PromptCAL: Contrastive Affinity Learning via Auxiliary Prompts for Generalized Novel Category Discovery – *Supplementary Material*

Appendix

In this appendix, we further provide detailed descriptions on the following contents:

- Additional details on our SemiAG method in Appendix A.
- Dataset profiles in Appendix B.
- The complete implementation details in Appendix C.
- Additional experimental results in Appendix D.
- Training algorithm of PromptCAL in Appendix E.
- Qualitative and visualization results in Appendix F.
- Efficiency analysis in Appendix G.
- Broader impact and limitations in Appendix H.
- License for experimental datasets in Appendix I.

A. Additional details on SemiAG

In this section, we present an extended description of TPG [18] affinity propagation algorithm that underlies our SemiAG method.

Suppose we have a graph $G = (V, \mathbf{E})$ with a node set V and an edge set $\mathbf{E}$. In our context, V is a set of N embeddings and $\mathbf{E} \in \mathbf{R}^{N \times N}$ represents the pairwise affinity matrix. TPG runs a graph diffusion process on a tensor product graph $\mathcal{G} = (V \times V, \mathcal{E})$ defined on G, where $\mathcal{E} = \mathbf{E} \otimes \mathbf{E}$ represents a 4-dim tensor. In particular, for $i, j, k, l = 1..., N$, the tensor element $\mathcal{E}_{i,j,k,l} = \mathbf{E}_{i,j}\mathbf{E}_{k,l} \in \mathbf{R}^{NN \times NN}$. In other words, the tensor graph $\mathcal{G}$ can be intuitively considered as a higher-order graph through cartesian product between G and itself. Then the graph diffusion process on $\mathcal{G}$ is formulated as:

$$\mathcal{E}^{(t)} = \sum_{i=0}^{t} \mathcal{E}^{i}$$

where $\mathcal{E}^{(t)}$ denotes the t-th step affinity matrix and $\mathcal{E}^{i}$ is i-power of $\mathcal{E}$. Theoretically, if the row-sum of $\mathcal{E}$ is less than one, $\mathcal{E}^{(t)}$ will converge to a nontrivial solution. To make computation tractable on large-scale data, TPG [18] proposes an iterative equation without multiplication on tensors which theoretically guarantees the same converged solution, which is formulated as:

$$\mathbf{Q}^{(t+1)} = \mathbf{E}\mathbf{Q}^{(t)}\mathbf{E}^{T} + \mathbf{I}$$

where $\mathbf{I}$ denotes an identity matrix, $\mathbf{E}$ is the affinity matrix, and $\mathbf{Q}^{(0)} = \mathbf{E}$.

In our work, we calibrates the affinity graph with only first-order structural information and, thus, set the diffusion step $\eta = 1$ since: firstly, online diffusion till convergence at each iteration will incur great computation overheads; besides, we find larger diffusion steps will include noisy false positives which significantly degrades the overall performance. Based on our further observation that the row-wise sum constraint has negligible effect on final performance, we exclude the row-wise sum threshold in TPG [18] as another hyperparameter.

B. Dataset details

We evaluate PromptCAL on six benchmarks, *i.e.*, CIFAR-10 [9], CIFAR-100 [9], ImageNet-100 [10], CUB-200 [17], StandfordCars [8], and Aircraft [11]. The profile of six benchmark datasets is displayed in Table 1. Our dataset splits follow GCD [16].

Dataset	CIFAR-10	CIFAR-100	ImageNet-100	CUB-200	Aircraft	StanfordCars
#Images in $\mathcal{D}$	50k	50k	127.2k	6k	6.6k	8.1k
#Classes ($\lvert\mathcal{C}\rvert$)	10	100	100	200	100	196
#Known Classes ($\lvert\mathcal{C}_{kwn}\rvert$)	5	80	50	100	50	98

Table 1. **The dataset profiles of six benchmarks for evaluation.**

C. Implementation details

Architecture and optimization. Following [16], we use a 12-layer base vision transformer [13] with a patch size of 16 (ViT-B/16) as our backbone in all experiments. The backbone weights are initialized with pre-trained DINO [3] on the ImageNet-1K [10] dataset. The first 11 blocks of the backbone are frozen as in [16]. For our PromptCAL, we further adapt pre-trained ViT [13] with prompts by prepending 5 prompts before each block (in VPT-Deep scheme [7]). We only supervise the first 2 of 5 prompts at the last block with DPR loss, and all remaining prompts are unsupervised and thus automatically learned. In practice, this ViT backbone can be of any architecture and pre-trained with any self-supervised learning method on large-scale datasets. Initially, we separately adopt two DINO [3] projection heads for `[CLS]` and `[P]` to avoid negative interferences, which are randomly initialized. In both stages, we fix the batch size to 128 on all datasets; besides, we optimize PromptCAL with standard SGD with a momentum of 0.9, a weight decay of 5×10^{-5}, and an initial learning rate of 0.1. For all datasets, we train PromptCAL with 200 epochs in the first stage; in the second stage, we train PromptCAL with 70 epochs on CIFAR-10/100 and ImageNet-100 datasetes; while, we optimize PromptCAL by 100 epochs on CUB-200, StanfordCars, and Aircraft datasets.

Warmup training. In the 1^{st} stage training of PromptCAL, we adopt an unsupervised L_2 distillation loss on ImageNet-

1K [10] with a loss weight of $\min\big(0, 0.5\times(1-\frac{E}{5})\big)$. Here, E denotes the epoch number. We add this loss based on consideration of potential adverse effects of randomly initialized visual prompts on the class token.

