
FedLPA: Local Prior Alignment for Heterogeneous Federated Generalized Category Discovery

Supplementary Document

Geeho Kim¹ Jinu Lee³ Bohyung Han^{1,2}

¹ECE & ²IPAI, Seoul National University

³Toss (Viva Republica)

{snow1234, bhhan}@snu.ac.kr

jinu.lee@toss.im

A Additional Details of the Proposed Algorithm: FedLPA

In this section, we further detail two components within our proposed algorithm: first, federated warm-up training performed by all participating clients before Fed-GCD training; and second, the identification pipeline within Fed-GCD’s Confidence-guided Local Category Discovery (CLCD) stage for discovering unlabeled examples likely belonging to seen classes.

A.1 Federated warm-up training

The federated warm-up training phase prepares clients by learning a backbone suitable for GCD and a classifier for seen classes, the latter being crucial for the proposed CLCD phase. This is achieved through an Expectation-Maximization (EM)-like alternating optimization strategy. Initially, each client trains the backbone network using Eq. (6) from the main paper on the entire local dataset and Eq. (5) from the main paper for labeled data, similar to representation learning objectives in GCD [7]. Subsequently, the backbone is frozen, and the seen-class classifier is trained using a standard cross-entropy (CE) loss exclusively on labeled data. This alternating approach is vital: updating the feature extractor with the CE loss from the seen-class classifier biases representations toward seen classes, thereby hindering novel class discovery [2]. By freezing the feature extractor during classifier training, we mitigate this bias, enabling the effective and concurrent learning of robust representations and an accurate seen-class classifier.

After the local training, the global model is updated according to the federated learning pipeline detailed in Section 3.2, facilitating collaborative training across clients.

A.2 Confidence-guided identification of seen-class samples

Confidence-based filtering aims to obtain additional category information from unlabeled data by identifying examples likely belonging to seen classes. While some GCD methods utilize category information from labeled data for clustering [7, 8, 6], relying solely on the small labeled set provides sparse supervision, a limitation that intensifies with fewer labeled examples. In contrast, our approach incorporates the category information from these identified unlabeled seen-class examples into the edges of the similarity graph, thereby enriching the graph representation. This technique is motivated by our observation that confidence scores—defined as the maximum softmax probability over the seen-class classifier’s logits—exhibit distinct patterns. As illustrated in Figure 1 and Figure 2, examples of unseen classes within unlabeled data show markedly different confidence distributions compared to those of seen classes. Conversely, seen-class examples in unlabeled data demonstrate high confidence, with distributions closely mirroring those of the original labeled data.

Based on this observation, we implement the filtering pipeline as follows. First, for each client, confidence scores are computed for all its labeled data using the current global backbone ϕ^t and seen-class classifier ψ^t . These scores are then used to model the client’s labeled data confidence distribution via kernel density estimation (KDE) with a Gaussian kernel. A client-specific confidence threshold is subsequently established: it is the confidence value whose log-likelihood under this KDE corresponds to the P -th percentile of log-likelihoods derived from the client’s labeled data points. For unlabeled data, confidence scores are computed using the same global backbone and seen-class classifier. Unlabeled examples with confidence exceeding this threshold are identified as seen-class instances and assigned a cluster label corresponding to the class with the maximum classifier output. Note that this thresholding technique is client-adaptive, based on individual labeled data confidences, ensuring robust operation across highly heterogeneous clients.

