GENERALIZED CATEGORY DISCOVERY VIA ADAPTIVE GMMS WITHOUT KNOWING THE CLASS NUMBER

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we address the problem of generalized category discovery (GCD), *i.e.*, given a set of images where part of them are labelled and the rest are not, the task is to automatically cluster the images in the unlabelled data, leveraging the information from the labelled data, while the unlabelled data contain images from the labelled classes and also new ones. GCD is similar to semi-supervised learning (SSL) but is more realistic and challenging, as SSL assumes all the unlabelled images are from the same classes as the labelled ones. We also do not assume the class number in the unlabelled data is known a-priori, making the GCD problem even harder. To tackle the problem of GCD without knowing the class number, we propose an EM-like framework that alternates between representation learning and class number estimation. We propose a semi-supervised variant of the Gaussian Mixture Model (GMM) with a stochastic splitting and merging mechanism to dynamically determine the prototypes by examining the cluster compactness and separability. With these prototypes, we leverage prototypical contrastive learning for representation learning on the partially labelled data subject to the constraints imposed by the labelled data. Our framework alternates between these two steps until convergence. The cluster assignment for an unlabelled instance can then be retrieved by identifying its nearest prototype. We comprehensively evaluate our framework on both generic image classification datasets and challenging fine-grained object recognition datasets, achieving state-of-the-art performance.

1 INTRODUCTION

The success of deep learning is driven by the availability of large-scale data with human annotations. Given enough annotated data, deep learning models are able to surpass human-level performance on many important computer vision tasks such as image classification (He et al., 2016). But the cost of collecting the large annotated dataset is not always affordable and it is also not possible to annotate all new classes emerging from the real world. Thus, designing models that can learn to deal with the large scale unlabelled data in the open world is of great value and importance. Semi-supervised learning (SSL) (Oliver et al., 2018) is proposed as a solution to learn a model on both labelled data and unlabelled data, with many works achieving promising performance (Berthelot et al., 2019; Tarvainen & Valpola, 2017; Sohn et al., 2020). However, SSL assumes that labelled instances are provided for all object classes in the unlabelled data. The novel category discovery (NCD) task is introduced (Han et al., 2019; 2021) to automatically discover novel classes by transferring the knowledge learned from the labelled instances of known classes, assuming the unlabelled data only contain instances from new classes. Generalized category discovery (GCD) (Vaze et al., 2022a) further relaxes the assumption in NCD, and tackles a more generalized setting where the unlabelled data contains instances from both known and novel categories. Existing methods for NCD (Han et al., 2019; 2021; Zhao & Han, 2021; Zhong et al., 2021a;b; Fini et al., 2021; Jia et al., 2021) and GCD (Vaze et al., 2022a) learn the representation and cluster assignment assuming the class number is known a priori (Zhao & Han, 2021; Jia et al., 2021; Zhong et al., 2021a; Fini et al., 2021; Zhong et al., 2021b) or precomputed (Han et al., 2019; Vaze et al., 2022a). In practice, the number of categories in the unlabelled data is often unknown, while precomputing the class number without taking the representation learning into consideration is likely to lead to a sub-optimal solution.

In this paper, we argue that representation learning and the estimation of class numbers should be considered together and could reinforce each other, *i.e.*, a strong representation could help a more accurate estimation of the class numbers, and an accurate class number could help learn a better

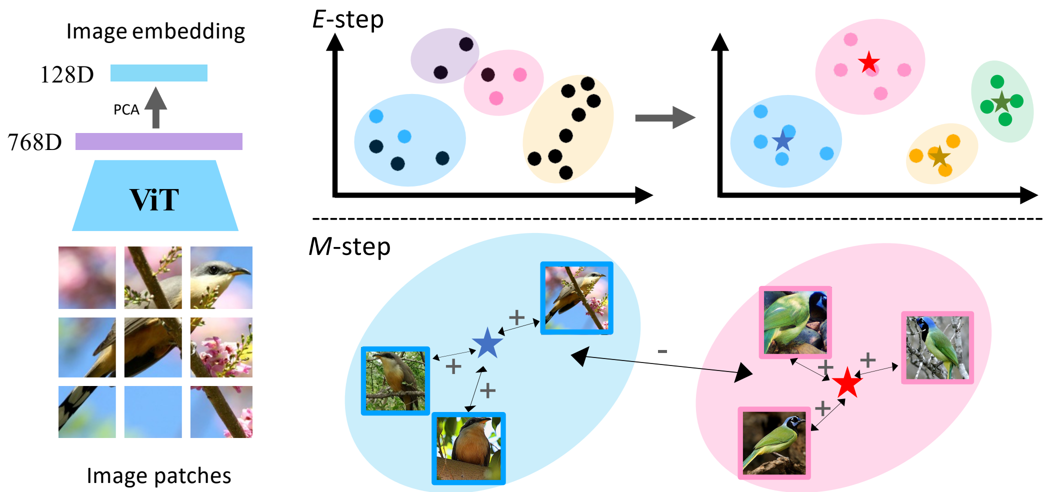


Figure 1: **Overview of our proposed EM-like framework.** The input images are fed into a ViT-B model to obtain a 768-dimensional feature vector, then the feature vector will be projected to a lower dimensional space using the projection calculated from PCA. We perform class number estimation and representation learning in this projected space. In the E-step, we use a semi-supervised GMM that can split separable clusters and merge cluttered clusters to estimate the class number and prototypes, which will be used in the M-step of representation learning with prototypical contrastive learning.

feature representation. To this end, we propose a unified EM-like framework that alternates between feature representation learning and class number estimation where the E-step is aimed at automatically estimating a proper class number and a set of class prototypes in the unlabelled data and the M-step is aimed at learning better representation with the class number and class prototypes estimated. In particular, we propose using a prototype contrastive representation learning (Li et al., 2020) method for GCD, which requires a set of prototypes to serve as anchors for representation learning. Prototypical contrastive learning (Li et al., 2020) is developed for unsupervised representation learning to generalized to different tasks, where the prototypes are obtained by over clustering the dataset with one or multiple given prototype numbers, using non-parametric clustering algorithms like k-means. Instead, to handle the problem of GCD, we propose to estimate the prototype number and prototypes automatically and simultaneously. To do so, we introduce a semi-supervised variant of the Gaussian Mixture Model (GMM) with a stochastic splitting and merging mechanism to determine the most suitable clusters based on current representation. These clusters can then be used to form prototypes to facilitate contrastive representation learning. Our framework alternates between the E- and M-step until converging to achieve robust representation and reliable category estimation. After learning, the cluster assignment for an unlabelled instance, either from known or novel classes, can be retrieved by finding the nearest prototypes. Thus we name our framework as **GPC**: **G**aussian mixture model for generalized category discovery with **P**rotypical **C**ontrastive learning.