Contrastive affinity learning. In the 2^{nd} stage training of PromptCAL, model parameters (prompt-adapted backbone with two heads) are initialized by the best warmed-up checkpoint at the 1^{st} stage. For SemiAG parameters, we fix the neighborhood size $K = |\mathcal{M}|/(4|\mathcal{C}|)$ for all datasets unless otherwise specified. We fix sizes of both memories as $|\mathcal{M}| = |\mathcal{M}_{\text{P}}| = 4096$ and set $N_{neg} = 1024$ in all experiments. Furthermore, since most edges of the binarized affinity graph $\mathbf{G}'_b$ are of small values, we first compute the mean value of non-zero affinities; then, we fix threshold q to 80% quantile of affinities above this value for all fine-grained datasets, and 50% for all generic datasets. We fix diffusion step $\eta = 1$. For loss parameters, we fix $\alpha = 0.35$, $\tau = 1.0$, and $\tau_a = 0.07$ based on existing literature [3,6,16]. Besides, we determine $\gamma = 0.35$ and $\beta = 0.6$ via first and second stage validation scores on the held-out validation set. Our teacher model is initialized by the student weights at the beginning, and we conduct momentum updates with a momentum of 0.999 at each iteration. During the inference, the `[CLS]` representation of the student model is used for prediction.

Validation scheme. Follow GCD [16] setup, we assume access to a small validation set, in which only samples from known classes are labeled. In the first stage, we keep the best checkpoint with the highest clustering accuracy on `Known` on the validation set. In the second stage, we keep the best checkpoint with the highest clustering quality on the validation set for evaluation. We define clustering quality as the average score of the clustering accuracy on `Known` classes and unsupervised Silhouette score [12] on `New`. Note that there is no information leakage, since Silhouette score does not need ground-truth label.

Other baselines. For GCD [16], since our dataset splits are consistent with theirs, we report their official scores for main comparisons. In our ablations, we reproduce its results based on their official codes. For ORCA [2], we adapt their backbone from ResNet to the same pre-trained DINO and obtain results based on their official codes. For our baseline (PromptCAL w/o prompt), we remove all the prompts and DPR loss on them; besides, we keep the warmup training stage for fair comparison. Other parameters follow the standard setups.

D. Additional experiment results

D.1. Inductive category discovery

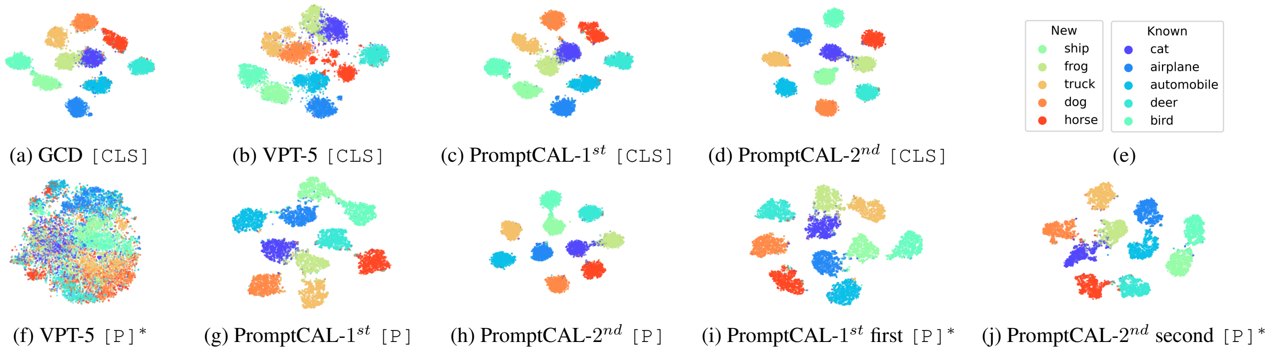
In contrast to the evaluation protocol on transductive category discovery GCD [16], we also conduct ablation experiments on inductive category discovery protocol proposed in ORCA [2]. In other words, besides achieving high performance on category discovery on the unlabeled training data (transductive protocol), we also expect models to learn general rules applied to unseen test sets (inductive protocol). Therefore, we conduct experiment under this inductive evaluation protocol on three benchmarks (CUB-200 [17], CIFAR-100 [9], and ImageNet-100 [10] datasets). In this experiment, we hold out 10% (labeled and unlabeled) training data as the validation set for GCD and PromptCAL. From displayed results in Table 7, we can conclude that our PromptCAL achieves the best performance on three datasets, which manifests its good generalization capability. Meanwhile, we observe that PromptCAL boosts performance on `New` with significant margins.

D.2. Additional ablation on SemiAG and DPR

To further validate the effectiveness of our SemiAG, we conduct ablation on different positive mining methods integrated into our online contrastive learning framework with CAL. Besides, we supplement more ablation results on larger datasets (*i.e*., CIFAR-100 and ImageNet-100 datasets) to showcase that learning with semantically discriminative prompts can achieve notable improvements across various datasets. The experiment results are presented in Table 3. Firstly, we notice that SemiAG significantly outperforms other positive mining methods, *i.e*., naive KNN with SemiPriori (KNN w/ S.P.) and Ranking Statistics (R.S.) [5]. The results unveil that both KNN with SemiPriori and RankingStats fail to reliably uncover the substantial semantic information in embedding spaces, which proves that our SemiAG method is the most effective in this open-set setting. On the other hand, removing either DPR loss or entire prompt-related components in PromptCAL causes noticeable performance drop, *e.g*., nearly 3% and 2% drops on `All` on CIFAR-100 dataset after removing prompts and DPR loss. Moreover, removing either component also leads to severe overfitting on `Known` classes.