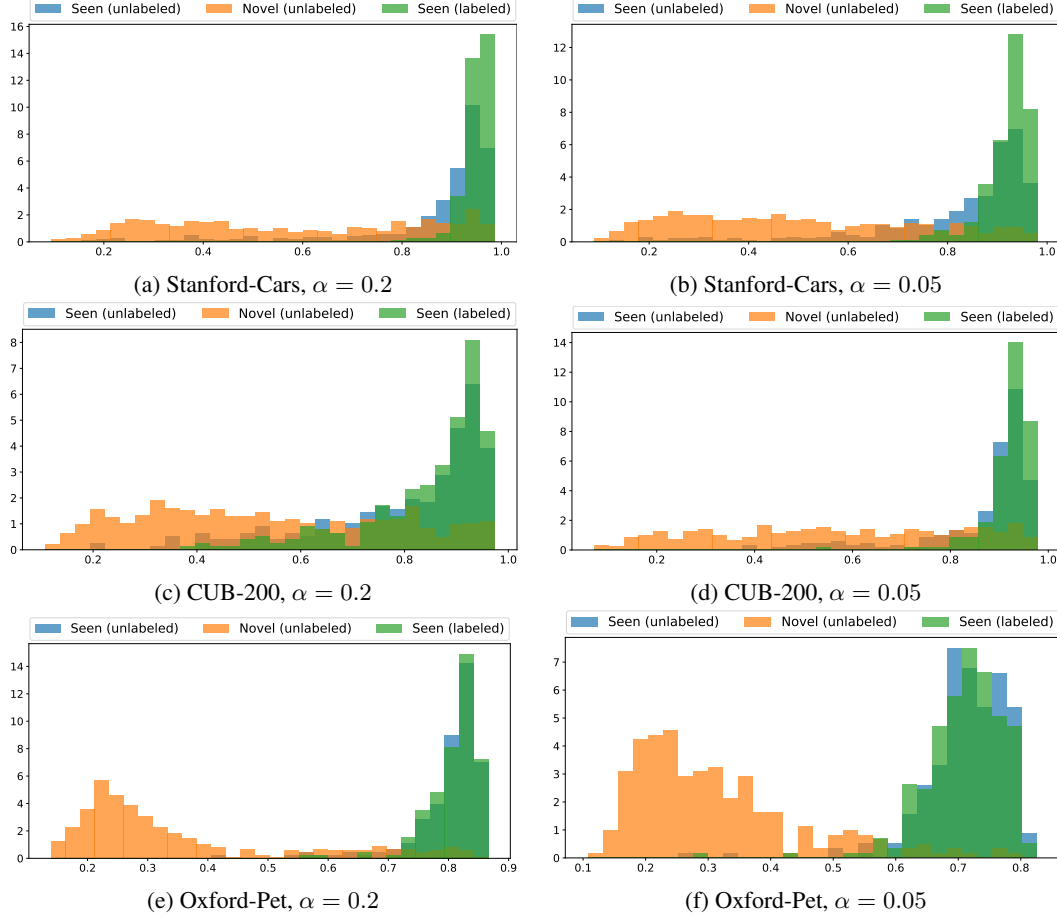


Figure 1: Normalized confidence score distributions (histogram) for labeled seen samples, unlabeled seen samples, and unlabeled novel samples from a single client on fine-grained datasets under non-*i.i.d.* settings.

