneity level. FedLPA achieves these gains without any server-side labeled validation data, which is a common prerequisite for baselines. This indicates FedLPA's robustness in realistic, resource-constrained federated settings, even when the server-side labeled validation data is not available. For direct comparison, FedLPA+ leverages the server-held validation data and applies the proposed CLCD on the combined validation and unlabeled test sets, which further improves the performance in most cases. Among the baselines, FedAvg + SimGCD often struggles – especially on generic datasets – likely due to its restrictive uniform-prior assumption. Similarly, FedAvg + GPC performs worse than FedAvg + PCL in most cases because it assumes balanced clusters. These observations suggest that heuristic priors assuming data or cluster balance are ill-suited for the non-*i.i.d.* and imbalanced Fed-GCD settings, underscoring the advantages of our adaptive, data-driven structure discovery mechanisms.

Table 1: Results on fine-grained datasets with two different degrees of data heterogeneity. Bold numbers indicate the best accuracies. Methods with a dagger † report results from [19]. The 'Server val.' column indicates whether server-side labeled validation data is used for evaluation.

| Method | Server val. | CUB-200 $\alpha = 0.2$ All | Old | New | CUB-200 $\alpha = 0.05$ All | Old | New | Stanford-Cars $\alpha = 0.2$ All | Old | New | Stanford-Cars $\alpha = 0.05$ All | Old | New | Oxford-Pet $\alpha = 0.2$ All | Old | New | Oxford-Pet $\alpha = 0.05$ All | Old | New |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FedAvg + GCD† [22] | ✓ | 46.3 | 54.8 | 40.1 | 43.3 | 52.8 | 38.9 | 32.4 | 49.8 | 28.3 | 30.4 | 46.1 | 26.5 | 76.2 | 77.8 | 75.2 | 72.1 | 76.4 | 71.5 |
| FedAvg + SimGCD [27] | ✓ | 36.8 | 49.7 | 30.4 | 34.6 | 48.5 | 27.7 | 35.1 | 56.3 | 24.9 | 30.3 | 43.9 | 23.7 | 43.6 | 39.7 | 45.6 | 36.7 | 34.1 | 38.1 |
| FedAvg + PCL† [13] | ✓ | 51.3 | 53.5 | 49.8 | 47.5 | 53.0 | 46.3 | 35.3 | 47.7 | 33.4 | 32.6 | 45.5 | 29.2 | 79.4 | 80.3 | 79.1 | 76.6 | 77.9 | 74.7 |
| FedAvg + GPC† [30] | ✓ | 49.1 | 51.3 | 47.0 | 45.3 | 51.2 | 44.7 | 34.1 | 45.5 | 32.6 | 30.9 | 45.3 | 27.8 | 78.8 | 78.5 | 79.1 | 73.1 | 77.3 | 73.5 |
| FedAvg + GCL† [19] | ✓ | 53.7 | 54.6 | 53.2 | 52.2 | 53.1 | 52.9 | 36.0 | 48.1 | 33.7 | 35.3 | 45.7 | 31.5 | 80.7 | 81.3 | 80.2 | 79.5 | 81.5 | 78.6 |
| FedAvg + AGCL† [19] | ✓ | 55.2 | 52.5 | 56.7 | 53.1 | 52.9 | 54.2 | 38.2 | 50.8 | 36.0 | 36.4 | 44.9 | 32.8 | 82.7 | 83.9 | 82.3 | 81.4 | 82.0 | 80.7 |
| FedLPA (ours) | | 62.3 | 63.3 | 61.8 | 61.2 | 63.1 | 60.1 | 52.1 | 67.6 | 44.6 | 51.8 | 64.9 | 45.4 | 84.6 | 85.3 | 84.2 | 83.3 | 86.6 | 81.5 |
| FedLPA+ (ours) | ✓ | **63.5** | **63.6** | **63.4** | **62.6** | **64.3** | **61.8** | **57.7** | **70.1** | **51.7** | **54.2** | **69.4** | **46.8** | **86.7** | **90.7** | **84.7** | **85.0** | **86.9** | **84.0** |