Our contributions in this paper are as follows: (1) We demonstrate that in generalized category discovery, the class number estimation and representation learning can reinforce each other in the learning process. Strong representations can give a better estimation of the class number, and vice versa. (2) We propose an EM-like framework that alternates between prototype estimation with a variant of GMM (E-step) and representation learning based on prototypical contrastive learning (M-step). (3) We introduce a semi-supervised variant of GMM with a stochastic splitting and merging mechanism to allow dynamic change of the prototypes by examining the cluster compactness and separability based on the Metropolis-Hastings ratio (Hastings, 1970). (4) We comprehensively evaluated our framework on both the generic image classification benchmark, including CIFAR10, CIFAR100, ImageNet-100, and the challenging fine-grained Semantic Shifts Benchmark suite, which includes CUB-200, Stanford-Cars, and FGVC-aircrafts, achieving the state-of-the-art results.

2 Related Work

Novel category discovery (NCD) is first formalized in DTC (Han et al., 2019) where the task is to discover new categories leveraging the knowledge of a set of labelled categories. Earlier methods like MCL (Hsu et al., 2018) and KCL (Hsu et al., 2019) for generalized transfer learning can also be applied to this problem. RankStat (Han et al., 2020; 2021) shows that this task benefits from self-supervised pretraining and proposes a method to transfer knowledge from the labelled data to the unlabelled data using ranking statistics. NCL (Zhong et al., 2021a) and Jia et al. (2021) adopt contrastive learning for novel category discovery. OpenMix (Zhong et al., 2021b) shows that mixing up labelled and unlabelled data can help avoid the representation from overfitting to the labelled categories. Zhao & Han (2021) propose a dual ranking statistics framework to focus on the local visual cues to improve the performance on fine-grained classification benchmarks. UNO (Fini et al., 2021) introduces a unified cross-entropy loss that enables the model to be jointly trained on unlabelled and labelled data. Most recently, generalized category discovery (GCD) is introduced in Vaze et al. (2022a) to extend NCD to a more open-world setting where unlabelled instances can come from both labelled and unlabelled categories. Concurrent work ORCA (Cao et al., 2022) also tackles a similar setting as GCD, termed open world semi-supervised learning. Despite the advance in this setting, most methods still assume that the novel class number is known *a priori* which is often not the case in the real world. To address this problem, DTC (Han et al., 2019) and GCD (Vaze et al., 2022a) precompute the number of novel classes using a semi-supervised k-means algorithm, without considering the representation learning. In this paper, we demonstrate that class number estimation and representation learning can be jointly considered to mutually benefit each other.

Contrastive learning (Chen et al., 2020a;b; He et al., 2020) (CL) has been shown very effective for representation learning in a self-supervised manner, using the instance discrimination pretext (Wu et al., 2018) as the learning objective. The instance discrimination task learns a representation by pulling positive samples from the augmentations of the same images closer and pushing negative samples from different images apart in the embedding space. Instead of contrasting over all instances in a mini-bath, prototypical contrastive learning (PCL) (Li et al., 2020) proposes to contrast the features with a set of prototypes which can provide a higher level abstraction of dataset than instances and has been shown to be more data efficient without the need of large batch size. Though PCL is developed for unsupervised representation learning, if the prototypes are viewed as cluster centers, it can be leveraged in the partially supervised setting of GCD for representation learning to better fit the GCD task of partitioning data into different clusters. Thus, in this paper, we adopt PCL to fit the GCD setting for representation learning in which the downstream clustering task is directly considered.

Semi-supervised learning (SSL) has been a long standing research topic which many effective method proposed (Rebuffi et al., 2020; Sohn et al., 2020; Berthelot et al., 2019; Laine & Aila, 2017; Tarvainen & Valpola, 2017). In SSL, the labelled and the unlabelled data are assumed to come from the same set of classes, and the task is to learn a classification model that can take advantage of both labelled and unlabelled data. Consistency-based methods are among the most effective methods for SSL, such as Mean-teacher (Tarvainen & Valpola, 2017), MixMatch (Berthelot et al., 2019), and FixMatch (Sohn et al., 2020). Self-supervised representation learning also shows to be helpful for SSL because it can provide a strong representation (Zhai et al., 2019; Rebuffi et al., 2020).

Unsupervised clustering has been studied for decades, and there are many existing classical approaches (MacQueen, 1967; Ester et al., 1996; Comaniciu & Meer, 2002) as well as deep learning based approaches (Xie et al., 2016; Rebuffi et al., 2021; Ghasedi Dizaji et al., 2017). Recently, DeepDPM (Ronen et al., 2022) is proposed to automatically determine the number of clusters for a given dataset by adopting a similar split/merge framework that changes the inferred number of clusters. However, due to the unsupervised nature of these methods, there is no prior or supervision over how a cluster should be formed, thus multiple equally valid clustering results following different clustering criteria can be produced. Thus, directly applying unsupervised clustering methods to the task of generalized category discovery is not feasible, as we would want the model to use one unique clustering criteria implicitly given by the labelled data.

3 Method

Given a collection of partially labelled data, $\mathcal{D} = \mathcal{D}^l \cup \mathcal{D}^u$, where $\mathcal{D}^l = \{(x_i, y_i^l)\} \in \mathcal{X} \times \mathcal{Y}_l$ is labelled, $\mathcal{D}^u = \{x_i, y_i^u\} \in \mathcal{X} \times \mathcal{Y}_u$ is unlabelled, and $\mathcal{Y}_l \subset \mathcal{Y}_u$, Generalized category discovery (GCD) aims at automatically assign labels for the unlabelled instances in $\mathcal{D}^u$, by transferring knowledge acquired from $\mathcal{D}^l$. Let the category number in $\mathcal{D}^l$ be $K^l = |\mathcal{Y}_l|$ and that in $\mathcal{D}^u$ be $K^u = |\mathcal{Y}_u|$. The number of new categories $\mathcal{D}^u$ is then $K^n = |\mathcal{Y}_u \setminus \mathcal{Y}_l| = K^u - K^l$. Though K^l can be accessed from the labelled data, we do not assume K^n or K^u to be known. This is a realistic setting to reflect the real open world, where we often have access to some labelled data, but in the unlabelled data, we also have instances from unseen new categories.

The key challenges for GCD are representation learning, category number estimation, and label assignment. Existing methods for NCD and GCD (Han et al., 2019; Vaze et al., 2022a) deal with these three challenges independently. However, we believe they are inherently linked with each other. Label assignment depends on representation and category number estimation. A good class number estimation can facilitate representation learning, thus better label assignment, and vice versa. Thus, in this paper, we aim to jointly handle these challenges in the learning process for a more reliable GCD.

