Contrastive Mean-Shift Learning for Generalized Category Discovery

Supplementary Material

7. Additional details

7.1. Uniform and Gaussian kernel on Mean-shift

We denote a uniform kernel and a Gaussian kernel as φ_u and φ_g, formulated as follows:

$$\varphi_u(x) = \begin{cases} 1 & \text{if } x \leqslant \delta \\ 0 & \text{if } x > \delta \end{cases} \tag{10}$$

$$\varphi_g(x) = e^{-\frac{x}{2\sigma^2}} \tag{11}$$

where δ indicates a distance threshold and σ indicates a standard deviation for determining the bandwidth of the kernel. In these way, the conventional mean-shift defines the neighborhood kernels with a fixed radius from the query data point where an arbitrary number of neighborhood data points can be included in $\mathcal{N}$. In contrast, our method sets no limit on the kernel radius yet always exploits a fixed number of neighborhood data points, which is realized with kNN search. For a fair comparison, we adopt cosine similarity for a distance metric same as our method by reformulating the Eqs. 10 and 11 as follows:

$$m(\boldsymbol{v}_i) = \frac{\sum_{\boldsymbol{v}_j \in \mathcal{N}(\boldsymbol{v}_i)} \varphi(cos(\boldsymbol{v}_j, \boldsymbol{v}_i))\boldsymbol{v}_j}{\sum_{\boldsymbol{v}_j \in \mathcal{N}(\boldsymbol{v}_i)} \varphi(cos(\boldsymbol{v}_j, \boldsymbol{v}_i))} \tag{12}$$

$$\varphi_u(x) = \begin{cases} 1 & \text{if } x \geqslant \delta \\ 0 & \text{if } x < \delta \end{cases} \tag{13}$$

$$\varphi_g(x) = e^{-\frac{1-x}{2\sigma^2}} \tag{14}$$

where $cos(\cdot)$ refers to cosine similarity.

Implementation details. We replace kNNs with distance-based NNs with the following implementation details. We empirically set $\delta = 0.9$, $\sigma = 0.1$, and limit the maximum number of retrieved embeddings to 1000 to prevent including too many points within the neighborhood. During evaluation, we modify the inference process by replacing the fixed kNN search with the uniform or Gaussian kernel as well. For a fair comparison, all the other hyperparamters remain the same as ours.

7.2. Details on benchmarks

Table 9 shows the number of labeled and unlabeled classes of each benchmark. In Table 10, we also denote the size of training data used for kNN retrieval.

	known classes	unknown classes
CIFAR100 [28]	80	20
ImageNet100 [39]	50	50
CUB-200-2011 [49]	100	100
Stanford-Cars [27]	98	98
FGVC-Aircraft [35]	50	50
Herbarium19 [45]	341	342

Table 9. Number of known and unknown classes.

	ImageNet100	CIFAR100	Stanford Cars	CUB
size	127115	50000	8144	5994

Table 10. Size of the kNN search space.

$\mathcal{L}_S$	CIFAR100			ImageNet100			CUB			Stanford Cars		
	All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel
z	80.1	**86.0**	68.2	84.6	95.5	79.1	65.9	**82.4**	62.6	40.1	62.4	29.4
v	**82.3**	85.7	**75.5**	**84.7**	**95.6**	**79.2**	**68.2**	76.5	**64.0**	**56.9**	**76.1**	**47.6**

Table 11. Ablation study on different embeddings on $\mathcal{L}_S$. The symbol z denotes mean-shifted vector, and v denotes the feature extracted from the backbone.

8. Additional experimental results

8.1. Design choices on supervised loss.

In Table 11, we replace the input of $\mathcal{L}_S$ with the mean-shifted embedding z as the same as in $\mathcal{L}_{CMS}$. The use of mean-shift in the supervised loss even harms the performance especially on fine-grained benchmarks. Since CMS learning is designed to incorporate the neighborhood collaboration of the query, integrating unlabeled (thus noisy) kNNs with the supervised loss turns out to be unreliable.