Table 2: Results on standard object recognition datasets with two different degrees of data heterogeneity. Bold numbers indicate the best accuracies. Methods with a dagger † report results from [19]. The 'Server val.' column indicates whether server-side labeled validation data is used for evaluation.

| Methods | Server val. | CIFAR-10 $\alpha = 0.2$ All | Old | New | CIFAR-10 $\alpha = 0.05$ All | Old | New | CIFAR-100 $\alpha = 0.2$ All | Old | New | CIFAR-100 $\alpha = 0.05$ All | Old | New | ImageNet-100 $\alpha = 0.2$ All | Old | New | ImageNet-100 $\alpha = 0.05$ All | Old | New |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FedAvg + GCD† [22] | ✓ | 80.7 | 82.3 | 80.3 | 78.7 | 80.1 | 78.3 | 49.6 | 52.1 | 49.3 | 47.3 | 49.2 | 45.9 | 69.8 | 77.1 | 65.7 | 66.4 | 74.8 | 62.1 |
| FedAvg + SimGCD [27] | ✓ | 53.6 | 53.5 | 53.6 | 52.9 | 66.2 | 46.3 | 43.1 | 57.0 | 36.2 | 33.6 | 40.1 | 30.4 | 54.8 | 77.1 | 43.5 | 43.4 | 62.2 | 33.9 |
| FedAvg + PCL† [13] | ✓ | 81.6 | 82.7 | 80.9 | 80.0 | 80.7 | 79.4 | 53.2 | 54.1 | 51.7 | 50.4 | 51.6 | 49.0 | 72.4 | 79.5 | 66.0 | 70.1 | 77.0 | 63.3 |
| FedAvg + GPC† [30] | ✓ | 81.3 | 81.7 | 80.5 | 80.1 | 80.4 | 78.4 | 52.8 | 53.5 | 51.4 | 50.0 | 51.3 | 48.9 | 72.1 | 78.2 | 65.7 | 69.8 | 76.8 | 63.1 |
| FedAvg + GCL† [19] | ✓ | 83.2 | 84.9 | 82.8 | 82.2 | 82.4 | 81.9 | 54.1 | 55.7 | 54.0 | 52.1 | 53.2 | 51.9 | 74.1 | 81.8 | 67.3 | 72.5 | 79.8 | 65.3 |
| FedAvg + AGCL† [19] | ✓ | 84.7 | 85.5 | 84.6 | 82.5 | 83.4 | 82.2 | 56.1 | 56.8 | 55.3 | 54.2 | 54.6 | **54.0** | 74.8 | 80.2 | 69.8 | 73.1 | 78.1 | 67.0 |
| FedLPA (ours) | | 94.5 | 94.5 | **94.6** | 93.9 | 94.5 | **93.7** | 57.7 | 61.7 | **55.7** | 54.2 | 60.7 | 50.9 | 75.9 | 89.7 | 69.0 | 73.2 | 87.6 | 65.9 |
| FedLPA+ (ours) | ✓ | **95.1** | **96.3** | 93.9 | **94.1** | **95.1** | 93.3 | **58.1** | **63.4** | 55.4 | **56.5** | **64.9** | 52.3 | **76.6** | **90.6** | **69.9** | **74.4** | **88.6** | **67.3** |

