1. Data details

Fig. 5 illustrates the data difference between traditional classification and Generalized Category Discovery (GCD). Unlike traditional classification models trained in a closed set, where both training and test data only come from labeled data, GCD operates in an open set—a more realistic and challenging setting. In GCD, the training data includes unlabeled samples that consist of both known classes (*e.g.* dog and bird) and novel classes (*e.g.* penguin and horse) without annotations. During testing, the model should accurately classify the known class samples and recognize the novel class samples.

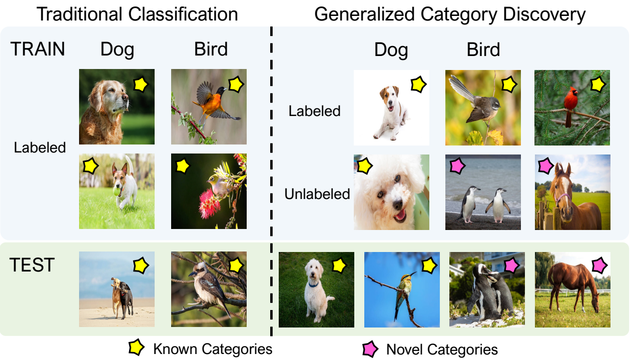


Figure 5. Data details of traditional classification and Generalized Category Discovery.

2. Training visualization

Fig. 6 shows the "Old" accuracy across training epochs for both SimGCD and our LegoGCD, employing the same random seed. Our method (depicted by green curves) consistently outperforms SimGCD (shown by orange curves) across diverse datasets. Notably, LegoGCD effectively addresses the catastrophic forgetting problem, particularly in fine-grained datasets like CUB and Stanford Cars, as well as in generic image recognition datasets CIFAR10/100 and ImageNet-100. Meanwhile, LegoGCD enhances known class accuracies, even in datasets with less pronounced forgetting, such as the unbalanced Herbarium 19. Additionally, improvements are observed in FGVA-Aircraft and ImageNet-1k datasets without forgetting.

3. Representations visualization

In this section, we employ t-distributed stochastic neighbor embedding (t-SNE) to visualize the learned representations of LegoGCD and compare them with the baseline SimGCD. The result of this comparison is presented in Fig. 7. Spectively, we randomly select 10 categories, each composed of 5 known and novel classes, with known and novel samples marked with ● and ✖, respectively. Fig. 7a and Fig. 7b display the visualizations on ImageNet-100 in SimGCD and

Algorithm 1 Pseudo code on one step for LegoGCD

```
#x1, x2: two view samples
#s_proj, s_pred, t_pred: projection feature, logits (similarities) for student and teacher
#mask: label mask
#x1_pred, x2_pred: logits of two view samples
def training_step(x1, x2):
    s_proj, s_pred = model([x1, x2])
    t_pred = s_pred.detach()
    #(1)Representation learning (unsupervised)
    unsup_con_loss = UnsupConLoss(s_proj)#Eq.(1)
    #(2) Representation learning (supervised)
    sup_con_loss = SupConLoss(s_proj, label=target[mask=1]) #Eq.(2)
    #(3) Supervised classification loss uses ground-truth labels on labeled data
    sup_loss = cross_entropy(s_pred[mask=1], target[mask=1]) #Eq.(5)
    #(4) Unsupervised (Self-distillation) classification loss on all data in Eq.(5)
    unsup_loss = cross_entropy(t_pred, s_pred)
    #(5) DKL
    x1_pred, x2_pred.detach() = s_pred.chunk(2)
    unsup_loss += DKL(x1_pred, x2_pred) #Eq.(11)
    #(6) LER in Eq.(10)
    loss_ler = LER(s_pred, s_pred+delta_logits)
    # Total representation learning loss
    loss_rep =  (1-lambda)*unsup_con_loss + sup_con_loss
    # Total classification loss
    loss_cls = (1-lambda)*unsup_loss + sup_loss
    # Overall loss
    loss = alpha * (loss_rep + loss_cls ) +beta* loss_ler #Eq.(13)
    return loss
```

our method, respectively. In Fig. 7a, some representations of novel classes are closer to known classes 24 and 48 than their truth labels, which are circled by red color. On the contrary, the representations of our method in known categories in Fig. 7b exhibit clear margins, indicating our method can more effectively distinguish known samples. Furthermore, Fig. 7c illustrates the logit distribution of known samples in unlabeled data. The predictions of our method exhibit higher logits, indicating enhanced sample discriminability.