8.2. Effect of the number of nearest neighbors k

In Table 12, we examine the effect of the number of nearest neighbor k by varying the value with 4, 8, 16, 32, and 64. The result shows that retrieving 8NNs shows reasonable performance overall. The higher k value tends to more negatively affect on smaller-scale benchmarks such as CUB and Stanford Cars since more unrelated NNs are included due to the smaller size of the search space as outlined in Table 10. This also aligns with the experiment on section 7.1, in the sense that the different number of nearest neighbors can be interpreted as adopting a different bandwidth of a kernel. In other words, a large number of k leads to a sim-

k	CIFAR100			ImageNet100		
	All	Old	Novel	All	Old	Novel
4	81.7	85.4	74.3	83.2	95.5	77.0
8	**82.3**	85.7	**75.5**	**84.7**	95.6	**79.2**
16	80.4	86.5	68.1	83.7	95.6	77.7
32	79.4	**86.6**	65.1	83.1	**95.7**	76.7
64	77.1	84.8	61.7	83.8	**95.7**	77.8
k	Stanford Cars			CUB		
	All	Old	Novel	All	Old	Novel
4	66.2	73.6	62.5	48.7	70.7	38.1
8	**68.2**	**76.5**	**64.0**	**56.9**	**76.1**	**47.6**
16	64.0	74.8	58.5	50.5	71.5	40.3
32	58.9	75.6	50.5	50.6	72.7	39.9
64	49.4	61.9	43.1	45.1	72.2	32.0

Table 12. Ablation study with varying numbers of nearest neighbors k.

α	ImageNet100			CUB		
	All	Old	Novel	All	Old	Novel
0.5	**84.7**	**95.6**	**79.2**	**68.2**	**76.5**	**64.0**
0.6	83.4	95.7	77.3	63.7	74.3	58.4
0.7	83.5	95.7	77.4	64.3	75.7	58.6
0.8	82.4	95.7	75.6	62.7	73.8	57.2
1/(k+1)	82.1	93.9	76.2	58.7	68.4	53.9

Table 13. Ablation study on different choice of scaling hyperparameter α in Eq. 6

ilar outcome of using a larger kernel during the mean-shift clustering, which might over-smooth the embedding space. This indicates that retrieving an appropriate number of k is essential for stable learning.

8.3. Effect of the scaling hyperparameter α

We investigate the effect of the scale parameter α in Table 13. As α increases, kNN embeddings are assigned higher weights, which can be interpreted as increasing the uniformity of the mean-shift kernel. Unlike conventional mean-shift algorithms which are often sensitive to kernel parameters, our method demonstrates stable performance across different combinations of queries and kNN embeddings. The method tends to show better performance as α decreases, suggesting that approximating a Gaussian kernel is efficient when well-optimized for the target data distribution. This optimization is facilitated by the kNN-based kernel, which dynamically adjusts the kernel's bandwidth. Through this, we validate that adopting the mean-shift in a learnable manner leads to a consistent shifting even when a kernel is approximated to a discrete and non-continuous format of the kNN retrieval process.

λ	ImageNet100			CUB		
	All	Old	Novel	All	Old	Novel
0.25	84.7	95.5	79.3	66.8	74.9	62.8
0.35	**84.7**	**95.6**	**79.2**	**68.2**	**76.5**	**64.0**
0.5	84.4	95.9	78.6	65.3	74.4	60.8

Table 14. Ablation study on the weight of supervised contrastive loss λ.

8.4. Effect of varying the weight of supervised contrastive loss λ

In Table 13, we compare the performance with varying the weights of the supervised contrastive loss λ. The higher the weight is, the more contribution the labeled images from known classes in learning. Similar to the ablation studies on other hyerparameters, the performance is comparable on ImageNet100, while more sensitive on the fine-grained benchmark, e.g, CUB. Overall, using larger weight on the supervised loss tends to deteriorates the performance on unknown classes.

8.5. Results with a different backbone

We examine the generalizability of our method on different backbones by switching from DINO-ViT-B/16 to CLIP-ViT-B/16 [38]. For a fair comparison, we reproduce Vaze *et al*. [48] and PromptCAL [56] using CLIP by fine-tuning the last layer of CLIP-ViT-B/16 [38] and the projection head. To match the dimension between the backbone and the projection head, the input dimension of the projection head is changed from 768 to 512. All hyperparameters remain fixed, except for the learning rate of PromptCAL [56], which is adjusted from 0.1 to 0.01 for improved performance.