### 4.3 Analysis

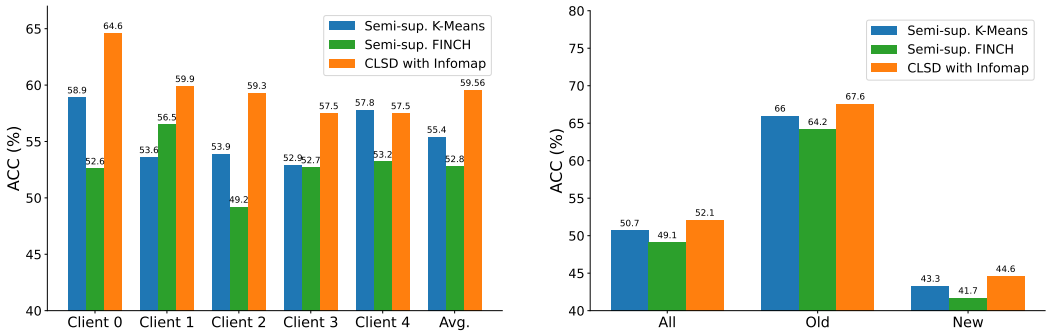
**Ablation study** To validate the efficacy of individual components within our FedLPA framework, we conduct an ablation study on Stanford-Cars under non-*i.i.d.* settings ($\alpha = 0.2$ and $\alpha = 0.05$), and the results are presented in Table 3. The results show that each proposed element contributes significantly to the final performance. Specifically, applying our Local Prior Alignment loss ($\mathcal{L}^u_{\text{LPA}}$) substantially enhances performance, even with a fixed target prior derived once from an initial graph of local unlabeled data (row 1). Our proposed regularizer on adaptive prior, empirically computed from each local batch, yields further significant gains (row 2). Notably, even without the confidence-guided graph refinement, our framework, LPA loss with an adaptive prior (row 3 and row 4), already demonstrates strong performance, outperforming all compared algorithms in Table 1. This highlights the robustness of our core LPA mechanism. Furthermore, our proposed confidence-guided graph refinement provides an additional performance gain (row 5).

**Analysis of CLCD algorithm** To validate the effectiveness of our proposed CLCD algorithm, we ablate the clustering module by replacing it with semi-supervised K-Means (used in [22, 30]) and semi-supervised FINCH (used in [19]) within the FedLPA framework, and the results are presented in Figure 2. It is important to note that for the implementation of both semi-supervised K-Means and FINCH, we assume that the knowledge of the true number of classes in both local data and the global test data is given. We report two outcomes: (a) clustering accuracy on each client's local data immediately after warmup (Figure 2a); and (b) final clustering accuracy on the server test set after 70 federated training rounds (Figure 2b). Our proposed method achieves superior clustering performance on local training data compared to the other algorithms. Consequently, this leads to a more significant improvement in the final test accuracy of FedLPA both on seen and novel classes, underscoring the effectiveness of the proposed CLCD algorithm on the overall model performance.

**Increased number of clients** We validate our framework in more challenging scenarios with an increased number of distributed clients ($N = 10$). All methods suffer from performance degradation,

Table 3: Component analysis of the proposed methods in the non-*i.i.d.* settings on Stanford-Cars with two different degrees of data heterogeneity ($\alpha = 0.2$ and $\alpha = 0.05$).

| $\mathcal{L}^u_{\text{LPA}}$ | Target prior | Initial graph | CLCD | $\alpha = 0.2$ | | | $\alpha = 0.05$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | All | Old | New | All | Old | New |
| - | - | - | - | 34.1 | 50.8 | 26.0 | 32.0 | 50.3 | 23.1 |
| ✓ | Fixed | $\mathcal{D}^u_n$ | - | 48.3 | 61.7 | 41.8 | 47.3 | 55.5 | 43.2 |
| ✓ | Adaptive | $\mathcal{D}^u_n$ | - | 51.4 | 66.1 | 44.3 | 50.5 | 61.6 | 45.1 |
| ✓ | Adaptive | $\mathcal{D}^l_n \cup \mathcal{D}^u_n$ | - | 54.4 | 67.9 | 47.2 | 52.3 | 66.4 | 45.4 |
| ✓ | Adaptive | $\mathcal{D}^l_n \cup \mathcal{D}^u_n$ | ✓ | **57.7** | **70.1** | **51.7** | **54.8** | **69.4** | **46.8** |