4. Experimental supplements

In this section, we give detailed analyses of CIFAR10 and FGVC-Aircraft which improvements are not obvious in "Old" classes.

4.1. Results on CIFAR10

In this section, we conduct an ablation study on the confidence threshold in CIFAR10, as detailed in Tab. 8. Notably, the "Old" accuracy consistently surpasses that of SimGCD when $\delta = 0$. Despite a marginal drop in "New" accuracy ranging from 0.3 to 0.5, significant improvements are observed in "Old" accuracy, effectively mitigating the for-

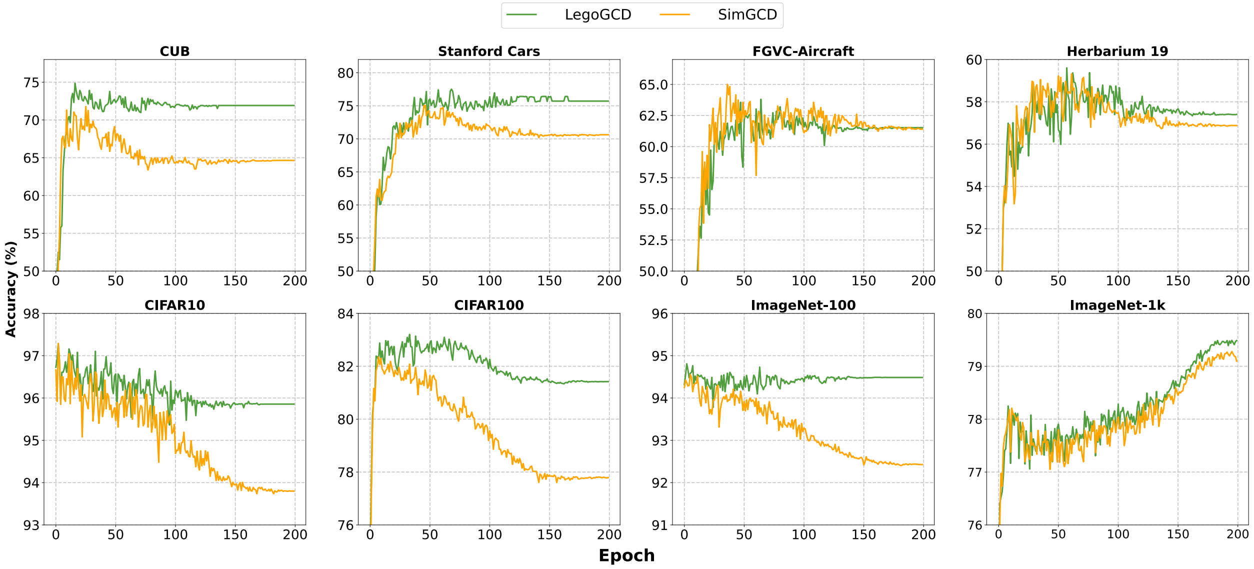
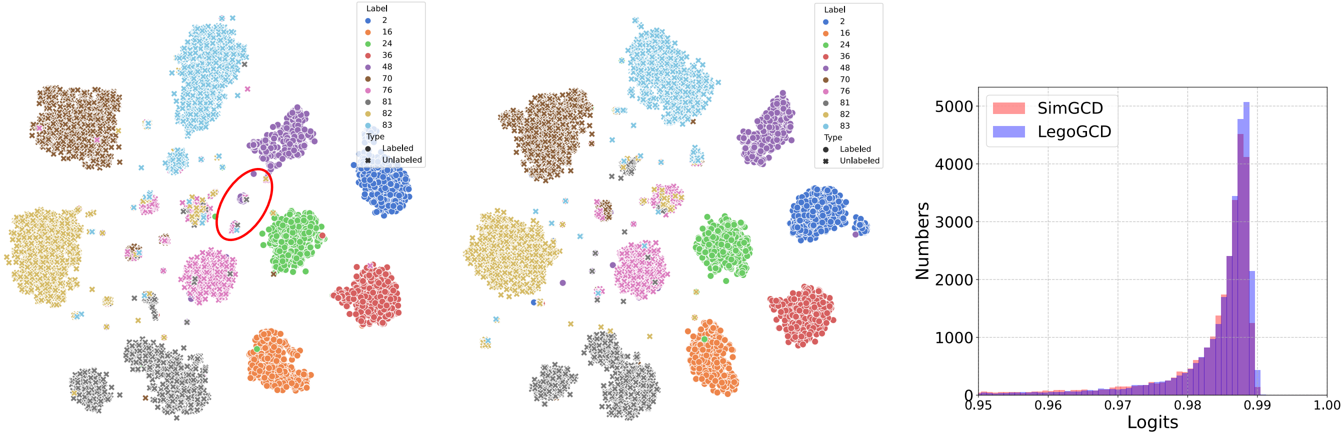