To this end, we propose a unified EM-like framework that alternates between representation learning and class number estimation, while the label assignment turns out to be a by-product during class number estimation. In the E-step, we introduce a semi-supervised variant of the Gaussian Mixture Model (GMM) to estimate the class numbers by dynamically splitting separable clusters and merging cluttered clusters based on current representation, forming a set of class prototypes for both seen and unseen classes, and in the M-step, we train the model to produce discriminative representation by prototypical contrastive learning using the cluster centers from the GMM prototypes derived from the E-step during class number estimation. After training, the class assignment for each instance can be retrieved by simply identifying the nearest prototype.

3.1 Representation learning

The goal of representation learning is to learn a discriminative representation that can well separate different categories, not only the old ones, but also the new ones. Contrastive learning (CL) has been shown to be an effective choice for NCD (Jia et al., 2021) and GCD (Vaze et al., 2022a). Self-supervised contrastive learning is defined as

$$\mathcal{L}_{CL} = -\log \frac{\exp(z_i \cdot z_i'/\tau)}{\sum_{j=1}^{n} \exp(z_i \cdot z_j'/\tau)} \tag{1}$$

where z_i and z_i' are the representations of two views obtained from the same image using random augmentations and τ is the temperature. Two views of the same instance are pulled closer, and different instances are pushed away during training. Self-supervised contrastive learning and its supervised variant, in which different instances from the same category are also pulled closer, are used in Vaze et al. (2022a) for representation learning. However, as a stronger training signal is used for the labelled data, the representation is likely biased to the labelled data to some extent. Moreover, such a method does not take the downstream clustering task into account during learning, thus a clustering algorithm is required to run independently after the representation learning.

In this paper, we adopt prototypical contrastive learning (PCL) (Li et al., 2020) to the GCD setting to learn the representation $z_i = f(x_i) \in \mathbb{R}^d$. PCL uses a set of prototypes $\mathcal{C} = \{\mu_1, \ldots, \mu_K\}$ to represent the dataset for contrastive learning instead of the random augmentation generated views z_i'. PCL loss can be written as

$$\mathcal{L}_{PCL} = -\log \frac{\exp(z_i \cdot \mu_s/\tau)}{\sum_{j=1}^{K} \exp(z_i \cdot \mu_j/\tau)} \tag{2}$$

where μ_s is the corresponding prototype for z_i. It was originally designed as an alternative for self-supervised contrastive learning by over clustering the training data to obtain the prototypes during training. We employ PCL here to learn reliable representation while taking the downstream clustering into account for GCD, where we have a set of partially labelled data. In our case, the prototypes can be interpreted as the class centers for each of the categories. To obtain the prototypes for the seen categories, we directly calculate the class mean by averaging all the feature vectors of the labelled instances. For the unseen categories, we obtain the prototypes with a semi-supervised variant of the Gaussian Mixture Model (GMM), as will be introduced in Section 3.2. This way, the cluster assignment for an unlabelled image can be readily achieved by finding the nearest prototype.

Additionally, we observe that only a few principal dimensions can already recover most of the variances in the representation space of z_i, which is known as *dimensional collapse* (DC) in (Jing et al., 2021; Hua et al., 2021), and it is shown that DC can be caused by strong augmentations or implicit regularizations in the model, and preventing DC during training can lead to a better feature representation. To alleviate DC for representation learning in our case, we propose to first project the feature to a subspace obtained by principal component analysis (PCA) before the contrastive

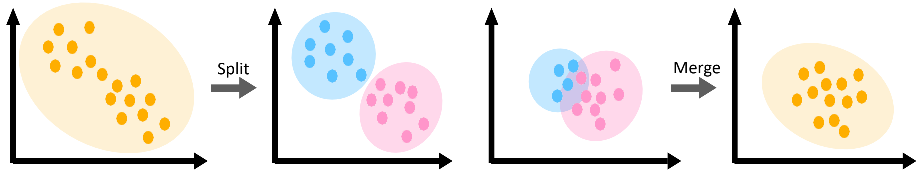


Figure 2: **Examples for splitting a separable cluster and merging two cluttered clusters.** Left: the cluster is split because the two sub-component in this cluster are easily separable. Right: two clusters are merged as they are cluttered and likely from the same class.

learning. Specifically, we apply PCA on a matrix Z formed by a mini-batch of features z_i, with a batch size of n, the feature dimension d, and the number of effective principal directions q. We have $Z \approx Udiag(S)V^\top$, where $U \in \mathbb{R}^{n\times q}$, $S \in \mathbb{R}^q$ and $V \in \mathbb{R}^{d\times q}$. We can then project features z_i to principal directions to obtain a more compact feature $v_i = V \cdot z_i$, and replace feature z_i with v_i in Eq. (2) for PCL. The prototypes are also computed in the projected space.

We jointly use self-supervised contrastive learning and PCL to train our model. The overall learning objective can be written as

$$\mathcal{L} = \mathcal{L}_{CL} + \lambda(t)\mathcal{L}_{PCL} \tag{3}$$

where $\lambda(t)$ is a linear warmup function defined as $\lambda(t) = \min(1, \frac{t}{T})$ where t is the current epoch and T warmup length ($T = 20$ in our experiments). The reason we use both CL and PCL is that, in the beginning, the representation is not well suited for clustering, and thus the obtained prototypes are not informative to facilitate the representation learning. Hence, we gradually increase the weight of PCL during training from 0 to 1 in the first T epochs.

3.2 Class number and prototypes estimation with semi-supervised Gaussian mixture model

In this section, we present a semi-supervised variant of the Gaussian mixture model (GMM) with each Gaussian component consisting of two sub-components to estimate the prototypes for representation learning in Section 3.1 and the unknown class number. GMM estimates the prototypes and assigns a label for each data point by finding its nearest prototype. The cluster label assignment and the prototypes are then used for prototypical contrastive learning. The GMM is defined as

