Contrastive Mean-Shift Learning for Generalized Category Discovery

Sua Choi Dahyun Kang Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
https://cvlab.postech.ac.kr/research/cms

Abstract

We address the problem of generalized category discovery (GCD) that aims to partition a partially labeled collection of images; only a small part of the collection is labeled and the total number of target classes is unknown. To address this generalized image clustering problem, we revisit the mean-shift algorithm, i.e.*, a classic, powerful technique for mode seeking, and incorporate it into a contrastive learning framework. The proposed method, dubbed Contrastive Mean-Shift (CMS) learning, trains an embedding network to produce representations with better clustering properties by an iterative process of mean shift and contrastive update. Experiments demonstrate that our method, both in settings with and without the total number of clusters being known, achieves state-of-the-art performance on six public GCD benchmarks without bells and whistles.*

1. Introduction

Clustering is one of the most fundamental problems in unsupervised learning, which aims to partition instances of a data collection into different groups [2, 15, 34, 42]. Unlike the classification problem, it does not assume either predefined target classes or labeled instances in its standard setup. However, in a practical scenario, some data instances may be labeled for a subset of target classes so that we can leverage them to cluster all the data instances while also discovering the remaining unknown classes. The goal of Generalized Category Discovery (GCD) [48] is to tackle such a semi-supervised image clustering problem given a small amount of incomplete supervision.

Clustering is a transductive reasoning process based on the neighborhood data in the given data collection. To learn an image embedding for this clustering purpose, we are motivated to incorporate the neighborhood embeddings into learning. We revisit mean shift [8, 11, 12, 18, 44], *i.e.*, a classic, powerful technique for mode seeking and clustering analysis. The mean-shift algorithm consists of iterative mode-seeking steps of updating each data point by kernel-weighted aggregation of its neighboring data points; this process is non-parametric and does not require any information about the target clusters, *e.g.*, the number of clusters. For GCD, we develop a GPU-friendly mean-shift variant and incorporate it into a contrastive representation learning framework [7, 24, 59].

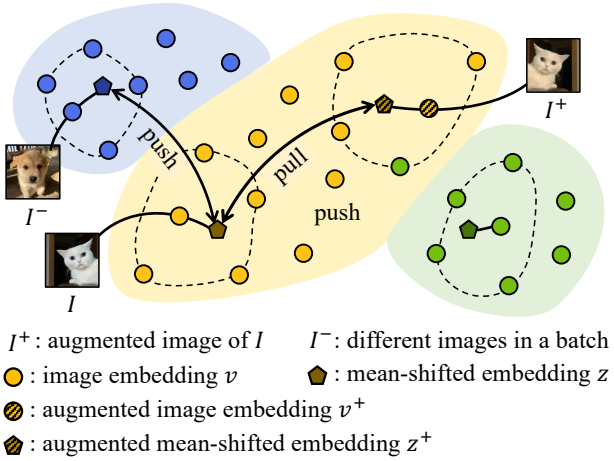

Figure 1. **Contrastive Mean-Shift (CMS) learning.** We integrate mean shift [11] into contrastive learning [7, 24, 59]. In training, image embeddings proceed a single-step mean shift with their kNNs. The contrastive learning objective pulls the mean-shifted embeddings of $\mathbf{I}$ and $\mathbf{I}^+$, while it pushes those from different image inputs. Colors denote different classes and k=4.

We introduce Contrastive Mean-Shift (CMS) learning for GCD. CMS learning aims to encode the semantic distance between image embeddings on a mean-shifted space via contrastive learning. Precisely, we perform a single-step mean shift for each image embedding by moving it toward the mean of its neighbors in an embedding space. To perform stable mean shifts in the embedding space, we use k nearest neighbors (kNNs) instead of typical distance-based neighbors [8, 11]. In learning, an image $\mathbf{I}$ and its augmented image $\mathbf{I}'$ are both mean-shifted and pull each other on the embedding space while pushing $\mathbf{I}$ and different images (Fig. 1). The training objective with mean-shifted em-

beddings encourages the embedding network to learn image representations with better clustering properties. After training the network, the actual clustering is performed by agglomerative clustering with the learned embeddings.

Compared to ours, prior arts [9, 37, 48, 51, 56] on GCD often employ K-means clustering [2, 34] during the validation and testing phases. To determine its hyperparameter K, the ground-truth number of classes K is often exploited in implementation. The use of the ground-truth K is not only critical to the clustering performance but also undesirable in practical clustering scenarios. We thus suggest jointly estimating K within the training stage to strictly avoid using the ground-truth K for clustering evaluation as well as for model validation during training.

Our proposed pipeline is extensively evaluated on the six public GCD datasets [19, 27, 28, 35, 45, 49] including the coarse-grained and fine-grained image classification datasets and achieves the state-of-the-art performance on the standard and the inductive GCD benchmarks [48, 56].

Our pipeline without given the ground-truth class number K shows comparable performance than the state of the arts that do exploit it, moreover, ours presents an even superior performance with using the ground-truth K. The contribution of our work can be summarized as follows:

- We revisit the mean-shift algorithm and integrate it with a contrastive learning framework for GCD.
- Our model performs clustering with the total number of target classes being completely unknown, which has been often overlooked by previous work on GCD.
- The proposed method is simple yet effective; it introduces zero extra trainable modules yet achieves state-of-the-art performance on the public GCD benchmark.

2. Related work

2.1. Generalized Category Discovery (GCD)

The task of GCD [48] aims to classify a collection of images into a class among the known and unknown classes, given a subset of images labeled with the known classes. Unlike the standard classification [29, 43] which assumes a preset number of total classes, the presence of the unknown classes is the essence of GCD. GCD stems from novel category discovery [20] of which unlabeled images are strictly from unknown classes, whereas in the GCD task, an unlabeled image might be from either known or unknown classes.