D.3. Visualization on embeddings

To inspect the learned semantic discriminativeness of PromptCAL, we visualize embeddings by t-SNE [15] algorithm in Fig. 2. Firstly, by comparing (a-d), we can conclude that PromptCAL can effectively learn better semantic clustering, witnessed by higher purity, larger interclass separation, and high compactness. Notice in (b) that naive VPT model suffer from degraded clustering performance compared with (a) baseline, which again proves that lack of semantic supervision is a critical issue (see ablations in main content) in prompt tuning. Interestingly, though not supervised, automatically learned prompts `[P]`* in (i) and (j) can still learn robust semantically meaningful representation, benefiting from DPR on `[P]`. Meanwhile, DPR loss reinforce this effect in (g) and (h). Furthermore, we



(a) GCD `[CLS]` (b) VPT-5 `[CLS]` (c) PromptCAL-1^{st} `[CLS]` (d) PromptCAL-2^{nd} `[CLS]` (e)

(f) VPT-5 `[P]`* (g) PromptCAL-1^{st} `[P]` (h) PromptCAL-2^{nd} `[P]` (i) PromptCAL-1^{st} first `[P]`* (j) PromptCAL-2^{nd} second `[P]`*

Table 2. **The t-SNE [15] visualization of ViT embeddings on CIFAR-10 test set** for GCD [16], naive VPT model [7], and PromptCAL-1^{st} stage and 2^{nd} stage, Here, `[CLS]`, `[P]`, and `[P]`* denote embeddings from ViT class token, ensembled prompts supervised by DPR loss, and unsupervised prompts, respectively. The embedding clustering shows that DPR reinforces the semantic discriminativeness of `[P]`, and for `[P]`* despite no explicit supervision. (e) exhibits the class name each color denotes. All figures share the same axis scale.

also observe that `[P]` supervised by CAL loss (h) can learn better semantic clustering than those supervised by SemiCL (g), and better benefit `[P]`* (j). Thanks to better semantic information supplied by CAL loss, `[CLS]` of PromptCAL-2^{nd} learns more compact and better-separated clusters compared with that of PromptCAL-1^{st}. To summarize the above, we can conclude that the second stage enhances the prompts potential using CAL loss, which further enables prompts and CAL to synergistically improve the overall performance.

D.4. Sensitivity analysis on hyper-parameters.

We conduct ablation experiments on critical hyper-parameters of PromptCAL, which includes: `(1)` CAL loss weight β; `(2)` neighborhood size K; `(3)` different pre-training methods; `(4)` number of auxiliary prompts.

CAL loss weight. We sample β values from 0.2 to 1.0 at an interval of 0.2 and run experiments on StanfordCars dataset. The results are visualized in Fig. 1. We observe that decreased weights of contrastive affinity learning will cause model suffer from low performance on `New`. We argue that, although different datasets exhibit different trends, the model performance is fairly robust within the modest value range (from 0.4 to 0.8).

Neighborhood size. We select $K = 5, 10, 15, 20$ for ablations on two datasets (CIFAR-100 and Aircraft, both with 100 `All` classes). Results in Table 4 display that PromptCAL is robust to small K; while, its performance degrades largely as the neighborhood expands. We guess it is because false positive has severer negative effects than false negatives.

Pretraining. We argue that PromptCAL can take advantage of the property of the high KNN precision of ViT, which are pre-trained in various schemes. In Table 6, we replace DINO [3] pre-trained ViT with iBoT [19] pre-trained ViT as our backbone in CIFAR-100 experiments [1]. We can show that PromptCAL further improves as iBoT possesses higher KNN precision [19]. It manifests that our PromptCAL performance is likely to correlate with better initial representations.

Dataset	Setup	All	Known	New
CUB-200	w/o prompt	60.3	64.8	58.0
CUB-200	w/o DPR	59.3	63.3	57.4
CUB-200	KNN w/ S.P.	60.1	**70.1**	55.1
CUB-200	R.S.	55.6	66.0	50.3
CUB-200	PromptCAL	**62.9**	64.4	**62.1**
CIFAR-100	w/o prompt	78.1	83.0	68.4
CIFAR-100	w/o DPR	79.0	83.4	70.3
CIFAR-100	KNN w/ S.P.	78.7	85.3	65.4
CIFAR-100	R.S.	75.9	**87.1**	53.4
CIFAR-100	PromptCAL	**81.2**	84.2	**75.3**
ImageNet-100	w/o prompt	81.8	94.7	75.3
ImageNet-100	w/o DPR	80.7	94.8	73.6
ImageNet-100	KNN w/ S.P.	81.9	95.0	75.3
ImageNet-100	R.S.	78.1	**95.2**	69.4
ImageNet-100	PromptCAL	**83.1**	92.7	**78.3**

Table 3. **Further ablation study on CUB-200 [17], CIFAR-100 [9], and ImageNet-100 [10] datasets.** We investigate four setups: the first is PromptCAL removing all prompt related components; the second is PromptCAL without DPR loss; the third is replacing SemiAG with naive KNN incorporated with SemiPriori; the last one is replacing our SemiAG with RankingStats [5] pseudo labeling.

Number of supervised prompts. We varies the number of supervised prompts to observe sensitivity of performance

[1] The KNN precision of DINO and iBoT on ImageNet-1K dataset are 76.1% and 77.1%, respectively [19].