$$p(z) = \sum_{i=1}^{K} \pi_i \mathcal{N}(z|\mu_i, \Sigma_i), \tag{4}$$

where $\mathcal{N}(z|\mu_i, \Sigma_i)$ is the Gaussian probability density function with mean $\mu_i \in \mathbb{R}^d$ and covariance $\Sigma_i \in \mathbb{R}^{d\times d}$, and π_i is the weight for i-th Gaussian component and we have $\sum_{i=1}^{N} \pi_i = 1$. Ideally, we would expect the component number K in the GMM to be equal to the class number K^u in $\mathcal{D}$. To estimate the unknown class number K^u, we leverage an automatic *splitting-and-merging* strategy into the modeling process to obtain an optimal K, which is expected to be as close to K^u as possible. We alternate between representation learning and K^u estimation until convergence to get discriminative representation learning and a reliable class number estimation. For initialization, K can be set to any number greater than K^l. In our experiments, we simply set the initial number of components to a default $K_{init} = K^l + \frac{K^l}{2}$. We run a semi-supervised k-means algorithm (Vaze et al., 2022a) with $k = K$ to obtain the μ and Σ for each component in the mixture model. Note that the semi-supervised k-means algorithm is constrained to the labelled data in a way that labelled instances from the same class are assigned to the same cluster, and labelled instances from different classes will not be assigned to the same cluster. To facilitate the splitting and merging process, for each Gaussian component defined by μ_i and Σ_i, we further depict it with a GMM with two sub-components $\mu_{i,1}, \mu_{i,2}$ and $\Sigma_{i,1}, \Sigma_{i,2}$ with $\pi_{i,1} + \pi_{i,2} = 1$. We run a k-means with $k = 2$ on the i-th component to obtain $\mu_{i,1}, \mu_{i,2}$ and $\Sigma_{i,1}, \Sigma_{i,2}$.

For a cluster whose two sub-components are roughly independent and equally sized (*e.g.*, left part of Fig. 2), *i.e.*, they are easily separable, we would like the model to split it into two such that the model can better fit the data distribution and the class assignment will be more accurate because it is less likely that such distinct clusters will belong to the same class. For two clusters that are cluttered with each other(*e.g.*, right part of Fig. 2), *i.e.*, difficult to distinguish, we would like to merge them into one, so that they will be considered as from the same class. Following this intuition, we use

Algorithm 1: The overall algorithm of our proposed framework.

Input:
$\mathcal{D}$, $\mathcal{D}^l$, and $\mathcal{D}^u$ — The dataset, and the subset for labelled and unlabelled images.
K_{init} — Initial guess of the number of classes.

$K \leftarrow K_{init}$
for $e = 1$ **to** E **do**
  $z \leftarrow f(x), x \in \mathcal{D}$ // extract features
  $\mu, \Sigma \leftarrow \arg\max \sum_{i=1}^{K} \pi_i \mathcal{N}(z|\mu_i, \Sigma_i)$ // estimate prototypes using GMM
  for $i = 1$ **to** $len(\mathcal{D})$ **do**
    $\mathcal{B}^l \leftarrow \{x_i^l \sim \mathcal{D}^l\}_{i=1}^{N^l}$ // sample a batch of N^l labelled images
    $\mathcal{B}^u \leftarrow \{x_i^u \sim \mathcal{D}^u\}_{i=1}^{N^u}$ // sample a batch of N^u unlabelled images
    $f \leftarrow \arg\min \mathcal{L}(f, \mu, \mathcal{B}^l, \mathcal{B}^u)$ // prototypical contrastive learning
  end
  $H_s, H_m \leftarrow$ calc_prob(μ, Σ) // probability for split and merge
  $\mu, \Sigma \leftarrow$ perform_op(H_s, H_m) // perform operations
  $K \leftarrow len(\mu)$ // update K
end
Output: feature extractor $f(\cdot)$, cluster centers μ_i

the Metropolis-Hastings framework (Hastings, 1970) to compute a probability $p_s = \min(1, H_s)$ to stochastically split a cluster into two. The Hastings ratio is defined as

$$H_s = \frac{\Gamma(N_{i,1})h(\mathcal{Z}_{i,1};\theta)\Gamma(N_{i,2})h(\mathcal{Z}_{i,2};\theta)}{\Gamma(N_i)h(\mathcal{Z}_i;\theta)}, \tag{5}$$

where Γ is the factorial function, *i.e.*, $\Gamma(n) = n! = n \times (n-1) \times \cdots \times 1$, $\mathcal{Z}_i$ is the set of data points in cluster i, $\mathcal{Z}_{i,j}$ is the set of data points in the j-th sub-cluster of cluster i, $N_i = |\mathcal{Z}_i|$, $N_{i,j} = |\mathcal{Z}_{i,j}|$, $h(Z;\theta)$ is the marginal likelihood of the observed data $\mathcal{Z}$ by integrating out the μ and Σ parameters in the Gaussian, and θ is the prior distribution of μ and Σ. More details can be found in the supplementary. The intuition behind this H_s is that, if the number of data points in two sub-components is roughly balanced, which is measured by the $\Gamma(\cdot)$ terms, and the data points in the two sub-components are independent of each other, which is measure by the $h(\cdot;\theta)$ terms, there should be a greater chance of splitting the cluster. After performing a split operation, the μ_i and Σ_i of previous components i will be replaced with $\mu_{i,1}, \mu_{i,2}$ and $\Sigma_{i,1}, \Sigma_{i,2}$ of two sub-components. We will then run two k-means within the two newly formed components to obtain their corresponding sub-components. On the contrary, if two clusters are cluttered with each other, they should be merged. Similar to splitting, we determine the merging probability by $p_m = \min(1, H_m)$, where $H_m = \frac{1}{H_s}$. Note that both H_s and H_m are within the range of $(0, +\infty)$, so we use $p_s = min(1, H_s)$ and $p_m = min(1, H_m)$ to convert it into a valid probability.

To take the labelled instances into consideration during the splitting-and-merging process, if a cluster consists of labelled instances, we set its $p_s = 0$; if for any two clusters containing instances from two labelled classes, we set their $p_m = 0$.

During the splitting-and-merging process, we first apply splitting according to the p_s and then apply merging according to p_m. The newly formed clusters by splitting will not be reused during the merging step. After finishing the splitting and merging, we can obtain the prototypes, and thus can estimate K, for our PCL-based representation learning. We alternate between representation learning and class number estimation for each training epoch until converge. The final K will be considered the estimated class number in $\mathcal{D}$. The cluster assignment for each unlabelled instance can be easily retrieved by identifying its nearest prototype, without the need of running a non-parametric clustering algorithm as Vaze et al. (2022a). The overall training process is summarized in Algorithm 1.

4 Experimental results

4.1 Experimental setup

Benchmark and evaluation metric. We validate the effectiveness of our method on the generic image classification benchmark (including CIFAR-10/100 (Krizhevsky & Hinton, 2009) and ImageNet-

100 (Tian et al., 2020)) and also the recently proposed Semantic Shift Benchmark (Vaze et al., 2022b) (SSB)(including CUB-200 (Wah et al., 2010), Stanford Cars (Krause et al., 2013), and FGVC-Aircraft (Maji et al., 2013)). For each of the datasets, we follow Vaze et al. (2022a) and sample a subset of all classes for which we have annotated labels during training. For experiments on SSB datasets, we directly use the class split from Vaze et al. (2022b). 50% of the images from these labelled classes will be used as the labelled instances in $\mathcal{D}^l$, and the remaining images are regarded as the unlabelled data $\mathcal{D}^u$ containing instances from labelled and unlabelled classes. See Table 1 for statistics of the datasets we evaluated. We evaluate the model performance with clustering accuracy (ACC) following standard practice in the literature. At test-time, given the ground truth labels y^* and the model predicted cluster assignments $\hat{y}$, the ACC is calculated as $ACC = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}(y_i^* = g(\hat{y}_i))$ where g is the optimal permutation for matching the predicted cluster assignment $\hat{y}$ to the actual class label y_i^* and $M = |\mathcal{D}^u|$.