Existing work focuses on transferring the clue from labeled images to unlabeled ones. A line of work suggests leveraging pseudo-labels of the unlabeled images for learning: DCCL [37] adopts InfoMap clustering for pseudo-labeling, PromptCAL [56] discovers pseudo-positive samples based on semi-supervised affinity generation, and SimGCD [51] adopts a parametric classifier by distilling reliable pseudo-labels. The other popular approach suggests semi-supervised learning objective [23, 39, 54, 58], which employs the supervised contrastive loss with labeled images and self-supervised contrastive loss [48], or adopting a bi-level optimization framework of mutual information [9].

Despite of this remarkable progress, it is worth noting that most previous work adopts the sequential two-stage framework of estimating the number of classes after training the model, which is more favorable to combine them jointly. A few [37, 57] enable to jointly estimate the number of classes during training; however, their cost is often expensive: heavy computation for updating pairwise similarities of the entire training set [37], the dependency to manual hyperparameter tuning for cluster merging and splitting [57]. Furthermore, most work has exploited the ground-truth number of classes for model selection [9, 37, 48, 56], or defining a classifier [51].

In comparison, our approach shows powerful performance without access to the ground-truth number of classes *at all*, which is estimated efficiently in the training phase without the extra post-estimation process. Moreover, our simple learning framework involves no additional learning techniques, *e.g.*, teacher-student framework [51, 56], pseudo-labeling [37]. In terms of our model architecture, no extra module is introduced than a feature extractor, *e.g.*, add-on classifiers [51].

2.2. Mean shift

Mean shift [11] is a fundamental statistical method that identifies the mode, *i.e.*, the local maximum of data points, from a density function. The process iterates shifting each data point to the weighted average of a sample set until convergence. Introduced by Fukunaga and Hostetler [18], the theory is well founded and has been rigorously studied over decades [8, 16, 44]. Among other applications of mean shift [3, 10, 13, 26, 31], clustering [1, 6, 25, 52, 55] is one of the major applications. For example, Kobayashi and Otsu [25] perform mean-shift clustering on hypersphere using von Mises-Fisher distribution [36].

3. Preliminaries

3.1. Problem definition of GCD

The task of GCD aims to classify images when partially labeled images are given without knowing the number of target classes in advance. In other words, the dataset consists of labeled-known images, unlabeled-known images, and unlabeled-unknown images. Formally, a train set $\mathcal{D}_{\mathrm{T}}$ consists of a labeled set $\mathcal{D}_{\mathrm{L}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{L}}} \in \mathcal{X} \times \mathcal{Y}_{\mathrm{L}}$ and an unlabeled set $\mathcal{D}_{\mathrm{U}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{U}}} \in \mathcal{X} \times \mathcal{Y}_{\mathrm{U}}$, where $\mathcal{Y}_{\mathrm{L}} \subset \mathcal{Y}_{\mathrm{U}}$. Here we denote the number of target classes as K, *i.e.*, $K = |\mathcal{Y}_{\mathrm{U}}|$, which is assumed to be unknown in clustering. A validation set $\mathcal{D}_{\mathrm{V}}$ consists of known and unknown images and only the labels of known images are given.

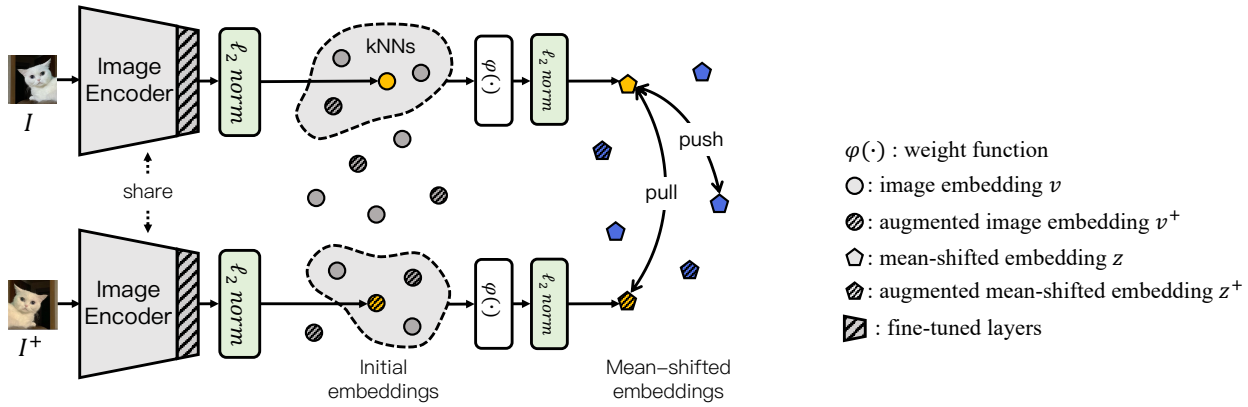

Figure 2. **Contrastive Mean-Shift learning.** The mean-shifted embedding $\boldsymbol{z}$ of an input $\mathbf{I}$ is represented as a combination of the input embedding $\boldsymbol{v}$ and the average of its kNN features, which are retrieved from the initial embeddings (gray points). The kNN search space is conducted on the pre-extracted all training images. The CMS loss pulls a mean-shifted embedding pair of $\mathbf{I}$ and its augmented image $\mathbf{I}^+$ (yellow points) and pushes the pair from different images apart (blue points).

3.2. Mean-shift algorithm

Mean shift is a non-parametric feature-space analysis technique for locating the maxima, *i.e.*, the modes, of a density function or data distribution [11]. Given a collection of data points $\mathcal{V}^0$ on a feature space, the weighted mean $m(\boldsymbol{v})$ of each data point $\boldsymbol{v}$ is calculated using its neighborhood $\mathcal{N}$. The mode of $\boldsymbol{v}$ sought by shifting its mean over multiple time steps t until convergence:

$$\boldsymbol{v}^{t+1} \leftarrow m(\boldsymbol{v}^t), \tag{1}$$

where $m(\boldsymbol{v}^t)$ is characterized by the two essential components: the set of neighbors $\mathcal{N}^t$ and the kernel function $\varphi(\cdot)$. For each time step, $\varphi(\cdot)$ determines the weight of neighbors in $\mathcal{N}^t$ for estimating the neighborhood mean:

$$m(\boldsymbol{v}^t) = \frac{\sum_{\boldsymbol{v}_i \in \mathcal{N}^t} \varphi(\boldsymbol{v}_i - \boldsymbol{v}^t)\boldsymbol{v}_i}{\sum_{\boldsymbol{v}_i \in \mathcal{N}^t} \varphi(\boldsymbol{v}_i - \boldsymbol{v}^t)}, \tag{2}$$

where $\mathcal{N}^t \subseteq \mathcal{V}^0$. In typical setups [8, 11, 12, 52], $\mathcal{N}$ is defined by a certain radius and the kernel function $\varphi(\cdot)$ is set to a uniform, Gaussian or Epanechnikov [41] kernel.