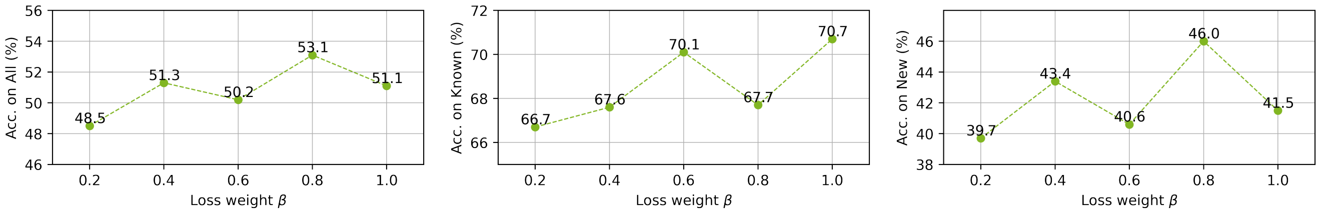


Figure 1. **Ablation study on the CAL loss weight β on StanfordCars [8] dataset.**

	CIFAR-100			Aircraft		
K	All	Known	New	All	Known	New
5	80.9	**85.5**	71.7	49.0	54.4	46.3
10	**81.2**	84.2	75.3	**52.2**	52.2	**52.3**
15	80.2	83.4	74.0	50.6	**55.1**	48.4
20	78.9	80.3	**76.1**	47.4	52.5	45.0

Table 4. **Ablation study on the neighborhood size K on the CIFAR-100 [9] and Aircraft [11] datasets.**

w.r.t. this parameter. Table 8 showcases the results under different setups. We can observe that leaving some unsupervised prompt to learn can provide extra flexibility to the backbone and thus achieves the best performance, especially on `New`. In general, PromptCAL is robust to different numbers of supervised prompts.

D.5. Additional results on Herbarium dataset

We also present evaluation results on the challenging Herbarium2019 [14] dataset, which consists of 683 classes and 34k images in total. Our dataset split follows [16]. Specifically, we set labeling ratio to 50% and known class number to 341. We compare PromptCAL with other SOTAs on this dataset. Considering larger class numbers, we enlarge the memory size to 2×10^4 and $N_{\text{neg}} = 5000$, accordingly. We set $K = |\mathcal{M}|/(4|\mathcal{C}|) \approx 7$ in this case. Other parameters follow the setup on fine-grained datasets. Table 5 display the results, which demonstrates our PromptCAL also excels at discovering categories on large vocabulary fine-grained datasets, especially on `New` classes.

E. Training algorithm of PromptCAL

Given a training dataset $\mathcal{D}$, we describe our entire training algorithm of PromptCAL in Algo. 1. Before PromptCAL training, we adapt the ImageNet pre-trained ViT backbone $f(\cdot|\theta)$ with prompts into $f(\cdot|\theta, \theta_{\text{P}})$, and randomly initialize two identity heads $g(\cdot|\theta_{\text{H}})$ and $g_{\text{P}}(\cdot|\theta_{\text{P,H}})$ for `[CLS]` and `[P]`, respectively.

In the 1^{st} stage, we sample a batch of images $\mathbf{X}$ with their corresponding labels $\mathbf{Y}$ at each iteration. Note that ground-truth labels of unlabeled images are masked in $\mathbf{Y}$.

Method	All	Known	New
KMeans [1]	12.9	12.9	12.8
RankStats+ [5]	27.9	**55.8**	12.8
UNO+ [4]	28.3	53.7	14.7
GCD [16]	35.4	51.0	27.0
ORCA [2]	25.5	34.7	15.8
PromptCAL (our)	**37.0**	52.0	**28.9**

Table 5. **Additional experiments on the Herbarium2019 [14] dataset.**

Method	All	Known	New
GCD [16]	73.0	76.2	66.5
PromptCAL (iBoT [19])	**83.0**	**85.0**	**78.9**
PromptCAL (DINO [3])	81.2	84.2	75.3

Table 6. **Ablation study on pretraining methods on CIFAR-100 [9] dataset.**

We obtain `[CLS]` and `[P]` projected features $(\mathbf{Z}, \overline{\mathbf{Z}}_{\text{P}})$ by forwarding $\mathbf{X}$ through backbone and two heads. Next, we compute SemiCL loss (Eq. 2) on the features based on the class labels and label-or-not information in $\mathbf{Y}$. All tunable parameters $(\theta, \theta_{\text{P}}, \theta_{\text{H}}, \theta_{\text{P,H}})$ are updated.

Before the 2^{nd} stage training, we initialize two empty embedding memory bank $\mathcal{M}, \mathcal{M}_{\text{P}}$ for `[CLS]` and `[P]`, respectively. Besides, we initialize the teacher model with the student weights. During the training, for each sampled batch ($\mathbf{X}$, $\mathbf{Y}$), we first obtain student embeddings of `[CLS]` and ensembled `[P]` $(\mathbf{H}, \overline{\mathbf{H}}_{\text{P}})$, and corresponding student features ($\mathbf{Z}$, $\overline{\mathbf{Z}}_{\text{P}}$) by forwarding images to the student. Meanwhile, we acquire the teacher embeddings and features ($\mathbf{H}_T$, $\overline{\mathbf{H}}_{\text{P},T}$, $\mathbf{Z}_T$, $\overline{\mathbf{Z}}_{\text{P},T}$) from the teacher, correspondingly.