Implementation details. We train and test all the methods with a ViT-B/16 backbone (Dosovitskiy et al., 2021) with pretrained weights from DINO (Caron et al., 2021). We use the output of `[CLS]` token with a dimension of 768 as the feature representation for an input image. We only finetune the last block of the ViT-B backbone to prevent the model from overfitting to the labelled classes during training. We set the batch size for training the model to 128 with 64 labelled images and 64 unlabelled images and use a cosine annealing schedule for the learning rate starting from 0.1. The number of principle directions in the PCA is set to 128, which we found performs the best across all the datasets evaluated. We train all the methods for 200 epochs on each dataset for a fair comparison with previous works, and the best performing model is selected using the accuracy on the validation set of the labelled classes. All experiments are done with an NVIDIA V100 GPU with 32GB memory.

Table 1: **Data splits in the experiments.**

	labelled	unlabelled
CIFAR-10	5	5
CIFAR-100	80	20
ImageNet-100	50	50
CUB-200	100	100
Stanford-Cars	98	98
FGVC-aircraft	50	52

4.2 COMPARISON WITH THE STATE-OF-THE-ART

In Table 2, we report the comparison with the-state-of-the-art method of Vaze et al. (2022a), strong baselines derived from NCD methods, and the k-means on the generic classification datasets. Notably, our method consistently achieves the best overall performance on all datasets, under the challenging setting where the class number is unknown. When the class number is known, our method also achieves the best performance on all datasets, except ImageNet-100, on which the best performance is achieved by ORCA (Cao et al., 2022). In rows 1-7 we compare with other methods with the known class number in the unlabelled data, while in rows 8-11 we compare with Vaze et al. (2022a) for the case of unknown class number. We can see that our proposed framework outperforms other methods in most cases and especially when the number of classes is unknown. Comparing rows 10 and 11 to row 5, we can see that our proposed method without knowing the number of classes can even matches the performance of previous strong baseline with the number of classes known to the model. Furthermore, from row 6 vs row 7 and row 10 vs row 11, we can see that the additional PCA layer can effectively improves the performance, also the performance improvement from PCA are larger on the 'New' classes than on the 'Old' classes, which validates that the PCA can keep the representation space from collapsing and improve the performance on classes without using any labels. Due to fact that the labelled instances provide a stronger training signal, we can see from rows 6 - 11 that the performance on 'Old' classes is generally steady.

Table 3 shows the performance comparison on the more challenging Semantic Shift Benchmark (Vaze et al., 2022b). A similar trend of Table 2 holds true for the results on SSB. Our approach achieves competitive performance in all cases and again reaches a better performance when the number of classes is unknown.

4.3 NOVEL CLASS NUMBER ESTIMATION

One of the important yet overlooked components in the NCD and GCD literature is the estimation of unknown class numbers. Our proposed framework leverages a modified GMM to estimate the class number, in which we need to define a initial guess of the class number. We validate the effects of

Table 2: **Results on generic image classification datasets.**

No.	Methods	Known K	PCA	CIFAR10 All	CIFAR10 Old	CIFAR10 New	CIFAR100 All	CIFAR100 Old	CIFAR100 New	ImageNet-100 All	ImageNet-100 Old	ImageNet-100 New
(1)	k-means (MacQueen, 1967)	✓	✗	83.6	85.7	82.5	52.0	53.7	51.1	73.4	75.5	71.3
(2)	RankStats+ (Han et al., 2021)	✓	✗	84.5	96.4	78.5	65.0	78.8	37.4	50.9	94.2	29.2
(3)	UNO+ (Fini et al., 2021)	✓	✗	72.4	95.0	61.1	64.1	73.8	44.8	62.3	94.7	46.0
(4)	ORCA (Cao et al., 2022)	✓	✗	91.4	88.0	91.2	68.9	76.1	46.6	79.8	93.6	74.9
(5)	Vaze et al. (2022a)	✓	✗	91.5	97.9	88.2	76.9	84.5	61.7	75.1	92.2	66.5
(6)	*Ours* (GPC)	✓	✗	92.0	98.3	88.7	77.4	84.8	62.4	76.5	94.0	68.5
(7)	*Ours* (GPC)	✓	✓	92.2	98.2	89.1	77.9	85.0	63.0	76.9	94.3	71.0
(8)	Vaze et al. (2022a)	✗	✗	88.6	96.2	84.9	73.2	83.5	57.9	72.7	91.8	63.8
(9)	Vaze et al. (2022a)	✗	✓	89.7	97.3	86.3	74.8	83.8	58.7	73.8	92.1	64.6
(10)	*Ours* (GPC)	✗	✗	88.2	97.0	85.8	74.9	84.3	59.6	74.7	92.9	65.1
(11)	*Ours* (GPC)	✗	✓	**90.6**	**97.6**	**87.0**	**75.4**	**84.6**	**60.1**	**75.3**	**93.4**	**66.7**

Table 3: **Results on Semantic Shift Benchmark datasets.**

No.	Methods	Known K	PCA	CUB All	CUB Old	CUB New	Stanford Cars All	Stanford Cars Old	Stanford Cars New	FGVC-aircraft All	FGVC-aircraft Old	FGVC-aircraft New
(1)	k-means (MacQueen, 1967)	✓	✗	34.3	38.9	32.1	12.8	10.6	13.8	16.0	14.4	16.8
(2)	RankStats+ (Han et al., 2021)	✓	✗	33.3	51.6	24.2	28.3	61.8	12.1	26.9	36.4	22.2
(3)	UNO+ (Fini et al., 2021)	✓	✗	35.1	49.0	28.1	35.5	70.5	18.6	40.3	56.4	32.2
(4)	ORCA (Cao et al., 2022)	✓	✗	45.2	57.2	29.7	37.0	68.2	22.6	47.1	45.3	42.3
(5)	Vaze et al. (2022a)	✓	✗	51.3	56.6	48.7	39.0	57.6	29.9	45.0	41.1	46.9
(6)	*Ours* (GPC)	✓	✗	54.2	54.9	50.3	41.2	58.8	31.6	46.1	42.4	47.2
(7)	*Ours* (GPC)	✓	✓	55.4	58.2	53.1	42.8	59.2	32.8	46.3	42.5	47.9
(8)	Vaze et al. (2022a)	✗	✗	47.1	55.1	44.8	35.0	56.0	24.8	40.1	40.8	42.8
(9)	Vaze et al. (2022a)	✗	✓	49.2	**56.2**	46.3	36.3	56.6	25.9	41.2	**40.9**	44.6
(10)	*Ours* (GPC)	✗	✗	50.2	52.8	45.6	37.2	56.3	26.3	39.7	39.6	42.7
(11)	*Ours* (GPC)	✗	✓	**52.0**	55.5	**47.5**	**38.2**	**58.9**	**27.4**	**43.3**	40.7	**44.8**