4. Our approach

We leverage the mean-shift clustering [11] for generalized category discovery and introduce *contrastive mean-shift learning*. For images to be clustered, we obtain their embeddings from an image embedding network [5] and update them with a single-step mean shift using their k-nearest-neighbors (kNNs) (Sec. 4.1). Then, we update the last layer of the image encoder based on the semantic distance of the mean-shifted embeddings through contrastive learning [7, 24, 59] (Sec. 4.2, Fig. 2). The number of classes K is jointly estimated during training based on agglomerative clustering [50]. After training, we iterate multiple mean shift steps, gradually identifying clusters on the embedding space (Sec. 4.3). The final cluster assignment is performed with the estimated K. The evaluation metric measures the accuracy between the optimal match between the ground-truth class set and the cluster assignment (Sec. 4.4).

4.1. Mean-shifted embedding

A collection of images on both the known and unknown set of classes is given: $\{\mathbf{I}_1, \cdots, \mathbf{I}_n\}$, which corresponds to the target data to be clustered. The images are first fed through an image feature extractor f to generate the corresponding set of d-dimensional $l2$-normalized image embeddings:

$$\mathcal{V} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n\}, \quad \text{where } \boldsymbol{v}_i = f(\mathbf{I}_i). \tag{3}$$

We use a self-supervised pre-trained image encoder, DINO [5], to provide a discriminative feature initialization, but our method is not restricted to a specific image encoder.

Hereafter we explain the formulation of a *mean-shifted embedding* $\boldsymbol{z}$ of the initial embedding $\boldsymbol{v}$ as a result of a single-step mean shift as in Eq. 1:

$$\boldsymbol{z} = m(\boldsymbol{v}), \tag{4}$$

with the following definition of the weighted mean $m(\cdot)$. The conventional mean-shift algorithm typically defines the neighborhood range engaged in the weighted mean based on the distance, *i.e.*, radius. In such a way, the number of neighbors within a radius may amount from zero to an arbitrary number, moreover, a fixed distance is unsuitable for learnable feature spaces that are constantly updated. To address these, we choose to approximate the fixed-radius nearest neighbors with the fixed-length kNNs, which is more suitable for parallel computation with GPUs. The neighborhood set $\mathcal{N}$ is defined with an input $\boldsymbol{v}$ and its kNNs:

$$\mathcal{N} = \{\boldsymbol{v}\} \cup \text{argmax}^k_{\boldsymbol{v}_i \in \mathcal{V}} \boldsymbol{v} \cdot \boldsymbol{v}_i, \tag{5}$$

where $\text{argmax}^k_{s\in\mathcal{S}}(\cdot)$ returns a subset of the top-k items that maximizes a target function. The obtained kNN embeddings are aggregated and then $l2$-normalized, ensuring the shifted embedding remains on a unit hypersphere:

$$m(\boldsymbol{v}) = \frac{\sum_{\boldsymbol{v}_i\in\mathcal{N}} \varphi(\boldsymbol{v}_i)\boldsymbol{v}_i}{||\sum_{\boldsymbol{v}_i\in\mathcal{N}} \varphi(\boldsymbol{v}_i)\boldsymbol{v}_i||}. \tag{6}$$

The $\varphi(\cdot)$ returns higher weights on the input $\boldsymbol{v}$ compared to the kNNs with a scaling hyperparameter α:

$$\varphi(\boldsymbol{v}_i) = \begin{cases} 1-\alpha & \text{if } \boldsymbol{v}_i = \boldsymbol{v} \\ \frac{\alpha}{k} & \text{otherwise,} \end{cases} \tag{7}$$

which can be interpreted as a rough approximation to a Gaussian kernel with an adaptive bandwidth. For updating the backbone, the single-step mean-shifted embedding $\boldsymbol{z}$ is utilized in a contrastive learning objective, described next.

4.2. Contrastive mean-shift learning

The proposed contrastive learning objective pulls positive image pair embeddings to each other and pushes the others, where the notion of positive and negative pairs differs with respect to the given image label. Note that the training set consists of the labeled and unlabeled set: $\mathcal{D}_\text{L}$ and $\mathcal{D}_\text{U}$. Our learning objective accordingly consists of two objectives: unsupervised contrastive mean-shift learning applied to all training images, $\mathcal{D}_\text{L} \cup \mathcal{D}_\text{U}$, and supervised contrastive learning applied to the labeled image portion, $\mathcal{D}_\text{L}$.

The contrastive mean-shift learning objective encourages the model to encode semantic distance on the mean-shift embedding space: mapping a pair of mean-shifted embeddings from the same image closely while pushing apart different image pairs in a batch. In detail, image augmentations from [48] are applied to all images in the current batch. An image $\mathbf{I}_i$ and its randomly augmented version $\mathbf{I}^+_i$ form an unsupervised positive image pair. Their mean-shifted embeddings, $\boldsymbol{z}_i$, $\boldsymbol{z}^+_i$, are pulled to each other, while the rest of the mean-shifted embeddings in the batch denoted as $\boldsymbol{z}'_j$ are pushed apart from $\boldsymbol{z}_i$. Formally,

$$\mathcal{L}_\text{CMS} = -\log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}^+_i/\tau_u)}{\sum_{j\neq i}\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}'_j/\tau_u)}, \tag{8}$$

where $\boldsymbol{z}'_j$ is a mean-shifted embedding of either an original or an augmented image in a batch, and τ_u is a hyperparameter for adjusting the temperature.