(a) Clustering accuracy on local data after warm-up

(b) Final clustering accuracy on server test data

Figure 2: Ablative results of CLCD algorithm in FedLPA under non-*i.i.d.* clients ($\alpha = 0.2$) on Stanford-Cars. We evaluate (a) clustering accuracy on individual client local training data right after the warmup training rounds, and (b) final clustering accuracy on the server test set after 70 federated training rounds. For the final clustering accuracy, all methods are evaluated identically at test time: we apply the same server-side Infomap clustering to the unlabeled test set with a fixed pruning threshold ($\tau_f = 0.6$), regardless of the clustering algorithm used during local training.

compared with the results in Table 4, due to the reduced local data per client, increased data disparity. Despite these challenges, FedLPA consistently shows promising performance on all tested datasets.

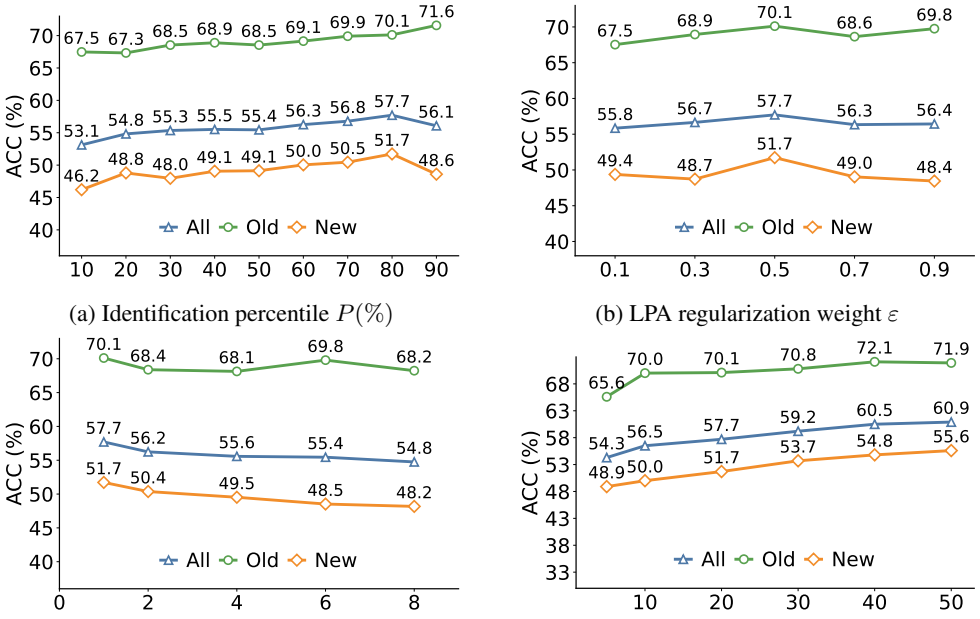
**Hyperparameters** We investigate the impact of our hyperparameters on the performance of FedLPA under a non-*i.i.d.* setting with $\alpha = 0.2$, and the results are presented in Figure 3. Both the CLCD identification percentile $P$ (Figure 3a) and the Local Prior Alignment (LPA) regularization weight $\varepsilon$ (Figure 3b) demonstrate robust performance across a reasonable range of values, indicating FedLPA's stability. For the CLCD update frequency $R$ (Figure 3c), while more frequent updates ($R = 1$) yield better results by enabling rapid adaptation, FedLPA maintains competitive accuracy even with sparser updates. This offers a valuable trade-off, allowing for reduced computational overhead with only a marginal performance decrease, beneficial in resource-constrained federated scenarios. For the number of warmup rounds (Figure 3d), while a marginal performance drop is observed with very few initial rounds, FedLPA rapidly achieves competitive accuracy with a modest number of rounds (e.g., 10-20). The performance generally exhibits an upward trend and stabilizes as the number of warmup rounds increases (e.g., up to 50 rounds), indicating that sufficient warmup is beneficial.