Further, we construct a sub-graph for a token (line 14 for the class token and line 18 for ensembled prompts) based on its teacher embeddings of the current batch and all embeddings in its corresponding memory. Given the sub-graph, we sequentially perform three operations of SemiAG to obtain the calibrated binarized affinity graph (line 15 and 19). For each student embedding, we utilize its teacher embed-

Method	CUB-200					CIFAR-100					ImageNet-100				
	All	**Known**	**New**	**Known***	**New***	**All**	**Known**	**New**	**Known***	**New***	**All**	**Known**	**New**	**Known***	**New***
GCD [16]	57.5	64.5	50.6	69.2	57.6	70.1	76.8	43.5	78.7	58.2	79.7	92.7	66.7	92.7	66.9
ORCA (DINO) [2]	40.7	61.2	20.2	**76.3**	38.3	77.7	83.6	53.9	83.6	66.6	81.3	**94.5**	68.0	**94.5**	71.1
PromptCAL (our)	**62.4**	**68.1**	**56.8**	70.1	**60.1**	**81.6**	**85.3**	**66.9**	**86.2**	**71.3**	**84.8**	94.4	**75.2**	94.4	**75.3**

Table 7. **Evaluation in the inductive GCD setting [2] on three benchmarks.** The results are reported in accuracy scores on the test set. Here, we also adopt the task-informed evaluation protocol in [2, 4], *i.e.*, Known* and New* are evaluated by separate clustering and Hungarian assignment.

Method	Stage 1			Stage 2		
	All	**Known**	**New**	**All**	**Known**	**New**
GCD [16]	51.3	56.6	48.7	-	-	-
DPR-2-5	51.1	55.4	**48.9**	**62.9**	**64.4**	**62.1**
DPR-1-5	**51.7**	**57.2**	**48.9**	59.9	63.0	58.4
DPR-5-5	50.9	55.6	48.6	61.0	63.6	59.8

Table 8. **Ablation study on prompt numbers of our prompt-adapted ViT backbone.** Evaluation conducted on CUB-200 [17] dataset.

ding counterpart as a query on the affinity graph to acquire its pseudo positive set and pseudo anchor set with randomly sampled pseudo negatives (line 16 and 20). With these pseudo positive and anchor sets, we compute CAL loss on embeddings of each token (line 17 and 21) by Eq. 7.

Along with CAL loss, we also compute SemiCL loss on the projected features; here, we utilize student embeddings as queries and teacher embeddings as keys in the contrastive loss (Eq. 8 and Eq. 9). In other words, for each student embedding, we construct its positive and anchor sets with teacher embeddings and then compute the semi-supervised contrastive loss. Next, we obtain the total loss for the [CLS] token by combining its SemiCL and CAL loss functions (Eq. 9). After adding our DPR counterpart loss on ensembled prompts, we finally get the total loss at this stage (Eq. 10).

At each iteration, all tunable parameters of the student are updated. Lastly, we update two memories with teacher embeddings of their corresponding token and update momentum teacher model with the updated student model. Note that for inference, we adopt embeddings from the [CLS] token of the student model $f(\cdot|\theta, \theta_\mathrm{P})$ for final predictions.

F. Qualitative results

In this section, we present qualitative results of categorization confusion matrix, attention map visualization, and KNN retrieval.

Confusion matrix on ImageNet-100. We present confusion matrix for GCD [16] and our PromptCAL on both Known and New classes on ImageNet-1K dataset in Fig. 2. We can observe that our PromptCAL can learn more robust clusters on New classes, while preserving high accuracy on Known. Moreover, our PromptCAL is less susceptible to confusion between Known and New.

Attention map visualization. We visualize and compare the attention maps of [CLS] tokens of DINO [3], GCD [16], PromptCAL-1st, and PromptCAL-2nd in Fig. 3. We summarize the following observations: (1) DINO attends to the instance discriminative regions, *e.g.*, licence plate, and may overfit on surrounding objects; while, PromptCAL lays more attention on class-specific features, *e.g.*, car lights for cars, and feather textures for birds. (2) Although both GCD and PromptCAL can attend to semantically meaningful regions, PromptCAL-2nd focuses on multiple semantically discriminative regions, *e.g.*, car lights and textures, feathers and wings. (3) After CAL training, attention maps of PromptCAL-2nd in contrast to that of PromptCAL-1st are remarkably refined.

Nearest-neighbor query. In Fig. 4, we visualize the 8 predicted nearest neighbors, from GCD [16] and our PromptCAL, of 20 randomly selected query images, which are labeled with correct (green) and incorrect (red). Specifically, we first randomly sample a subset from ImageNet-1K, and conduct KNN search (with cosine distance) for given random queries in [CLS] embedding space. We can observe that PromptCAL generally exhibits higher retrieval precision (*e.g.*, for "n02006656" in 3rd row, "02018207" in 5th row, "n02027492" in 8th row). To summarize, our PromptCAL learns more semantically calibrated local structures. We also notice that both GCD and PromptCAL fails on "n01695060" in 11th row, which, we guess, is due to the confusing view angle of the query image and high visual similarities between lizards of different species.

G. Efficiency analysis

Compared with the raw ViT backbone (GCD [16]), our PromptCAL only adds negligible computation overheads during inference, since the only overheads origin from visual prompts. In Table 9, we quantitatively list inference time per image, thoughput, and FLOPs for PromptCAL. It can be observed that our PromptCAL achieves comparable inference efficiency with the raw ViT backbone.