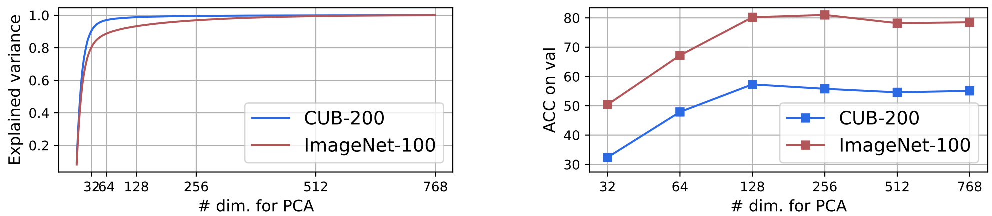
different choices of the initial guess K_{init} w.r.t. the estimated class number in Table 4. Note that the number in Table 4 is $K^n_{init} = K_{init} - K^l$. We can see that our proposed framework is generally robust to a wide range of initial guesses. We found that $K_{init} = K^l + \frac{K^l}{2}$ is a simple and reliable choice. Hence we use this for all datasets.

Table 4: **Results of varying the initial guessed K^n_{init}.** 'GT K^n' is the ground truth number of novel classes. K^n is the estimated number of novel classes.

Dataset	K^l	GT K^n	Vaze et al. (2022a)	$K^n_{init} =$	3	5	10	20	30	50	100
CIFAR-10	5	5	4	$K^n =$	5	5	5	6	6	8	14
CIFAR-100	80	20	20	$K^n =$	16	20	20	21	22	27	36
ImageNet-100	50	50	59	$K^n =$	58	48	57	55	54	50	60
CUB	100	100	131	$K^n =$	79	87	86	88	92	112	101
SCars	98	98	132	$K^n =$	84	90	86	87	89	115	104

4.4 ABLATION STUDY

Number of dimensions in PCA The PCA in our framework requires setting a number for the number of principle directions to extract from data. In Fig. 3 we show the results of using a different number of principle directions in PCA on CUB-200 and ImageNet-100 datasets. We can see from Fig. 3a that for both datasets, 128 principle directions can already explain most of the variances in the data, thus we choose the PCA dimension to be 128 for all our experiments. We further experiment with other different choices of the PCA dimension and shows the result in Fig. 3b, which again confirms that 128 principle directions are already expensive enough, and obtain the best performance over other choices, that are either too few or too many, effectively avoiding the DC.



(a) Explained variance of the original feature w.r.t. the number of principle directions.

(b) The clustering ACC on validation set w.r.t. the number of principle directions

Figure 3: **The effects of the number of principle directions in PCA on the feature representations.**

Different methods for prototype estimation Our semi-supervised GMM plays an important role in prototype estimation for representation learning based on prototypical contrastive learning. Here, we replace our semi-supervised GMM with other alternatives that do not produce prototypes automatically. Particularly, we compare our method with DBSCAN (Ester et al., 1996), Agglomerative clustering (Murtagh & Legendre, 2014), and semi-supervised k-means (Han et al., 2019; Vaze et al., 2022a). The prototypes are then obtained by averaging the data points that are assigned to the same cluster. For a fair comparison, the same regulations to prevent the wrong clustering results for labelled instances are applied to all methods, *i.e.*, during the clustering process, two labelled instances with the same label will fall into the same cluster, and two instances with different labels will be assigned to different clusters. The results are reported in Table 5a. Our method achieves the best performance on all three datasets, indicating that better prototypes are obtained by our approach to facilitate representation learning. Note that DBSCAN requires two important user-defined parameters, radius, and minimum core points, the ideal values of which lack a principled way to obtain in practice, while our method is parameter-free and can seamlessly be combined with the representation learning to jointly enhance each other, obtaining better performance.

Table 5: **Combining components of GPC with other methods.** "IM-100" denotes ImageNet-100.

(a) Different prototype estimation methods.

Clustering Algo.	CUB	IM-100
Ester et al. (1996)	45.6	66.1
Murtagh & Legendre (2014)	52.1	74.6
Vaze et al. (2022a)	49.2	73.2
Ours (GPC)	54.1	76.6

(b) Combining our GMM with other methods.

Representation	CUB	IM-100
Han et al. (2021)	34.6	38.4
Zhao & Han (2021)	37.8	39.7
Vaze et al. (2022a)	50.6	73.4
Ours (GPC)	54.1	76.6

Combining our GMM with other GCD methods We further combine our semi-supervised GMM with automatic splitting and merging with other methods, allowing joint representation learning and category discovery without a predefined category number. As the state-of-the-art GCD method (Vaze et al., 2022a) does not contain any parametric classifier during representation, so it can be directly combined with our GMM. For the RankStat and the DualRank methods that have a parametric classifier for category discovery, we treat the weights of the classifier as the cluster centers and run our GMM to automatically determine the category number during representation learning. The results are presented in Table 5b. Comparing with row 9 in Table 2 and Table 3, we can see using our GMM can also improve Vaze et al. (2022a) on CUB and Stanford Cars, while our proposed framework consistently achieves better performance on all datasets, again validating that our design choices.

5 Conclusion

In this paper, we present an EM-like framework for the challenging GCD problem without knowing the number of new classes, with the E-step automatically determining the class number and prototypes and the M-step being robust representation learning. We introduce a semi-supervised variant of GMM with a stochastic splitting and merging mechanism to obtain the prototypes and leverage these evolving prototypes for representation learning by prototypical contrastive learning. We demonstrated that class number estimation and representation learning can facilitate each other for more robust category discovery. Our framework obtains state-of-the-art performance on multiple public benchmarks.