## 5 Conclusion

We introduced Federated Local Prior Alignment (FedLPA), a novel framework for generalized category discovery in heterogeneous federated environments. Unlike prior approaches often reliant on unrealistic global knowledge or fixed class priors ill-suited for federated settings, FedLPA operates entirely at the client level. It first constructs a client-specific similarity graph, enhanced by reliably pseudo-labeled known-class samples, to capture local data structures without requiring global information or pre-defined category counts. Building on this, our Local Prior Alignment (LPA) regularizer, integrated within a self-distillation scheme, dynamically adapts to local data distributions by aligning model predictions with an online empirical prior derived from these discovered structures. This

Table 4: Results with increased number of clients ($N = 10$) on standard benchmarks in non-*i.i.d.* setting ($\alpha = 0.05$). Methods with a dagger † report results from [19]. The 'Server val.' column indicates whether server-side labeled validation data is used for evaluation.

| Method | Server val. | CIFAR-10 | | | CIFAR-100 | | | ImageNet-100 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | All | Old | New | All | Old | New | All | Old | New |
| FedAvg + GCD† [22] | ✓ | 63.4 | 60.0 | 66.7 | 47.3 | 48.3 | 45.6 | 62.3 | 70.8 | 60.1 |
| FedAvg + GCL† [19] | ✓ | 68.2 | 64.2 | 70.1 | 52.5 | 53.9 | 51.0 | 67.3 | 74.5 | 60.8 |
| FedAvg + AGCL† [19] | ✓ | 68.1 | 63.8 | 70.3 | 52.2 | 53.6 | 52.4 | 67.5 | 74.8 | 61.1 |
| FedLPA (ours) | | 92.3 | 94.5 | 91.2 | 53.6 | 54.5 | 53.1 | 71.7 | 85.6 | 64.7 |
| FedLPA+ (ours) | ✓ | **93.7** | **96.2** | **92.3** | **55.9** | **59.7** | **54.0** | **72.1** | **86.8** | **64.7** |




(a) Identification percentile $P(\%)$

(b) LPA regularization weight $\varepsilon$

(c) CLCD frequency $R$

(d) The number of warmup rounds

Figure 3: Ablative results of FedLPA hyperparameters in non-*i.i.d.* clients ($\alpha = 0.2$) on Stanford-Cars. We examine the impact of: (a) the percentile $P$ for known sample filtering in CLCD; (b) the weight $\varepsilon$ for the LPA regularizer in Eq (3); (c) communication rounds $R$ between CLCD executions; and (d) the number of rounds for federated warmup training.

synergy of local structure grounding and dynamic prior adaptation enables robust category discovery under severe data heterogeneity and class imbalance, leading to substantial performance gains over existing Fed-GCD methods on diverse benchmarks.

**Limitations & Future Work** While FedLPA's per-round communication overhead is comparable to that of FedAvg, exploring ways to further reduce communication rounds—for example via adaptive client sampling or asynchronous updates—could be a promising direction for future work. Future work includes evaluating the framework under client churn and adversarial settings. In addition to these directions, we plan to extend the framework to multimodal data and streaming class discovery.

## Acknowledgements

This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grants [No.RS-2022-II220959 (No.2022-0-00959), (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making, No.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University), No.RS-2021-II212068, Artificial Intelligence Innovation Hub] funded by the Korean government (MSIT).

## References


[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021. 1

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022. 3

[3] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022. 1, 2, 3

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 6

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[6] Yixin Fei, Zhongkai Zhao, Siwei Yang, and Bingchen Zhao. Xcon: Learning with experts for fine-grained category discovery. In *BMVC*, 2022. 3

[7] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *CVPR*, 2019. 2

[8] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 1, 7

[9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. In *ICML*, 2020. 1

[10] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 6

[11] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 6

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[13] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021. 7, 8

[14] Haonan Lin, Wenbin An, Jiahao Wang, Yan Chen, Feng Tian, Mengmeng Wang, QianYing Wang, Guang Dai, and Jingdong Wang. Flipped classroom: Aligning teacher attention with student in generalized category discovery. In *NeurIPS*, 2024. 2, 3