We similarly form a supervised contrastive learning loss [24, 48] with the labeled images, which pulls the same class features to each other and pushes different class features based on the given ground-truth class labels:

$$\mathcal{L}_\text{S} = -\frac{1}{|\mathcal{P}_s(i)|}\sum_{p\in\mathcal{P}_s(i)} \log \frac{\exp(\boldsymbol{v}_i \cdot \boldsymbol{v}'_p/\tau_s)}{\sum_{j\notin\mathcal{P}_s(i)}\exp(\boldsymbol{v}_i \cdot \boldsymbol{v}'_j/\tau_s)}, \tag{9}$$

Algorithm 1 Iterative mean shift and clustering at inference

Input: $\mathcal{V}^0 = \mathcal{V}_\text{L} \cup \mathcal{V}_\text{U} = \{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N\}$: union set of labeled and unlabeled image embeddings
Input: k: number of retrieved nearest neighbors

```
acc_max ← 0
t ← 0

while True do
    for v^t in V^t do
        N^t ← {v^t} ∪ kNN set of v^t                    – Eq. (5)
        v^{t+1} ← v^t + m(v^t)                          – Eqs. (4-7)
    end for
    preds^{t+1} ← Agglomerative Clustering(V^{t+1})
    acc^{t+1} ← Compute Accuracy(preds^{t+1})

    if t > 1 and acc_L^{t-1} ≥ acc_L^t ≥ acc_L^{t+1} then
        i ← argmax_{i∈{t-1,t,t+1}} acc_L^i
        break
    end if
    t ← t + 1
end while
```

Output: acc^i: final clustering accuracy

where $\mathcal{P}_s(i)$ is a set of image indices of the same class in the current batch, $\boldsymbol{v}'$ denotes either an original or an augmented image feature, $\boldsymbol{v}'_j$ is a feature labeled as a different class in the current batch. The learning loss combines the two losses:

$$\mathcal{L} = \lambda\mathcal{L}_\text{S} + (1-\lambda)\mathcal{L}_\text{CMS}, \tag{10}$$

where λ denotes the weights of supervised contrastive loss.

Note that the mean-shifted embedding learning process introduces *zero extra trainable modules*. The only trainable part affected by the learning loss corresponds to the last block of the pre-trained image encoder [5] and projection heads of which 6.3M trainable parameters.

4.3. Iterative mean shift at inference

The proposed mean-shifted embedding update proceeds with multiple steps once learning is converged. At inference, we iterate 1) the kNN retrieval, 2) additive feature update, and 3) agglomerative clustering accuracy evaluation on the labeled data until the accuracy drops or maintains two consecutive iterations. Algorithm 1 summarizes this inference process. Note that the conventional mean shift is to analyze the data points by finding the modes of the static data points, thus neighborhood is determined on the initial data points, *i.e.*, $\mathcal{N}^t \subseteq \mathcal{V}^0$, with an arbitrary mean-shift iteration step t. On the other hand, our iterative mean-shift process shifts all of the training data representations on the embedding space, therefore we set the kNN search space as the currently updated embedding space: $\mathcal{N}^t \subseteq \mathcal{V}^t$.

While exhibiting the identical form with the mean shift, our proposed mean-shift formulation does not necessarily guarantee convergence to the optimum. Rigorously speaking, the primary assumption for convergence of mean shift includes that the neighborhood kernel must be bounded, continuous, non-negative, normalized, and radially symmetric [32, 53]. However, our choice of mean vector involves kNN retrieval for the neighborhood set for the sake of computational efficiency, which discretizes the kernel weights at the cost of violating the continuity and radially symmetricity. Having said that, we empirically observe our kNN-based mean shift roughly converges as plotted in Figure 3 and contributes significantly to performance gain.

4.4. Clustering with cluster number estimation

In training, our framework jointly estimates the number of classes K and measures the validation accuracy with the predicted value. To estimate K, we adopt an agglomerative clustering algorithm [50] with ward linkage criterion, which iteratively merges the closest pair of instances until it reaches a certain threshold, *e.g.*, distance or number of clusters. At the end of every training epoch, $\mathcal{D}_{\text{V}}$ is clustered with different values of K. The validation accuracy and the optimal value of K are determined by the maximum accuracy of labeled images and its corresponding value of K. Once training is converged, the epoch with the highest accuracy is selected as the final model, and the value of K is also determined with the best model. This approach allows us to avoid accessing the ground-truth number of classes during both training and validation, unlike the previous work [9, 37, 48, 51, 56].

The proposed cluster number estimation is computationally more efficient than the prevalent one based on K-means clustering particularly on a small-sized clustering set; $\mathcal{D}_{\text{V}}$ is used for this purpose as $|\mathcal{D}_{\text{V}}| \ll |\mathcal{D}_{\text{T}}|$. To be specific, agglomerative clustering computes all instance-wise distances as a linkage matrix and produces different granularity of clusters via hierarchical grouping. The pre-computed linkage matrix is reused for any K-class clustering, thus $O(N^2)$ regardless of K, being especially effective with a few elements N and many trials of K. On the other hand, K-means clustering requires updating distance matrix for T times with different K centroids, hence $O(NKT)$. Since K is unknown, a series of trials T with different Ks costs extensive distance computation. In practice, K estimation with K-means clustering induces prohibitively long training time to run at every model validation, whereas, our efficient process enables joint estimation of K during training.

After clustering with the obtained K, the assignments are matched with ground-truth classes by the Hungarian optimal matching [30], based on the number of intersected instances between each pair of classes. The unpaired classes are considered incorrect predictions, while the instances of the most dominant class within each ground-truth cluster are considered correct when calculating the accuracy.