Evaluation results on GCD. As shown in Table 15, our method on GCD task outperforms in most cases by a large margin. Noticeably, the clustering accuracy measured *without* the ground-truth number of K shows comparable performance as well.

Evaluation results on inductive GCD. We further compare the result on inductive GCD setup as well. As shown in Table 16, our method achieves better accuracy overall, even *without* a given number of the ground-truth number K.

Estimated number of clusters. In Table 17, we report the number of estimated clusters of our method with CLIP backbone. The reported numbers correspond to the estimated cluster numbers in the experiment shown in Table 15. This validates the robustness of our method on discovering and estimating clusters with a different backbone.

Method	Known K	CIFAR100			ImageNet100			CUB			Stanford Cars			FGVC Aircraft			Herbarium 19		
		All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel
GCD [48]	✓	70.4	79.3	52.6	71.6	86.0	64.6	51.1	56.2	48.6	62.5	73.9	57.0	41.2	43.0	40.2	**39.7**	**58.0**	29.9
PromptCAL [56]	✓	69.4	77.3	53.5	75.2	87.0	69.3	53.7	61.4	49.9	60.1	77.9	51.5	42.2	48.4	39.0	37.4	50.6	**30.3**
Ours	✓	**80.3**	**85.2**	70.6	**85.9**	**93.8**	**82.0**	**65.8**	**75.3**	61.1	**77.9**	**89.0**	**72.6**	50.3	**59.1**	45.9	36.2	56.5	25.3
Ours		78.0	81.2	**71.5**	84.8	**93.8**	80.2	65.6	74.0	**61.3**	77.2	87.3	72.3	**50.6**	52.8	**49.5**	38.8	57.7	28.6

Table 15. Comparison of ours and the state of the arts on GCD with CLIP-ViT/B16, evaluated *with* or *without* the ground-truth class number K for clustering.

Method	Known K	CIFAR100			ImageNet100			CUB			Stanford Cars			FGVC Aircraft			Herbarium 19		
		All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel
PromptCAL [56]	✓	79.9	82.7	68.5	-	-	-	56.0	67.7	44.5	62.3	76.9	48.2	43.6	49.5	37.7	37.6	50.3	30.7
Ours	✓	**80.7**	83.9	**68.0**	**91.0**	**95.9**	**86.0**	**57.9**	**70.2**	45.6	**78.2**	**88.0**	**68.7**	**54.1**	**59.8**	**48.4**	**43.0**	51.1	**34.5**
Ours		80.5	**84.3**	65.2	84.7	95.8	73.6	56.1	64.9	**47.4**	75.0	87.2	64.2	53.8	59.7	47.9	40.4	**51.3**	28.9

Table 16. Comparison of ours and the state of the art on Inductive GCD with CLIP-ViT/B16, evaluated *with* or *without* the ground-truth class number K for clustering.

Method	CIFAR100			ImageNet100			CUB			Stanford Cars			FGVC Aircraft			Herbarium 19		
	GT	Pred	Err(%)	GT	Pred	Err(%)	GT	Pred	Err(%)	GT	Pred	Err(%)	GT	Pred	Err(%)	GT	Pred	Err(%)
Ours	100	94	6	100	103	3	200	193	3.5	196	189	3.6	100	77	23	683	443	35

Table 17. Estimated cluster numbers K and the error rates on the GCD task with CLIP-ViT/B16.