REFERENCES

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 1, 3

Kaidi Cao, Maria Brbić, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022. 3, 7, 8, 15, 16

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 7, 15, 16

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020a. 3

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b. 3

Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 2002. 3

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 17

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 *ICLR*, 2021. 7

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. 3, 9

Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *ICCV*, 2021. 1, 3, 8

Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV*, 2017. 3

Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *ICCV*, 2019. 1, 2, 3, 4, 9

Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020. 3

Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *IEEE TPAMI*, 2021. 1, 3, 8, 9, 14

Wilfred Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 1970. 2, 6

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 3

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018. 3

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *ICLR*, 2019. 3

Tianyu Hua, Wenxiao Wang, Zihui Xue, Yue Wang, Sucheng Ren, and Hang Zhao. On feature decorrelation in self-supervised learning. In *ICCV*, 2021. 4

Xuihui Jia, Kai Han, Yukun Zhu, and Bradley Green. Joint representation learning and novel category discovery on single-and multi-modal data. In *ICCV*, 2021. 1, 3, 4

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *ICLR*, 2021. 4

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 7, 17

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 6, 17

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 3

Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2020. 2, 3, 4

James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967. 3, 8

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 7, 17

Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 2014. 9

Avital Oliver, Augustus Odena, Colin Raffel, Ekin D Cubuk, and Ian J Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018. 1

Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Semi-supervised learning with scarce annotations. In *CVPR Deep-Vision workshop*, 2020. 3

Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Lsd-c: Linearly separable deep clusters. In *ICCV Workshop on Visual Inductive Priors for Data-Efficient Deep Learning*, 2021. 3

Meitar Ronen, Shahaf E. Finder, and Oren Freifeld. Deepdpm: Deep clustering with an unknown number of clusters. In *CVPR*, 2022. 3

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 1, 3

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 1, 3

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 7

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *CVPR*, 2022a. 1, 3, 4, 5, 6, 7, 8, 9, 14, 15, 16

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? In *ICLR*, 2022b. 7, 16

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. Caltech-UCSD Birds 200. *Computation & Neural Systems Technical Report*, 2010. 7, 17

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016. 3

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, 2019. 3

Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In *NeurIPS*, 2021. 1, 3, 9, 14

Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021a. 1, 3

Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *CVPR*, 2021b. 1, 3

A DETAILS OF THE FRAMEWORK FOR SPLITTING AND MERGING CLUSTERS

In this section, we provide details for the Metropolis-Hastings framework. In the Gaussian Mixture Model, we have three sets of parameters, (π_i, μ_i, Σ_i) where π_i is the mixture weight and μ_i, Σ_i are the mean and covariance matrix. These parameters are assumed to be sampled from a prior distribution. When μ_i and Σ_i are unknown for the multivariate Gaussian distribution, we adopt the Normal Inverse Wishart (NIW) distribution as the prior to sample them for algebraic convenience, because NIW distribution is a conjugate prior and the conjugacy property can lead to a closed-form expression of the posterior.

The Inverse Wishart (IW) distribution is defined as follows:

$$p(\Sigma_i) \sim \mathcal{W}^{-1}(\nu, \Psi) = \frac{|\nu\Psi|^{\frac{\nu}{2}}}{2^{\frac{\nu d}{2}}\Gamma_d(\frac{\nu}{2})}|\Sigma_i|^{-\frac{\nu+d+1}{2}} \exp(-\frac{1}{2}tr(\nu\Psi\Sigma_i^{-1})), \tag{6}$$

where Σ_i is a $d \times d$ Symmetric and Positive Definite(SPD) matrix, $\nu > d-1$, $\Psi \in \mathbb{R}^{d\times d}$ is SPD, and Γ_d is a d-dimensional multivariate factorial function. The positive real number ν and the SPD matrix Ψ are the parameters of the IW distribution. The data distribution determined by μ_i and Σ_i follows NIW distribution, if the joint probability density function is defined by

$$p(\mu_i, \Sigma_i) \sim \text{NIW}(\kappa, \mathbf{m}, \nu, \Psi) \triangleq \mathcal{N}(\mu_i; \mathbf{m}, \frac{1}{\kappa}\Sigma_i)\mathcal{W}^{-1}(\Sigma_i; \nu, \Psi), \tag{7}$$

where $\mathbf{m} \in \mathbb{R}^d$, $\kappa > 0$, and $\mathcal{N}(\mu_i; \mathbf{m}, \frac{1}{\kappa}\Sigma_i)$ is a d-dimensional Gaussian with mean $\mathbf{m}$ and covariance $\frac{1}{\kappa}\Sigma_i$ evaluated at μ_i.

Given a set of features $\mathcal{Z}_i$ (with $N_i = |\mathcal{Z}_i|$) assigned to the Gaussian component μ_i, Σ_i, we can have a posterior distribution of μ_i, Σ_i in a closed-form thanks to the conjugacy:

$$p(\mu_i, \Sigma_i|\mathcal{Z}_i) = \text{NIW}(\mu_i, \Sigma_i; \kappa^*, \mathbf{m}_i^*, \nu^*, \Psi_i^*), \tag{8}$$

where the posterior parameters are obtained by:

$$\kappa_i^* = \kappa + N_i \tag{9}$$

$$\mathbf{m}_i^* = \frac{1}{\kappa_i^*}[\kappa\mathbf{m} + \sum_{z_k\in\mathcal{Z}} z_k] \tag{10}$$

$$\nu_i^* = \nu + N_i \tag{11}$$

$$\Psi_i^* = \frac{1}{\nu^*}[\nu\Psi + \kappa\mathbf{m}\mathbf{m}^\top + (\sum_{z_k\in\mathcal{Z}_i} z_k z_k^\top) - \kappa_i^*\mathbf{m}_i^*\mathbf{m}_i^{*\top}] \tag{12}$$

In Eq.5 of the main paper, we need to calculate the marginal likelihood function of the observed data $\mathcal{Z}_i$ by integrating out the μ_i and Σ_i parameters in the Gaussian. Let $\theta = (\mathbf{m}, \kappa, \Psi, \nu)$ be the parameters of the NIW distribution. The marginal likelihood can be defined as follows:

$$h(\mathcal{Z}_i; \theta) = \int p(\mathcal{Z}_i|\mu_i, \Sigma_i)p(\mu_i, \Sigma_i; \theta)d(\mu_i, \Sigma_i) \tag{13}$$

$$= \frac{1}{\pi^{Nd/2}}\frac{\Gamma_d(\nu^*/2)}{\Gamma_d(\nu/2)}\frac{|\nu\Psi|^{\nu/2}}{|\nu^*\Psi_i^*|^{\nu^*/2}}\frac{\kappa^{d/2}}{\kappa^{*d/2}}, \tag{14}$$

with which we can compute the Eq. 5 in the main paper.