5. Experiments

5.1. Experimental setup

Datasets. We evaluate our method on 6 image classification benchmarks: 2 generic image datasets CIFAR100 [28], ImageNet100 [19, 40] and 4 fine-grained datasets CUB-200-2011 [49], Stanford Cars [27], FGVCAircraft [35], and Herbarium19 [45]. For splitting the target classes into known and unknown class sets, we follow the SSB [47] split for CUB-200, Stanford Cars, and Aircraft. For the other benchmarks, random splits are used with the same seed from [48]. A subset of labeled images is sampled from the known classes with 80% on CIFAR100 and 50% on other datasets following [48]. For the detailed class split for each dataset, please refer to Appendix 7.2 Table 9.

Training details. We use pre-trained DINO ViT-B/16 [5, 14] including the projection head layer on top of it as a backbone, following the existing methods [37, 48, 56] for comparison. The last layer and the projection heads are fine-tuned, where the projection head consists of three consecutive pairs of a 2048-dimensional linear layer followed by GeLU activation [22]. To reduce the computational cost of kNN retrieval, we keep the final output dimension of the projection head fixed at 768, unlike other methods that typically set it to 65536. The pre-extracted embedding set $\mathcal{V}$ consists of image features $\boldsymbol{v}$ of $\mathcal{D}_{\text{T}}$, detached from gradient computation and updated every epoch. The kNN size is set to 8. The scaling hyperparameter α for mean-shifted embedding is set to 0.5. The temperature hyperparameter τ_u and learning rate are set to 0.3, 0.01 for coarse-grained benchmarks and 0.25, 0.05 for fine-grained benchmarks. Other hyperparameters such as batch size, τ_s, weight decay, and the number of augmented images are set to 128, 0.07, 5e^{-5}, and 2, respectively, following the previous work [48]. All experiments are run on an RTX-3090 GPU.

Evaluation. In GCD, a test set is utilized for validation, and the performance is evaluated by clustering a train set $\mathcal{D}_{\text{T}}$, and then measuring the accuracy score on $\mathcal{D}_{\text{U}}$. The accuracy is reported on "All" unlabeled data as well as the accuracy on those of known and unknown classes, denoted as "Old" and "Novel" in tables, respectively. We evaluate the performance using the output of the iterative inference step. We report the accuracy with the estimated K as well as using the GT number K, following the previous work that assumes given number of target classes during evaluation.

5.2. Main results

Evaluation on GCD. Table 1 presents a comparison on the GCD setup in both coarse-grained and fine-grained benchmarks with or without the ground-truth (GT) number of

Method	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
(a) Clustering with the ground-truth number of classes K given																		
Agglomerative [50]†	56.9	56.6	57.5	73.1	77.9	70.6	37.0	36.2	37.3	12.5	14.1	11.7	15.5	12.9	16.9	14.4	14.6	14.4
RankStats+ [21]	58.2	77.6	19.3	37.1	61.6	24.8	33.3	51.6	24.2	28.3	61.8	12.1	26.9	36.4	22.2	27.9	55.8	12.8
UNO+ [17]	69.5	80.6	47.2	70.3	95.0	57.9	35.1	49.0	28.1	35.5	70.5	18.6	40.3	56.4	32.2	28.3	53.7	14.7
ORCA [4]	69.0	77.4	52.0	73.5	92.6	63.9	35.3	45.6	30.2	23.5	50.1	10.7	22.0	31.8	17.1	20.9	30.9	15.5
GCD [48]	73.0	76.2	66.5	74.1	89.8	66.3	51.3	56.6	48.7	39.0	57.6	29.9	45.0	41.1	46.9	35.4	51.0	27.0
DCCL [37]	75.3	76.8	70.2	80.5	90.5	76.2	63.5	60.8	**64.9**	43.1	55.7	36.2	-	-	-	-	-	-
PromptCAL [56]	81.2	84.2	75.3	83.1	92.7	78.3	62.9	64.4	62.1	50.2	70.1	40.6	52.2	52.2	**52.3**	37.0	52.0	28.9
GPC [57]	77.9	85.0	63.0	76.9	94.3	71.0	55.4	58.2	53.1	42.8	59.2	32.8	46.3	42.5	47.9	-	-	-
SimGCD [51]	80.1	81.2	**77.8**	83.0	93.1	77.9	60.3	65.6	57.7	53.8	71.9	45.0	54.2	59.1	51.8	**44.0**	**58.0**	**36.4**
PIM [9]	78.3	84.2	66.5	83.1	95.3	77.0	62.7	75.7	56.2	43.1	66.9	31.6	-	-	-	42.3	56.1	34.8
Ours	**82.3**	**85.7**	75.5	**84.7**	**95.6**	**79.2**	**68.2**	**76.5**	64.0	**56.9**	**76.1**	**47.6**	**56.0**	**63.4**	**52.3**	36.4	54.9	26.4
(b) Clustering without the ground-truth number of classes K given																		
Agglomerative [50]†	56.9	56.6	57.5	72.2	77.8	69.4	35.7	33.3	36.9	10.8	10.6	10.9	14.1	10.3	16.0	13.9	13.6	14.1
GCD [48]	70.8	77.6	57.0	77.9	91.1	71.3	51.1	56.4	48.4	39.1	58.6	29.7	-	-	-	37.2	51.7	29.4
GPC [57]	75.4	**84.6**	60.1	75.3	93.4	66.7	52.0	55.5	47.5	38.2	58.9	27.4	43.3	40.7	44.8	36.5	51.7	27.9
PIM [9]	75.6	81.6	63.6	**83.0**	95.3	**76.9**	62.0	**75.7**	55.1	42.4	65.3	31.3	-	-	-	**42.0**	55.5	**34.7**
Ours	**79.6**	83.2	**72.3**	81.3	**95.6**	74.2	**64.4**	68.2	**62.2**	**51.7**	**68.9**	**43.4**	**55.2**	**60.6**	**52.4**	37.4	**56.5**	27.1

Table 1. Comparison with the state of the arts on GCD, evaluated *with* or *without* the GT K for clustering. † denotes reproduced results.