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017. 1, 3, 4, 7

[16] Jona Otholt, Christoph Meinel, and Haojin Yang. Guided cluster aggregation: A hierarchical approach to generalized category discovery. In *WACV*, 2024. 2, 3

[17] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 6

[18] Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *CVPR*, 2023. 2, 3

[19] Nan Pu, Wenjing Li, Xingyuan Ji, Yalan Qin, Nicu Sebe, and Zhun Zhong. Federated generalized category discovery. In *CVPR*, 2024. 2, 3, 7, 8, 10

[20] Sarah Rastegar, Mohammadreza Salehi, Yuki M Asano, Hazel Doughty, and Cees GM Snoek. Selex: Self-expertise in fine-grained generalized category discovery. In *ECCV*, 2024. 2, 3

[21] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4):1118–1123, 2008. 5

[22] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022. 1, 2, 3, 7, 8, 10

[23] Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. Improving category discovery when no representation rules them all. In *NeurIPS*, 2023. 2, 3

[24] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[25] Hongjun Wang, Sagar Vaze, and Kai Han. SPTNet: An efficient alternative framework for generalized category discovery with spatial prompt tuning. In *ICLR*, 2024. 2, 3

[26] Ye Wang, Yaxiong Wang, Yujiao Wu, Bingchen Zhao, and Xueming Qian. Beyond known clusters: probe new prototypes for efficient generalized class discovery. *arXiv preprint arXiv:2404.08995*, 2024. 2

[27] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *ICCV*, 2023. 2, 3, 7, 8

[28] Jie Zhang, Xiaosong Ma, Song Guo, and Wenchao Xu. Towards unbiased training in federated open-world semi-supervised learning. In *ICML*, 2023. 2, 3

[29] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *CVPR*, 2023. 2

[30] Bingchen Zhao, Xin Wen, and Kai Han. Learning semi-supervised gaussian mixture models for generalized category discovery. In *CVPR*, 2023. 2, 3, 7, 8

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly claim that FedLPA enables generalized category discovery in federated, non-IID settings; Sections 1, 4 and 5 provide the algorithm and empirical evidence that substantiate this claim.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 ("Limitations & Future Work") articulates reliance on early-round feature quality, communication-round cost, and the need to study client churn and adversarial settings.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper contains no formal theorems or proofs; it contributes an algorithm and empirical study.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 details datasets, Dirichlet partitioning, model architecture, training schedule, hyperparameters, and evaluation metrics, providing all information required to reproduce the main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets are public benchmarks. The authors will release anonymized code and detailed instructions on GitHub upon acceptance so that anyone can reproduce the reported results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Datasets, split ratios, Dirichlet $\alpha$ values, optimizer (SGD), learning-rate schedule, temperature parameters, and other hyperparameters are provided in Section 5 (Experimental Setup).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For each experiment, we report the average accuracy over multiple independent runs with different random seeds (3 runs). While we do not include error bars in plots or tables due to space limitations, the averaging procedure reduces the impact of variance and reflects stable results across runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 states that experiments used NVIDIA RTX A6000/A5000 GPUs, batch 128, 50 communication rounds plus 20 warm-up rounds; this suffices to estimate compute cost.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Only public datasets are used; no personal or sensitive data is processed, and experiments follow standard ethical practices.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 6 highlights privacy benefits of FL and notes possible misuse risks in adversarial client scenarios.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The work does not release high-risk models or scraped datasets; only standard public benchmarks are used.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (e.g., CIFAR, CUB-200) and models (ViT-B/16 DINO) are cited with original references and used under their respective open licenses.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new dataset or pretrained model is released; only algorithmic code will be shared.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The study does not involve human participants or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human-subject research is included, so IRB approval is not required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No large-language model is used in the core methodology; any LLM-assisted editing was purely for language polishing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.