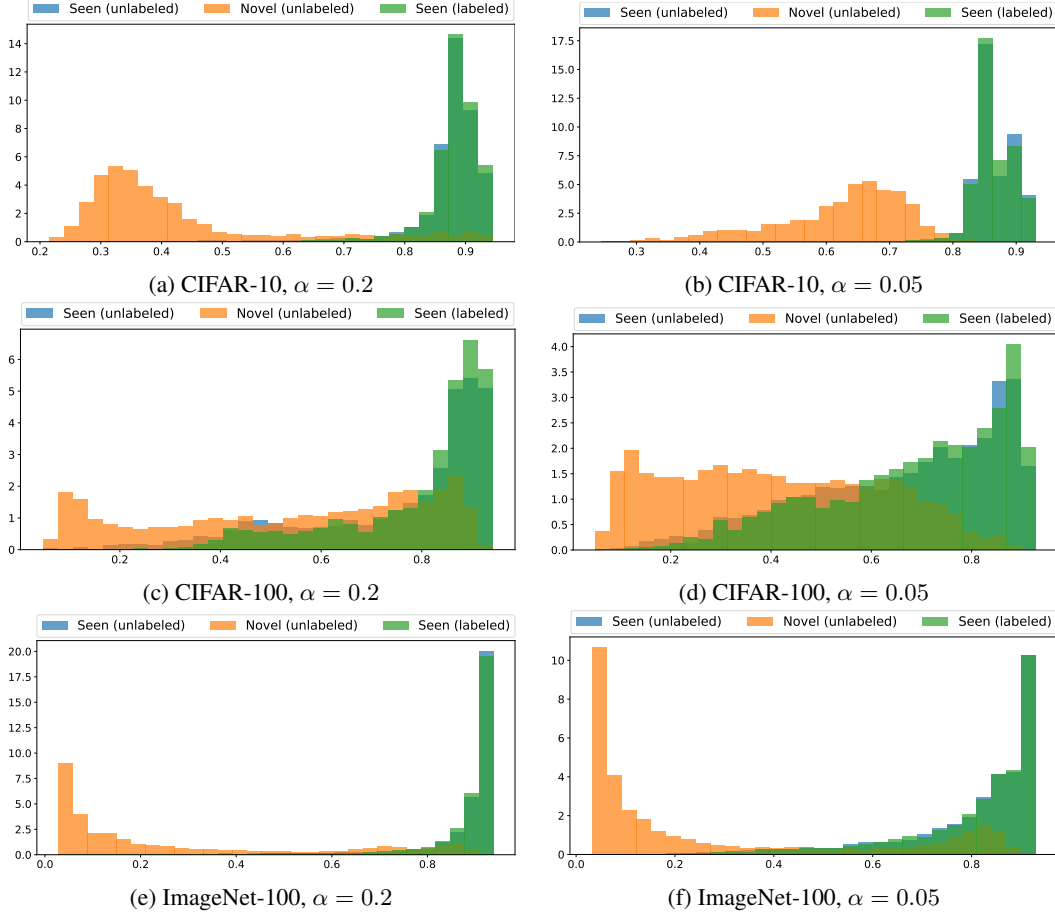


Figure 2: Normalized confidence score distributions (histogram) for labeled seen samples, unlabeled seen samples, and unlabeled novel samples from a single client on standard object recognition datasets under non-*i.i.d.* settings.

A.3 Training algorithm for FedLPA

The Fed-GCD training process in our FedLPA framework is outlined in Algorithm 1.