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]	✓	70.1	76.8	43.5	79.7	92.7	66.7	57.5	64.5	50.6	-	-	-	-	-	-	-	-	-
ORCA [4]	✓	77.7	83.6	53.9	81.3	94.5	68.0	40.7	61.2	20.2									
PromptCAL [56]†	✓	**81.6**	**85.3**	**66.9**	84.8	94.4	75.2	62.4	68.1	56.8	**62.3**	**76.9**	**48.2**	43.6	49.5	37.7	37.6	50.3	30.7
Ours	✓	80.7	84.4	65.9	**85.7**	**95.7**	**75.8**	**69.7**	**76.5**	**63.0**	57.8	75.2	41.0	53.3	**62.7**	43.8	**46.2**	**53.0**	**38.9**
Ours		80.5	84.5	64.4	84.2	95.6	72.9	69.0	76.4	61.7	57.9	75.6	40.8	**53.8**	62.6	**44.9**	42.4	53.5	30.7

Table 2. Comparison of our model and the state-of-the-art models on the inductive GCD setup on six datasets. The performance of [4, 48] is taken from PromptCAL. We reproduced PromptCAL on Stanford Cars, FGVC Aircraft and Herbarium19 with its official implementation.

classes K given for clustering. In Table 1 (a), we compare our method with the state-of-the-art methods, all evaluated with the ground-truth number of classes. We present an agglomerative clustering [50] baseline with the pre-trained DINO not trained further. Our model achieves significant gains over the training-free agglomerative clustering baseline, which signifies the efficacy of the contrastive mean-shift learning. The other state-of-the-art methods adopt semi-supervised K-means clustering, where the K centroids are initialized by the labeled data with the GT K. Note that we do not access the GT K for model selection during training in this setup as well, but use it only for model evaluation after training. Our method outperforms existing methods and achieves state-of-the-art performance on five out of six datasets. The performance gain is particularly significant on Standard Cars and CUB benchmarks, with 4.7% and 3.1% point higher accuracy, respectively. Generally, our model gains notable performance on both Old and Novel classes, which implies that the knowledge acquired from known classes is successfully transferred to unknown classes through the use of nearest neighbor embeddings according to the input query.

In Table 1 (b), we present the comparison of ours and the state of the arts on the same setup with Table 1 (a) but without having the GT number of classes K known for clustering. For the result of Vaze *et al*. [48], we take the numerical results from PIM [9]. Our method shows outstanding performance in most scenarios even though it does not access to the GT K in both training and testing. Our method is even superior to the state-of-the-art methods measured with the known value of K on CUB and FGVC Aircraft. The results show that our K-estimation process incorporated in the training phase performs effectively with no significant performance drop compared to the known-K counterparts.

Evaluation on inductive GCD. We also compare the clustering results on the inductive GCD setup presented in PromptCAL [56]. Contrary to the *transductive* GCD problem setup in Table 1, the inductive GCD setup evaluates the performance on the *unseen test set*. In this setup, a subset of the training set is used for validation, and the labeled

Method	CIFAR100		ImageNet100		CUB		Stanford Cars		FGVC Aircraft		Herbarium 19	
	K	Err(%)	K	Err(%)	K	Err(%)	K	Err(%)	K	Err(%)	K	Err(%)
Ground truth	100	-	100	-	200	-	196	-	100	-	683	-
GCD [48]	100	0	109	9	231	15.5	230	17.3	-	-	520	23.8
DCCL [37]	146	46	129	29	172	9	192	0.02	-	-	-	-
PIM [9]	95	5	102	2	227	13.5	169	13.8	-	-	563	17.6
GPC [57]	100	0	103	3	212	6	201	0.03	-	-	-	-
Ours	95	5	116	16	168	16	156	20.4	90	10	622	8.9
Ours*	97	3	116	16	170	15	156	20.4	98	2	666	2.5

Table 3. Estimated number and an error rate of a class number K

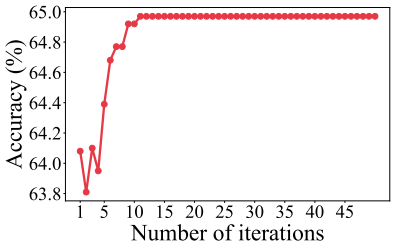

Figure 3. Clustering accuracy over mean-shift iteration on CUB.

	training	inference		CIFAR100			ImageNet100			CUB			Stanford Cars		
	CMS	SSK	IMS	All	Old	Novel	All	Old	Novel	All	Old	Novel	All	Old	Novel
(1)		✓		71.5	77.3	60.1	74.1	89.8	66.3	51.2	49.2	52.2	37.9	57.8	28.3
(2)			✓	71.6	77.3	60.0	80.3	91.7	74.6	58.7	62.0	57.1	40.8	54.5	34.2
(3)	✓	✓		81.1	85.6	72.1	83.4	**95.8**	77.2	66.7	75.3	62.5	54.5	76.4	43.9
(4)	✓		⋄ (1-step)	80.1	**86.0**	68.4	84.1	95.6	78.3	**68.2**	76.4	**64.1**	56.1	74.6	47.1
(5)	✓		✓	**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 4. Effectiveness of each component of our method. SSK denotes semi-supervised K-means clustering and IMS iterative mean-shift.

validation set is utilized to verify the termination condition during the iterative inference process. The comparison is presented in Table 2, where our method exhibits superior performance in the inductive category discovery scenario as well. We also report the performance measured with the estimated number of classes K, which is more practical as it assumes both unseen data and unknown classes in a test set. Our method shows comparable or even higher performance without the ground-truth K than the other models. The result demonstrates that incorporating nearest-neighbor embeddings enhances image clustering by ensuring consistency among relevant images, thus enabling generalization to discover novel classes.