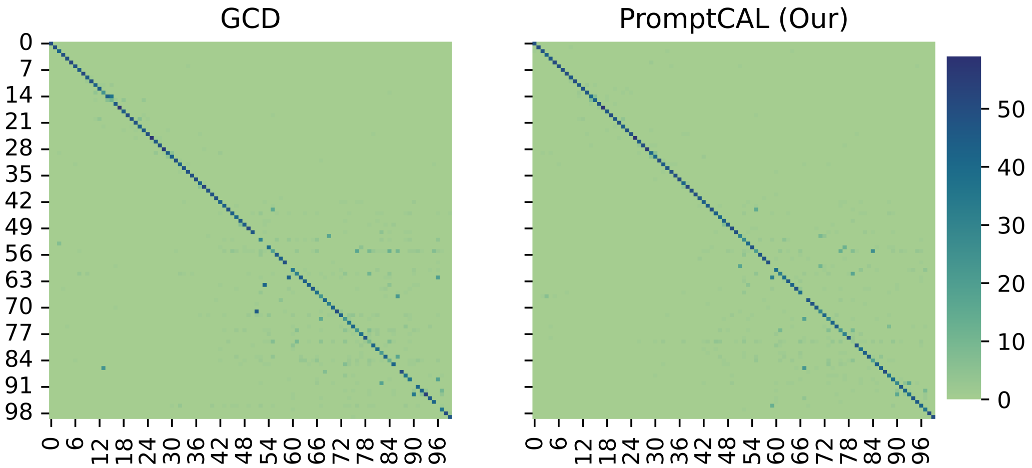


Figure 2. **Confusion matrix of PromptCAL on ImageNet-100 [10] test set.** The labels on the x-axis and y-axis denotes the class index of our generated split. The first 50 classes are `Known`, and the last 50 classes are `New`.

Method	Time (s/per img)	Throughput (img/s)	FLOPs
GCD [16]	1.70×10^{-3}	586	35.1
PromptCAL (our)	1.79×10^{-3}	558	36.1

Table 9. **Comparison on inference time, throughput, and FLOPs based on ViT-B/16 backbone.**

H. Broader impact and limitations

It should be noticed that although our method achieves state-of-the-art performance on generalized novel category discovery problem, the performance gap between the fully supervised counterpart and our method still exists. Besides, in real world, the data can be more complicated and uncurated. For instance, realistic data may follow long-tail distributions, human-annotation may incur noises, and the vocabulary maybe huge. We leave these for future research.

I. License for experimental datasets

All datasets used in our experiments are permitted for research use. CIFAR-100 and CIFAR-10 [9] are released under MIT license for research use. ImageNet-100, the subset of ImageNet [10], also allows for research purpose. Besides, CUB-200 [17], Aircraft [11], StanfordCars [8] also permits for research purpose. Herbarium19 [14] are released for non-commercial purposes.

Algorithm 1: PromptCAL training algorithm.

```
Input: Training dataset $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_l$, an ImageNet pre-trained ViT backbone $f(\cdot|\theta)$, and a randomly-initialized [CLS]
       projection head $g(\cdot|\theta_{\text{H}})$.
Output: Trained prompt-adapted model $f(\cdot|\theta, \theta_{\text{P}})$.
Initialize prompt-adapted backbone with random prompts into $f(\cdot|\theta, \theta_{\text{P}})$.
Randomly initialize prompt projection head $g_{\text{P}}(\cdot|\theta_{\text{P,H}})$ from $g$.
/* Stage 1: Warm-up Training                                                        */
for each epoch $e$=1...$E_1$ do
    for each batch $(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}$ do
        $\mathbf{Z}, \overline{\mathbf{Z}}_{\text{P}} = \text{Forward}(\mathbf{X}, f, g, g_{\text{P}})$ // forward backbone and heads
        Compute overall SemiCL loss $L_1$ by Eq. (2) on $\mathbf{Z}, \overline{\mathbf{Z}}_{\text{P}}$.
        Back-propagation and optimize $\theta, \theta_{\text{P}}, \theta_{\text{H}}, \theta_{\text{P,H}}$.
/* Stage 2: Contrastive Affinity Learning                                           */
Initialize memory $\mathcal{M}, \mathcal{M}_{\text{P}}$.
Initialize teacher $f_T, g_T, g_{\text{P},T}$ from the student model.
for each epoch $e$=1...$E_2$ do
    for each batch $(\mathbf{X}, \mathbf{Y}) \in \mathcal{D}$ do
        /* Forward                                                                      */
        $\mathbf{H}, \overline{\mathbf{H}}_{\text{P}}, \mathbf{Z}, \overline{\mathbf{Z}}_{\text{P}} = \text{Forward}(\mathbf{X}, f, g, g_{\text{P}})$ // forward student
        $\mathbf{H}_T, \overline{\mathbf{H}}_{\text{P},T}, \mathbf{Z}_T, \overline{\mathbf{Z}}_{\text{P},T} = \text{Forward}(\mathbf{X}, f_T, g_T, g_{\text{P},T})$ // forward teacher
        /* SemiAG for [CLS]                                                             */
        Concatenate embedding $E \leftarrow [\mathbf{H}_T; \mathcal{M}]$ for [CLS] token and construct sub-graph $\mathbf{G}'_{\mathcal{H}}$.
        Compute binarized affinity graph $\mathbf{G}'_b$ from $\mathbf{G}'_{\mathcal{H}}$ by applying SemiAG in Eq. (4) (5) (6) sequentially.
        Obtain pseudo positives $\mathcal{P}_a$ and pseudo anchors $\mathcal{A}_a$ from $\mathbf{G}'_b$.
        Compute CAL loss $L_{\text{CAL}}^{\text{CLS}}$ for [CLS] with $\mathcal{P}_a$ and $\mathcal{A}_a$ on $\mathbf{H}$ by Eq. (7).
        /* SemiAG for [P], similar process to [CLS]                                     */
        Concatenate embedding $E_{\text{P}} \leftarrow [\overline{\mathbf{H}}_{\text{P},T}; \mathcal{M}_{\text{P}}]$ for [P] token and construct sub-graph $\mathbf{G}'_{\text{P},\mathcal{H}}$.
        Compute $\mathbf{G}'_{\text{P},b}$ from $\mathbf{G}'_{\text{P},\mathcal{H}}$ by applying Eq. (4) (5) (6) sequentially.
        Obtain pseudo labels $\mathcal{P}_{\text{P},a}$ and $\mathcal{A}_{\text{P},a}$ from $\mathbf{G}'_{\text{P},b}$.
        Compute CAL loss $L_{\text{CAL}}^{\text{P}}$ for [P] with $\mathcal{P}_{\text{P},a}$ and $\mathcal{A}_{\text{P},a}$ on $\overline{\mathbf{H}}_{\text{P},T}$ by Eq. (7).
        /* SemiCL loss                                                                  */
        Compute $L_{\text{sup}}^{\text{CLS}}, L_{\text{self}}^{\text{CLS}}$ for [CLS] and $L_{\text{sup}}^{\text{P}}, L_{\text{self}}^{\text{P}}$ for [P] on $\mathbf{Z}$ and $\mathbf{Z}_T$ by Eq. (8).
        /* Compute total loss                                                           */
        Compute [CLS] total loss $L_2^{\text{CLS}}$ with $L_{\text{sup}}^{\text{CLS}}, L_{\text{self}}^{\text{CLS}}, L_{\text{CAL}}^{\text{CLS}}$ by Eq. (9).
        Compute overall total loss $L_2$ with $L_2^{\text{CLS}}$ and its DPR counterpart $L_2^{\text{P}}$ by Eq. (10).
        /* Back propagation                                                             */
        Back-propagation and optimize student $\theta, \theta_{\text{P}}, \theta_{\text{H}}, \theta_{\text{P,H}}$.
        $\mathcal{M} \leftarrow \text{Enqueue}(\mathcal{M}, \mathbf{H}_T), \mathcal{M}_{\text{P}} \leftarrow \text{Enqueue}(\mathcal{M}_{\text{P}}, \overline{\mathbf{H}}_{\text{P},T})$ // update memories
        Update momentum teacher with current student.
return $f(\cdot|\theta, \theta_P)$
```

