MetaGCD: Learning to Continually Learn in Generalized Category Discovery
– Supplementary Metearial –

1 Overview

In this supplementary document, we provide the following details to support the paper:

- Comparing with directly using pre-trained backbone.
- Different number of sessions and data.
- Sensitivity on threshold.
- Fine-grained dataset.
- Distribution shift.

2 Knowing the number of classes *k* in advance

Estimating the number of novel classes in unlabeled data is quite challenging in most existing NCD methods. The work in [1, 2] is the only exception which proposes to estimate the number of novel classes by virtue of the labeled classes. But such methods cannot be applied in C-GCD since labeled data is unavailable in our continual scenario. Moreover, k is solely utilized to calculate the clustering accuracy of test data, without being involved in the training process. To lift the known prior, we employ an existing cluster number estimation algorithm, *Silhouette Coefficient*, to obtain the number of classes in each incremental session for both our MetaGCD and GM on CIFAR100:

Methods	**CIFAR100** (Session Number)												Overall Ave.		
	1			2			3			4					
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
GM	78.45	78.79	64.00	76.10	77.62	58.90	73.20	76.40	56.13	67.52	72.01	51.33	73.81	76.20	57.59
MetaGCD	79.07	79.15	67.80	76.77	78.34	62.20	74.93	77.67	60.93	74.11	76.79	61.40	76.22	77.98	63.08

Despite the accuracy on novel classes decreases under such circumstances, our results still outperform GM. Since the traditional estimation processes (e.g. *Silhouette* and *DBSCAN*) are computationally expensive, we leave C-GCD-oriented efficient estimation as our future work.

3 Comparing with directly using pre-trained backbone

To show the effectiveness of our training pipeline, we conduct experiment by directly using the pre-trained backbone (e.g. DINO [3]):

Methods	CIFAR100 (Session Number)												Overall		
	1			2			3			4			Ave.		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
DINO	54.39	54.54	52.00	56.89	57.23	54.20	54.88	55.17	53.33	54.20	53.87	55.50	55.09	55.20	53.76
MetaGCD	79.07	79.15	67.80	76.77	78.34	62.20	74.93	77.67	60.93	74.11	76.79	61.40	76.22	77.98	63.08

The significant decrease of accuracy reveals the necessity of re-training, as well as the effectiveness of proposed meta-learning and soft-neighbor-based contrastive learning.

4 Different number of sessions and data

To show how our method performs when the novel classes and session numbers do not match the one at meta-training stage. We change the number of novel classes at meta-test. Correspondingly the number of incremental sessions will also be changed. Left of Fig. 1 shows that our method is more robust to the session/class number changes.

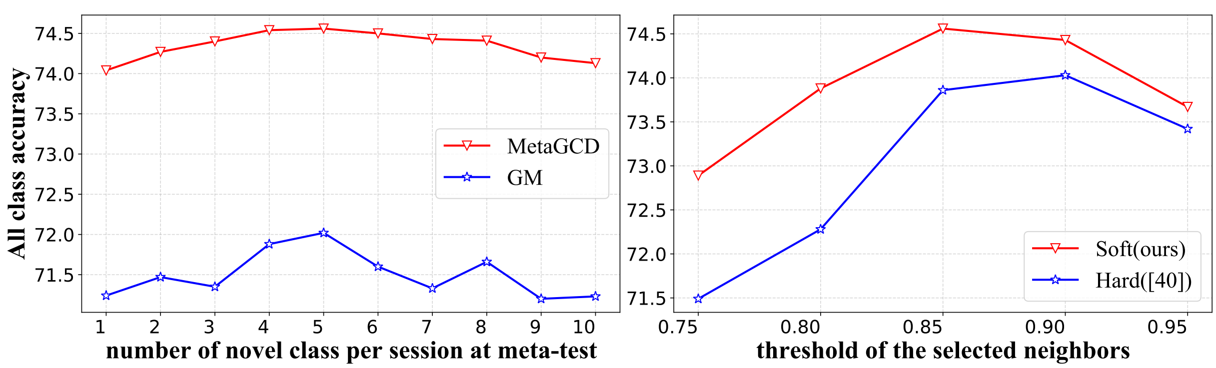

Figure 1: Left: various sess./class number. Right: soft vs hard threshold.

5 Sensitivity on threshold

Out method replies on the thresholding to select potential positive samples within each batch. Therefore, threshold value is an important factor. As shoun in Fig. 4, With a threshold of 0.75, our accuracy (72.91) surpasses the baseline (71.44 in Tab.3). Setting it too low will misclassify true negative samples to positive, suppressing the benefit of true positive. Moreover, mining the positiveness could be noisy due to the absence of true label, as mentioned in [4], which assigns hard pseudo-positives with a threshold of 0.95. The right of Fig.1 shows the comparison between soft and hard thresholds, demonstrating that our method is less sensitive.

6 Fine-grained dataset

We further conduct experiment on a fine-grained dataset: CUB200 [5]. We use 100 base classes and 5 incremental sessions with 20 novel classes each:

Methods	**CIFAR100** (Session Number)															Overall Ave.		
	1			2			3			4			5					
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
GM	63.99	67.95	63.10	62.28	66.47	61.85	60.41	66.75	54.97	55.65	63.79	47.11	51.33	65.29	54.07	58.73	66.05	54.22
MetaGCD	68.58	68.72	67.92	66.78	70.53	66.78	64.72	67.67	59.91	60.59	65.82	54.18	58.25	67.98	48.74	63.78	68.14	59.51

7 Distribution shift

To test the robustness of our method under domain shift: when meta-train and meta-test have different distribution of data, we conduct an experiment where we test a model trained on CIFAR100 using CUB200, as :

Methods	**CIFAR100** (Session Number)															Overall Ave.		
	1			2			3			4			5					
	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New	All	Old	New
GM	32.35	31.53	36.36	29.24	30.06	27.24	30.09	31.28	28.13	27.15	29.78	23.92	25.51	27.48	23.58	28.87	30.03	27.85
MetaGCD	44.85	44.03	48.89	44.41	44.34	44.59	40.67	41.48	39.35	42.26	44.69	39.26	37.57	40.82	34.40	41.95	43.07	41.30

We observed that although the performance drops, our method still outperforms GM.

[1] Han, Kai and Vedaldi, Andrea and Zisserman, Andrew. Learning to discover novel visual categories via deep transfer clustering In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[2] Vaze, Sagar and Han, Kai and Vedaldi, Andrea and Zisserman, Andrew. Generalized category discovery In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[3] Caron, Mathilde and Touvron, Hugo and Misra, Ishan and Jégou, Hervé and Mairal, Julien and Bojanowski, Piotr and Joulin, Armand. Emerging properties in self-supervised vision transformers In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[4] Zhong, Zhun and Fini, Enrico and Roy, Subhankar and Luo, Zhiming and Ricci, Elisa and Sebe, Nicu. Emerging properties in self-supervised vision transformers In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[5] Wah, Catherine and Branson, Steve and Welinder, Peter and Perona, Pietro and Belongie, Serge. The caltech-ucsd birds-200-2011 dataset In *California Institute of Technology*, 2011.