Estimated number of clusters. Table 3 shows the comparison of the ground-truth K, ours, and others reported by Vaze *et al*., DCCL, PIM, and GPC. Among these baselines, DCCL and GPC jointly estimate the number of classes during training, while the others post-estimate K after training as done in [48]. Our method estimates class number on par with others *without exploiting any dataset-specific hyperparameters*. When utilizing the ground-truth K during validation as with other baselines (Ours*), the estimates become more accurate. Note that the K values are estimated on the validation set, which is relatively small and hence computationally efficient for clustering compared to the previous work [9, 37, 48, 57] which uses the entire training set.

5.3. Ablation study

Performance over mean-shift iterations. In Figure 3, we examine the clustering accuracy of our method over iterations during the inference phase on the CUB benchmark. The iteration 1 indicates the clustering accuracy using the features extracted from the trained backbone, which is learned to transform a given image to a probable shifted-feature space. The result demonstrates that iterating the mean shift during inference leads to an additional performance gain without further training. We empirically observe that the accuracy saturates beyond a certain optimal number of iterations, akin to the behavior of the conventional mean-shift algorithm. This result shows that iterative mean shift based on kNNs roughly approximates the mean-shift kernels that guarantee convergence.

Effect of each proposed component. Table 4 shows the ablation of CMS learning (Sec. 4.2) and Iterative Mean Shift (IMS, Sec. 4.3). For training, we examine the effect of the embedding without mean shift, *i.e*., equivalent to the embedding in Vaze *et al*. [48]. At inference, semi-supervised K-means clustering (SSK) [48], single-step mean shift, and IMS are compared. Comparing (1) *vs* (3) and (2) *vs* (5), we observe that CMS learning boosts performance significantly. After training, IMS brings additional gains at inference when comparing (1) *vs* (2) and (3) *vs* (5), plus recursive iterations: (4) *vs* (5). The final model (5) outperforms others with the combined gain of each proposed component.

Comparison with different mean-shift kernels. We validate our kNN-based mean-shift learning framework by replacing the neighborhood criterion with uniform and Gaussian kernels, which are commonly used for mean shift. For implementation details, please refer to Appendix 7.1. As shown in Table 5, the performance significantly deteriorates with both kernels. For constantly updating embedding spaces, it is tricky to set a fixed distance radius for mean-shift kernels. Noticeably, we observe that the fixed-radius kernels tend to blur the embedding space by incorporating more neighbors within its radius as training progresses. In other words, this makes irrelevant embeddings to be involved over training, eventually leading the model to produce indistinguishable image embeddings to each other.

Kernel	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
Uniform	75.2	77.6	70.5	76.9	92.0	69.3	53.5	59.4	50.6	34.3	54.7	24.5	39.9	43.6	38.1	36.0	**55.8**	25.4
Gaussian	72.2	80.6	55.4	67.2	94.9	53.2	45.5	43.2	46.6	34.9	56.2	24.5	36.6	36.7	36.6	25.6	38.4	18.8
Ours	**82.3**	**85.7**	**75.5**	**84.7**	**95.6**	**79.2**	**68.2**	**76.5**	**64.0**	**56.9**	**76.1**	**47.6**	**56.0**	**63.4**	**52.3**	**36.4**	54.9	**26.4**

Table 5. Comparison of ours and different mean-shift kernels on GCD.

	$\mathcal{N}$	$\varphi(\cdot)$	ImageNet100			CUB		
			All	Old	Novel	All	Old	Novel
(1)	kNN	attention	82.2	95.2	75.7	63.6	70.1	60.3
(2)	random	mean	82.6	75.0	76.3	58.1	65.4	54.5
(3)	kNN	mean	**84.7**	**95.6**	**79.2**	**68.2**	**76.5**	**64.0**

Table 6. Comparison with different feature aggregation methods

method	CIFAR100	ImageNet100	CUB
Vaze *et al*. [48]	54.8	74.1	27.9
Ours	**60.5** (+5.7%p)	**77.9** (+3.8%p)	**32.4** (+4.5%p)

Table 7. Comparison of two methods on the unsupervised setup

	CIFAR100			ImageNet100			CUB		
setup	All	Old	Novel	All	Old	Novel	All	Old	Novel
Unsup.	67.4	67.0	67.6	88.1	92.2	86.8	29.5	35.2	27.6
GCD [48]	87.9	97.8	81.2	90.2	99.2	87.2	82.6	98.5	64.0

Table 8. The kNN retrieval performance of ours in Recall@8 on the unsupervised category discovery and the GCD setups

Notice that the performance gap with ours (with kNN) is larger on the small-scale benchmarks: CUB, Cars, and Aircraft, which might be more sensitively influenced by the kernel parameters on less data. Through this, we validate that kNN retrieval performs as a less brittle and more GPU-friendly approximation of a Gaussian kernel for feature learning, which adjusts the radius of the mean-shift kernel as the embedding space is updated.

Comparison with different embedding aggregation. We verify the proposed contrastive mean-shift learning by replacing the nearest neighbor retrieval and mean aggregation with random retrieval and learnable attentive aggregation. For the learnable attentive aggregation, we adopt the cross-attention mechanism [46] without the value projection, using a query embedding as a query and kNN embeddings as key and value. Without value projection, attentive pooling of kNN embeddings is analogous to the *attentive* mean shift. For random aggregation, we randomly select k embeddings for each query image instead of the nearest neighbors, which are then shared among both the original and augmented embeddings for stable learning.

As shown in Table 6, the mean aggregation with kNNs is the most effective one. We observe that the random aggregation significantly deteriorates the performance as training progresses. On the other hand, the attentive aggregation involves more trainable parameters and exhibits stable learning curves in training but generalizes worse than the mean aggregation method at inference with iterative mean shift, *e.g.*, 2.6% point drop after iterations on ImageNet100.