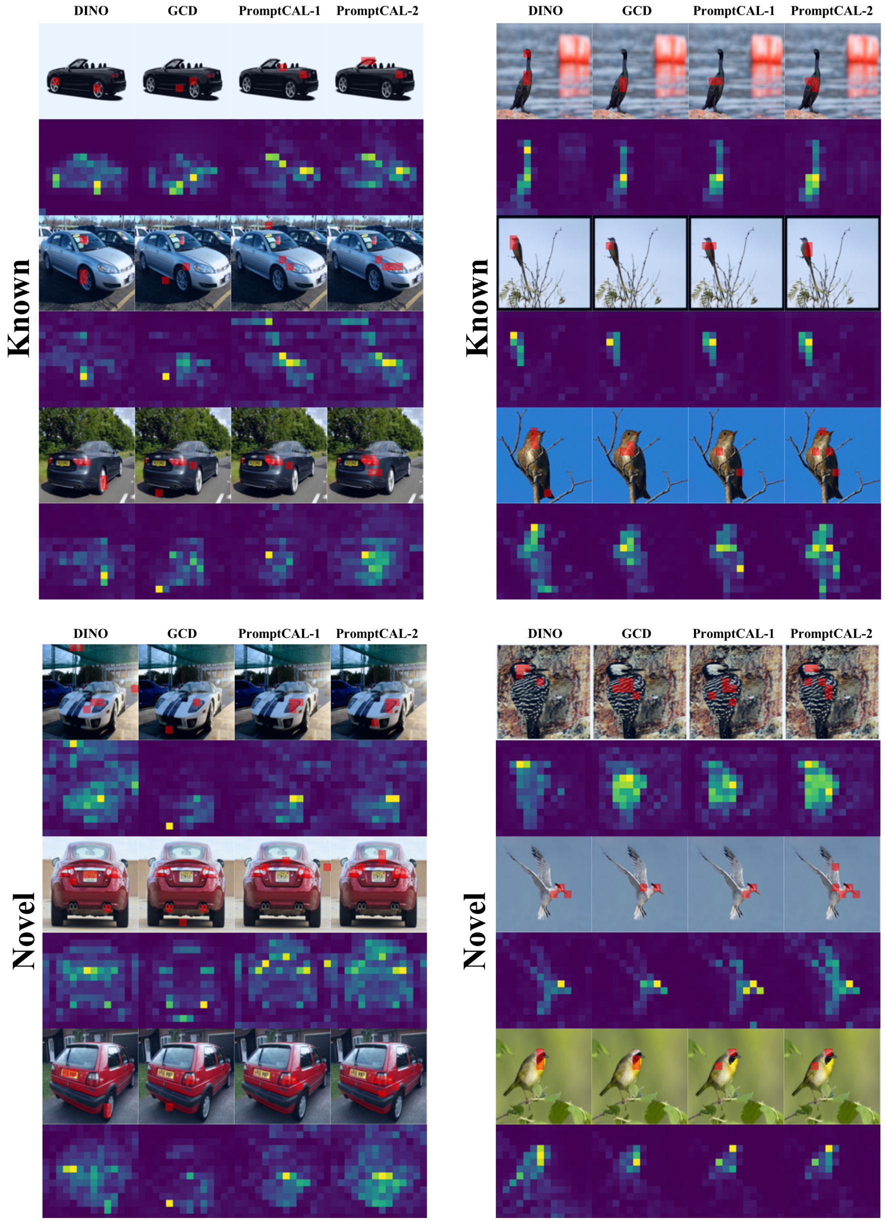


Figure 3. **Attention map visualization of class tokens for comparison on StandfordCars [8] (left) and CUB-200 [17] (right) datasets.** The columns from left to right refer to attention maps of DINO [3], GCD [16], our first stage PromptCAL, and our second stage PromptCAL. In the first row, attended areas are marked in red in each images; the second row display the complete attention maps corresponding to the first row images (yellow regions denote high attention values).