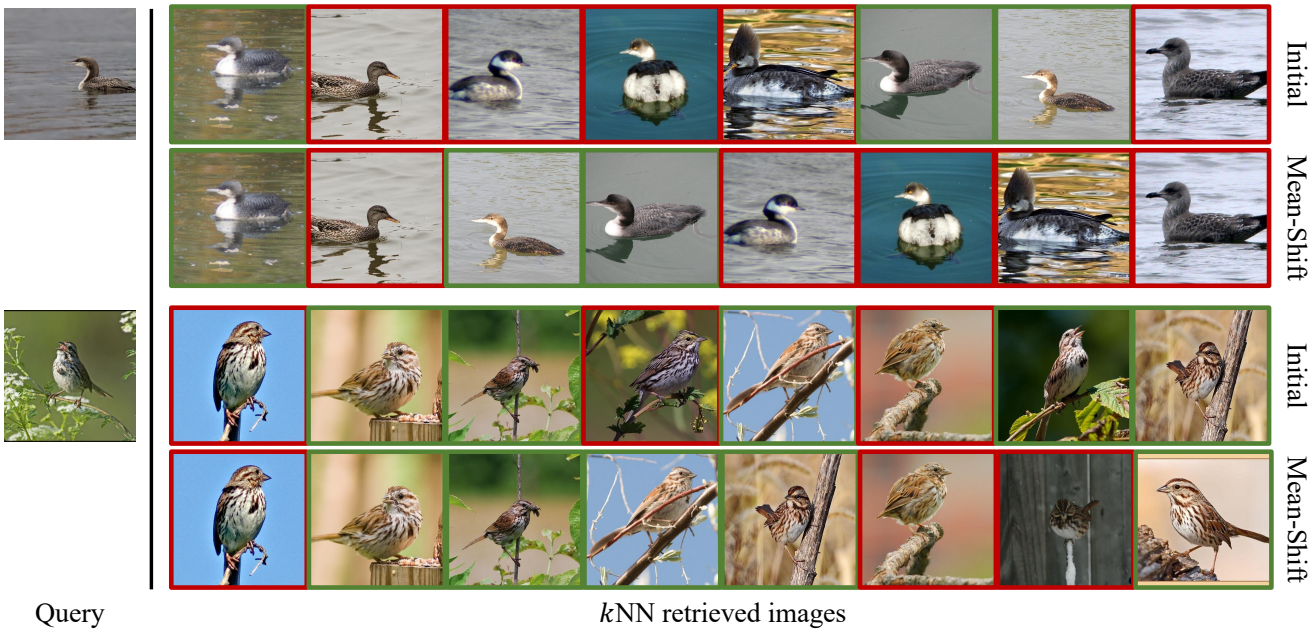
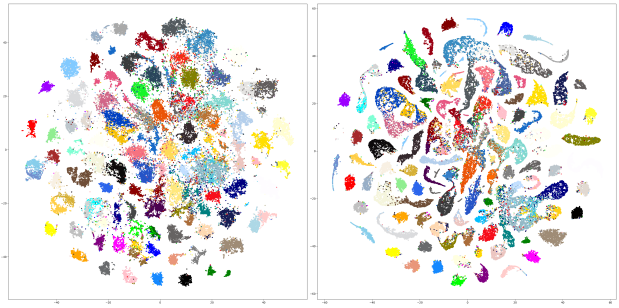


Figure 4. kNN retrieved images of the initial embedding $\boldsymbol{v}$ and mean-shift embedding $\boldsymbol{z}$ on CUB-200-2011. Green denotes the correct class and red an incorrect class.

9. Qualitative results

9.1. Qualitative results of the retrieved images.

In Figure 4, we present kNN retrieval results of our model after 1 iteration during inference. The rows denoted as 'Ini-



(a) DINO [5] (b) Ours

Figure 5. *t*SNE [33] visualization on ImageNet100. Each Color indicates a ground-truth class.

tial' refer to retrieval results using a feature extracted from a trained image encoder as a query. The rows denoted as 'Mean-Shift' indicate using a one-step mean-shift feature as a query. The retrieved words are ordered by their similarity scores starting from the top left. We can observe that applying the mean-shift on learned features enhances the grouping of the instances belonging to the same class, resulting in their retrieval at a higher rank than before.

9.2. tSNE visualization of the embedding space

In Figure 5, we visualize the embedding space of ImageNet100 before and after training. We observe that our method constructs clearer boundaries between clusters. Notably, the confusing classes are scattered around the center of the plots on the embedding space of DINO, where our method effectively clusters the confusing classes clearer than the DINO baseline once training converged.