Unsupervised category discovery with CMS. We further analyze our method to interpret its behavior and elucidate why it performs particularly well on GCD by comparing ours on the unsupervised category discovery setup. As shown in Table 7, CMS improves clustering effects even in an unsupervised setup but quickly enhances with additional labels, leveraging the help of higher-quality kNNs. Specifically, when comparing CMS and Vaze *et al*. [48] without labels, ours better performs by an average of 4.7%p across three benchmarks. The gap widens when using additional labels, *i.e.*, on the GCD task, as shown in Table 1. We then evaluate the quality of the retrieved 8NNs in our method across these setups (Table 8), and observe that the model trained on the GCD setup retrieves more accurate embeddings *even for unknown classes* than the one trained on the unsupervised setup. Since CMS propagates supervision in incorporating the relevant kNNs, and the relevant kNNs lead robust mean shift as examined in Table 6, it eventually establishes better representations for final clustering.

6. Conclusion

We have proposed to revisit the mean-shift algorithm and incorporated it with contrastive representation learning for generalized category discovery. The training procedure trains an embedding network via contrastive learning of the single-step mean-shifted embeddings. The evaluation procedure iterates the mean-shift steps, mapping the resultant clustered groups to categories. While the previous work on GCD often exploits the ground-truth number of classes for clustering, we avoid doing so and propose a cluster number estimation process based on agglomerative clustering to enable clustering in the absence of the ground-truth number of classes. In experiments, our method achieves state-of-the-art performance on the public GCD benchmarks without bells and whistles.

Acknowledgements. This work was supported by the NRF grant (NRF-2021R1A2C3012728 (50%)) and the IITP grants (2022-0-00113: Developing a Sustainable Collaborative Multi-modal Lifelong Learning Framework (45%), 2019-0-01906: AI Graduate School Program at POSTECH (5%)) funded by Ministry of Science and ICT, Korea.

References

[1] Saket Anand, Sushil Mittal, Oncel Tuzel, and Peter Meer. Semi-supervised kernel mean shift clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. 2

[2] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007. 1, 2

[3] Antonio Bandera, José Manuel Pérez-Lorenzo, Juan Pedro Bandera, and F Sandoval. Mean shift based clustering of hough domain for fast line segment detection. *Pattern Recognition Letters*, 2006. 2

[4] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *International Conference on Learning Representations*, 2022. 6

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 3, 4, 5

[6] José E Chacón. Mixture model modal clustering. *Advances in Data Analysis and Classification*, 2019. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. International Conference on Machine Learning (ICML)*, 2020. 1, 3

[8] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1995. 1, 2, 3

[9] Florent Chiaroni, Jose Dolz, Ziko Imtiaz Masud, Amar Mitiche, and Ismail Ben Ayed. Parametric information maximization for generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1729–1739, 2023. 2, 5, 6, 7

[10] Minsu Cho and Kyoung Mu Lee. Mode-seeking on graphs via random walks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012. 2

[11] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2002. 1, 2, 3

[12] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2000. 1, 3

[13] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2003. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 5

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, 1996. 1

[16] Mark Fashing and Carlo Tomasi. Mean shift is a bound optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2005. 2

[17] 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 (ICCV)*, 2021. 6

[18] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 1975. 1, 2

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proc. International Conference on Learning Representations (ICLR)*, 2019. 2, 5

[20] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8401–8409, 2019. 2

[21] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *International Conference on Learning Representations (ICLR)*, 2020. 6

[22] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[23] Dahyun Kang, Piotr Koniusz, Minsu Cho, and Naila Murray. Distilling self-supervised vision transformers for weakly-supervised few-shot classification & segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3, 4

[25] Takumi Kobayashi and Nobuyuki Otsu. Von mises-fisher mean shift for clustering on a hypersphere. In *2010 20th International Conference on Pattern Recognition*, pages 2130–2133. IEEE, 2010. 2

[26] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 10326–10335, 2021. 2

[27] 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. 2, 5, 1

[28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009. 2, 5, 1

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 2

[30] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[31] Abhishek Kumar, Oladayo S Ajani, Swagatam Das, and Rammohan Mallipeddi. Gridshift: A faster mode-seeking algorithm for image segmentation and object tracking. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[32] Xiangru Li, Zhanyi Hu, and Fuchao Wu. A note on the convergence of the mean shift. *Pattern recognition*, 2007. 5

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 2008. 4

[34] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Oakland, CA, USA, 1967. 1, 2

[35] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2, 5, 1

[36] Kanti V Mardia, Peter E Jupp, and KV Mardia. *Directional statistics*. Wiley Online Library, 2000. 2

[37] Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7579–7588, 2023. 2, 5, 6, 7

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[39] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 2, 1

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[41] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015. 3

[42] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 1973. 1

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Proc. International Conference on Learning Representations (ICLR)*, 2015. 2

[44] Maneesh Singh, Himanshu Arora, and Narendra Ahuja. A robust probabilistic estimation framework for parametric image models. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2004. 1, 2

[45] Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge Belongie. The herbarium challenge 2019 dataset. *arXiv preprint arXiv:1906.05372*, 2019. 2, 5, 1

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 8

[47] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? *arXiv preprint arXiv:2110.06207*, 2021. 5

[48] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 4, 5, 6, 7, 8, 3

[49] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, 2011. 2, 5, 1

[50] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963. 3, 5, 6

[51] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16590–16600, 2023. 2, 5, 6

[52] Kuo-Lung Wu and Miin-Shen Yang. Mean shift-based clustering. *Pattern Recognition*, 2007. 2, 3

[53] Ryoya Yamasaki and Toshiyuki Tanaka. Convergence analysis of mean shift. *arXiv preprint arXiv:2305.08463*, 2023. 5

[54] Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1978–1987, 2022. 2

[55] Xiao-Tong Yuan, Bao-Gang Hu, and Ran He. Agglomerative mean-shift clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2010. 2

[56] Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Shahbaz Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3479–3488, 2023. 2, 5, 6, 3

[57] Bingchen Zhao, Xin Wen, and Kai Han. Learning semi-supervised gaussian mixture models for generalized category discovery. *arXiv preprint arXiv:2305.06144*, 2023. 2, 6, 7

[58] Yi Zhou, Xiaodong He, Lei Huang, Li Liu, Fan Zhu, Shanshan Cui, and Ling Shao. Collaborative learning of semi-supervised segmentation and classification for medical images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[59] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings.

In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 3