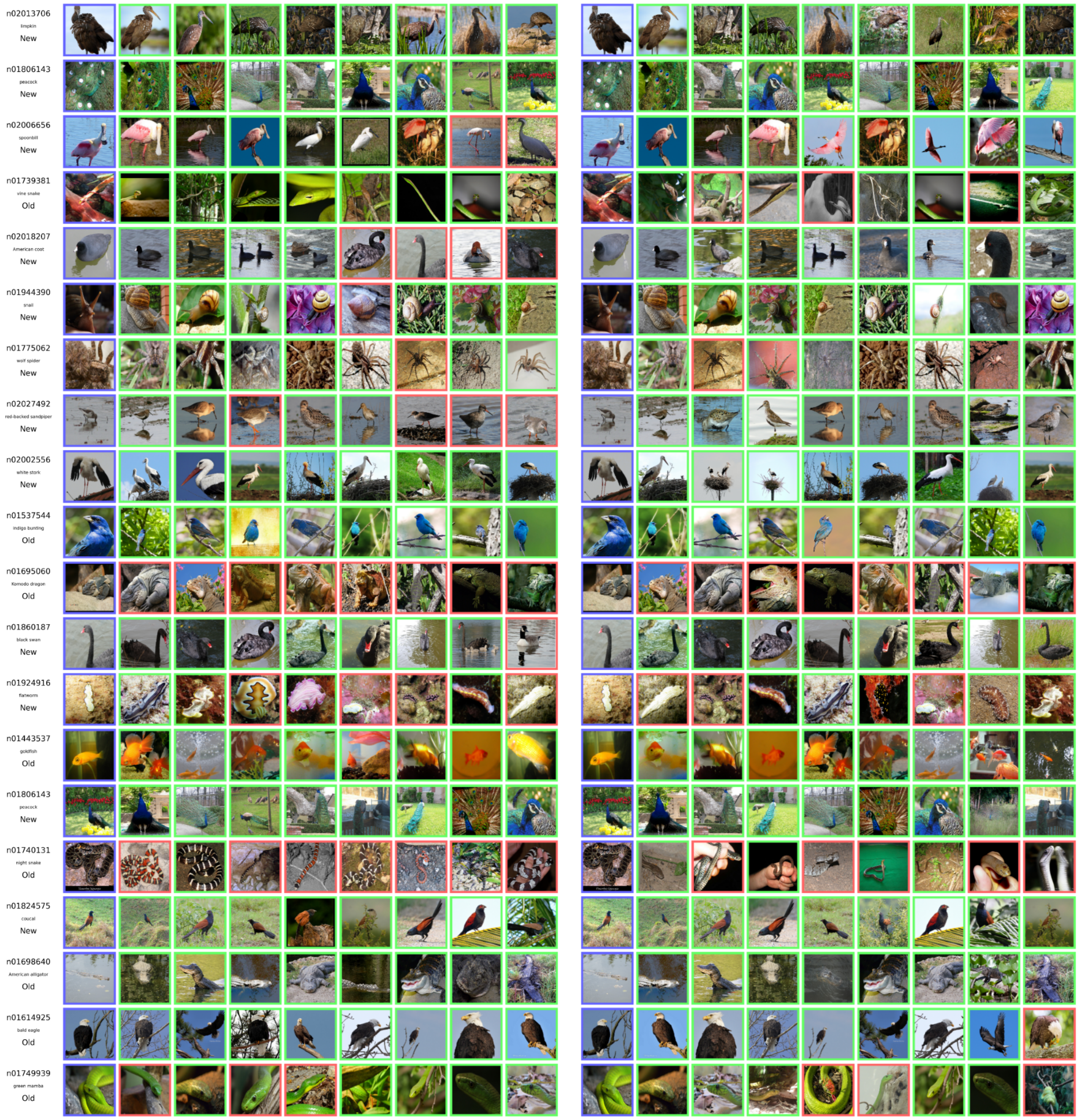


Figure 4. **Visualization of retrieved** 8**-NN for** 20 **randomly selected query images (with blue borders).** The correct/incorrect predictions are marked with green/red borders. The predictions on the left come from GCD, and the right is from PromptCAL. The first column contains ImageNet synsetIDs, category name, and `Known`/`New` for each query. Better view with zoom in.

References

[1] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006. 4

[2] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021. 2, 4, 5

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2, 3, 4, 5, 8

[4] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9284–9292, 2021. 4, 5

[5] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. *arXiv preprint arXiv:2002.05714*, 2020. 2, 3, 4

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[7] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 1, 3

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1, 4, 6, 8

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 2, 3, 4, 6

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1, 2, 3, 6

[11] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 4, 6

[12] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 2

[13] Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth? *arXiv preprint arXiv:2103.13915*, 2021. 1

[14] Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019. 4, 6

[15] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2, 3

[16] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7501, 2022. 1, 2, 3, 4, 5, 6, 8

[17] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1, 2, 3, 5, 6, 8

[18] Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki. Affinity learning with diffusion on tensor product graph. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):28–38, 2012. 1

[19] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 3, 4