Figure 6. "Old" accuracy in each epoch compared between SimGCD and our LegoGCD. Our method (depicted in green) consistently outperforms SimGCD (shown in orange) across all datasets.



(a) ImageNet-100 on SimGCD

(b) ImageNet-100 on LegoGCD

(c) Logits distributions

Figure 7. The t-SNE visualization and logit distributions of the unlabeled dataset for SimGCD and LegoGCD ImageNet-100.

getting problem and show significant robustness in "Old" classes. Ultimately, we select $\delta = 0.97$ as the optimal threshold. While this choice results in a 0.3% reduction in "New" accuracy, it boosts "Old" accuracy by 1.9%, leading to an overall improvement of 0.6% in "All" accuracy.

4.2. Results on FGVC-Aircraft

In Tab. 9, we analyze the accuracy in the FGVC-Aircraft dataset under different settings for comprehensive comparisons. Initially, we use the same random seed=0 in both SimGCD and LegoGCD. Subsequently, we conduct 5 training runs across SimGCD and LegoGCD without a fixed random seed and average the results. As depicted in Tab. 9, when utilizing the same random seed=0, our method only

Table 8. Ablation study on confidence threshold δ was conducted on CIFAR10. The green indicates the margins ahead SimGCD (*i.e.* δ=0), while the red donates lagging values.

	CIFAR10		
δ	All	Old	New
0.0	96.9	93.8	98.5
0.85	97.5**+0.6**	96.4**+2.6**	98.0**-0.5**
0.90	97.4**+0.5**	96.0**+2.2**	98.1**-0.4**
0.95	97.1**+0.2**	94.3**+0.5**	98.5**+0.0**
0.97	97.5**+0.6**	95.7**+1.9**	98.2**-0.3**

Table 9. The accuracy of the FGVC-Aircraft dataset compared with SimGCD and LegoGCD in different settings.

	SimGCD			LegoGCD		
Seed	All	Old	New	All	Old	New
Yes	54.6	61.4	51.1	55.0	61.5+0.1	51.7+0.6
No	51.8	57.2	49.0	53.5	62.0	49.2
	52.5	58.3	49.6	54.6	60.0	51.9
	53.8	58.8	51.3	56.1	64.2	52.0
	55.2	61.8	51.9	55.8	61.3	53.0
	56.6	60.9	54.4	56.3	62.7	53.0
Avg.	54.0±1.75	59.4±1.67	51.2±1.94	55.2±1.18	62.0±1.57	51.8±1.57

slightly outperforms SimGCD by 0.1%, as shown in Tab. 2. However, our method achieves a substantial improvement of 2.6% in "Old" classes after 5 runs. Additionally, the standard deviation of our method is 1.57 while 1.67 in SimGCD, proving LegoGCD exhibits less fluctuation than SimGCD in the FGVC-Aircraft dataset.