Algorithm 1: FedLPA

```

1: Input: Initial model  $\theta^0 = \{\phi^0, \psi^0\}$ , # of communication rounds  $T$ , # of local iterations  $M$ ,
   # of clients  $N$ , labeled data  $\{\mathcal{D}_1^l, \dots, \mathcal{D}_N^l\}$ , unlabeled data  $\{\mathcal{D}_1^u, \dots, \mathcal{D}_N^u\}$ 
2: Output: The final global model  $\theta^T$ 

3: Server executes:
4: Initialize  $\theta^0$ 
5: for each round  $t = 0, \dots, T - 1$  do
6:   Sample a subset of clients  $\mathcal{C}_t \subseteq \mathcal{C}$ 
7:   Server sends  $\theta^t$  to all active clients  $C_n \in \mathcal{C}_t$ 
8:   for each  $C_n \in \mathcal{C}_t$ , in parallel do
9:      $\theta_{n,M}^t \leftarrow \text{ClientUpdate}(C_n, \theta^t)$ 
10:  end for
11:   $\theta^{t+1} = \frac{1}{|\mathcal{C}_t|} \sum_{C_n \in \mathcal{C}_t} \theta_{n,M}^t$ 
12: end for
13: return  $\theta^t$ 

14: ClientUpdate( $C_n, \theta^t$ ):
15:  $\theta_{n,0}^t \leftarrow \theta^t$ 
16: if  $t \pmod R == 0$  then
17:   Build similarity graph  $\mathcal{G}_n^t$  using  $\mathcal{D}_n, \theta_{n,0}^t$ 
18:   Get concepts  $\mathcal{Y}_n^c$ , assignments  $\{c_j\}_{j \in \mathcal{D}_n^u}$ , prototypes  $\mathcal{M}_n^t$  (Section 3.3-3.4).
19:   Store  $\{c_j\}_{j \in \mathcal{D}_n^u}$  and  $\mathcal{M}_n^t$  for client  $C_n$ . Let  $K_n = |\mathcal{Y}_n^c|$ .
20: end if
21: for  $m = 0, \dots, M - 1$  do
22:   Sample  $B^l = (\mathbf{X}^l, \mathbf{Y})$  from  $\mathcal{D}_n^l$ , and  $B^u = \mathbf{X}^u$  from  $\mathcal{D}_n^u$ .
23:   Let  $\mathbf{c} = [c_1, \dots, c_{|B^u|}]^T$  be the stored cluster labels for the samples in  $B^u$ .
24:    $\pi_{n,B^u}[k] \leftarrow \frac{1}{|B^u|} \sum_{x_j \in B^u} \mathbb{I}(c_j = k), \quad \forall k \in \{1, \dots, K_n\}$ 
25:    $\mathcal{L}_{\text{LPA}}^u \leftarrow \mathcal{L}_{\text{LPA}}^u(\mathbf{X}^u, \pi_{n,B^u}, \mathcal{M}_n^t; \phi_{n,m}^t)$ 
26:    $\mathcal{L}_{\text{CE}}^l \leftarrow \mathcal{L}_{\text{CE}}^l(\mathbf{X}^l, \mathbf{Y}; \phi_{n,m}^t, \psi_{n,m}^t)$ 
27:    $\mathcal{L}_{\text{rep}}^l \leftarrow \mathcal{L}_{\text{rep}}^l(\mathbf{X}^l, \mathbf{Y}; \phi_{n,m}^t)$ 
28:    $\mathcal{L}_{\text{rep}}^u \leftarrow \mathcal{L}_{\text{rep}}^u(\text{concat}(\mathbf{X}^l, \mathbf{X}^u); \phi_{n,m}^t)$ 
29:    $\mathcal{L}_n \leftarrow \lambda(\mathcal{L}_{\text{LPA}}^u + \mathcal{L}_{\text{rep}}^u) + (1 - \lambda)(\mathcal{L}_{\text{CE}}^l + \mathcal{L}_{\text{rep}}^l)$ 
30:    $\theta_{n,m+1}^t \leftarrow \theta_{n,m}^t - \eta \nabla_{\theta_{n,m}^t} \mathcal{L}_n$ 
31: end for
32: return  $\theta_{n,M}^t$  to the server

```

B Additional Experimental Details

For training, the model undergoes a warm-up phase for 20 rounds. This is followed by 50 rounds of Fed-GCD training. Both stages employ stochastic gradient descent (SGD) with a momentum of 0.9 and a batch size of 128. The initial learning rate for both stages is set to 0.1. For the Fed-GCD training phase, the learning rate is decayed using a cosine annealing schedule. A weight decay of 5×10^{-5} is applied throughout the training. Note that for both stages, the optimizer is re-initialized at the beginning of each training round; thus, momentum statistics from previous local training epochs are not carried over. To ensure stable federated training, gradients are clipped at a norm of 10, following common practice in federated learning [1, 5, 4].

For evaluation, we distinguish between two variants of our approach: FedLPA and FedLPA+. For FedLPA, the similarity graph is constructed exclusively using unlabeled test data. In contrast, FedLPA+ constructs its similarity graph by leveraging both a labeled validation set and the unlabeled

test data. FedLPA+ incorporates our proposed CLCD mechanism for constructing the similarity graph. We adopt faiss [3] on both methods to accelerate the construction of their respective relation graphs.

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. 4
- [2] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022. 1
- [3] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 5
- [4] 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. 4
- [5] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, 2021. 4
- [6] Nan Pu, Wenjing Li, Xingyuan Ji, Yalan Qin, Nicu Sebe, and Zhun Zhong. Federated generalized category discovery. In *CVPR*, 2024. 1
- [7] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022. 1
- [8] Bingchen Zhao, Xin Wen, and Kai Han. Learning semi-supervised gaussian mixture models for generalized category discovery. In *CVPR*, 2023. 1