B ESTIMATING THE NUMBER OF CLUSTERS ON VALIDATION SET

In this section, we validate the choice of K^n_{init} using only the labelled data, to better reflect the real world use case. In particular, we further split the classes in the labelled data $\mathcal{D}^l$ into two parts, $\mathcal{D}^l_r$ and $\mathcal{D}^l_p$. We drop the labels in $\mathcal{D}^l_p$. We verify the effectiveness of different choice of K^n_{init} on $\mathcal{D}^l_p$ and report the results in Table 6. It can be observed that the overall best initial guess of K^n is still around $\frac{K^l}{2}$.

Table 6: **Results of varying the initial guessed K^n_{init}.** 'GT K^n' is the ground truth number of novel classes, splited from the labelled set. K^n is the estimated number of novel classes.

Dataset	K^l	GT K^n	$K^n_{init} =$ 1	3	5	10	20	25	50
CIFAR-10	3	2	$K^n =$ 2	2	4	5	3	6	8
CIFAR-100	60	20	$K^n =$ 15	18	18	20	21	22	29
ImageNet-100	25	25	$K^n =$ 18	19	22	21	23	27	29
CUB	50	50	$K^n =$ 38	37	41	46	49	52	50
SCars	49	49	$K^n =$ 39	38	40	42	43	48	51

C CLASS NUMBER ESTIMATION WITH DIFFERENT REPRESENTATIONS

Here, we validate our class number estimation method on top of the representations learned by other GCD approaches and report the results in Table 7. It can be seen, applying our method on other GCD representations can achieve reasonably well results. Notably, applying our class number estimation method on top of the representation by the existing state-of-the-art method, we can obtain better class number estimation results, though the overall best results are obtained with the representation learned in our framework.

Table 7: **Class number estimation with different learned representations.**

Representation	CIFAR-10	CIFAR-100	CUB	SCars	IM-100
Ground Truth K^n	5	20	100	98	50
Vaze et al. (2022a)	4	20	131	132	59
Ours w/ Han et al. (2021) feat.	5	19	111	94	55
Ours w/ Zhao & Han (2021) feat.	4	22	116	89	49
Ours w/ Vaze et al. (2022a) feat.	5	21	121	109	57
Ours (GPC)	5	20	112	103	53

D ERROR BARS FOR GENERALIZED CATEGORY DISCOVERY PERFORMANCE

We repeatedly run our method and the previous state-of-the-art three times with different random seeds to show the mean and standard deviation values in Table 8 and Table 9, for both known and unknown class number cases. We can see that the variation is relatively small for all methods, and our method consistently outperforms the previous state-of-the-art across the board for both known and unknown class number cases.

Table 8: **Results on generic image classification datasets.**

				CIFAR10			CIFAR100			ImageNet-100		
No.	Methods	Known K	PCA	All	Old	New	All	Old	New	All	Old	New
(1)	Vaze et al. (2022a)	✓	✗	91.5±0.4	97.9±0.2	88.2±0.6	76.9±0.3	84.6±0.3	61.5±0.2	75.0±0.3	92.1±0.2	66.6±0.4
(2)	*Ours* (GPC)	✓	✗	91.9±0.2	98.2±0.3	88.6±0.1	77.6±0.4	84.9±0.4	62.7±0.4	76.7±0.4	94.3±0.2	68.8±0.3
(3)	*Ours* (GPC)	✓	✓	**91.9±0.4**	**98.2±0.3**	**89.1±0.2**	**77.8±0.3**	**85.3±0.2**	**63.5±0.2**	**77.3±0.4**	**94.6±0.4**	**71.1±0.3**
(4)	Vaze et al. (2022a)	✗	✗	88.6±0.5	96.2±0.4	84.9±0.6	73.2±0.4	83.5±0.4	57.9±0.4	72.7±0.4	91.8±0.5	63.8±0.6
(5)	Vaze et al. (2022a)	✗	✓	89.7±0.4	97.3±0.5	86.3±0.4	74.8±0.5	83.8±0.4	58.7±0.6	73.8±0.4	92.1±0.5	64.6±0.6
(6)	*Ours* (GPC)	✗	✗	88.2±0.4	97.0±0.5	85.9±0.3	75.1±0.5	84.4±0.4	59.9±0.6	74.9±0.5	93.2±0.4	65.5±0.3
(7)	*Ours* (GPC)	✗	✓	**90.6±0.3**	**98.2±0.4**	**87.1±0.4**	**75.7±0.5**	**84.7±0.6**	**60.9±0.4**	**75.7±0.3**	**93.4±0.4**	**66.8±0.5**

Table 9: **Results on Semantic Shift Benchmark datasets.**

				CUB			Stanford Cars			FGVC-aircraft		
No.	Methods	Known K	PCA	All	Old	New	All	Old	New	All	Old	New
(1)	Vaze et al. (2022a)	✓	✗	51.1±0.2	56.4±0.1	48.4±0.3	39.1±0.3	57.6±0.4	29.9±0.3	45.1±0.2	41.2±0.3	46.8±0.2
(2)	*Ours* (GPC)	✓	✗	54.5±0.2	54.6±0.4	50.3±0.2	42.0±0.2	58.9±0.2	32.0±0.3	46.3±0.2	42.3±0.2	47.1±0.3
(3)	*Ours* (GPC)	✓	✓	**55.3±0.4**	**58.1±0.3**	**53.2±0.4**	**42.7±0.3**	**60.0±0.4**	**33.0±0.2**	**46.5±0.3**	**42.8±0.5**	**47.2±0.1**
(4)	Vaze et al. (2022a)	✗	✗	47.2±0.4	55.1±0.3	44.8±0.2	35.0±0.3	56.0±0.4	24.8±0.3	40.1±0.2	40.8±0.4	42.8±0.1
(5)	Vaze et al. (2022a)	✗	✓	49.2±0.3	**56.2±0.2**	**46.3±0.4**	36.3±0.3	56.6±0.4	25.9±0.5	41.2±0.3	**40.9±0.4**	44.6±0.2
(6)	*Ours* (GPC)	✗	✗	50.5±0.3	52.5±0.4	45.8±0.5	37.0±0.6	56.6±0.3	26.1±0.2	39.8±0.3	39.7±0.2	42.5±0.2
(7)	*Ours* (GPC)	✗	✓	**52.1±0.3**	55.4±0.2	45.7±0.3	**38.9±0.4**	**58.9±0.3**	**28.6±0.5**	**43.4±0.3**	40.8±0.4	**44.7±0.3**

Table 10: Estimated category numbers

Estimated K^n	CIFAR-10	CIFAR-100	ImageNet-100	CUB-200	Stanford-Cars
Ours (GPC)	5±1	22±2	54±3	110±4	104±3
Vaze et al. (2022a)	4±1	23±1	59±4	131±5	132±2
Ground Truth	5	20	50	100	98

E Error bars for category number estimation

In this section, we show the standard deviations on the estimated category numbers by repeatedly run our method with different random seeds. The results are shown in Table 10. We can see that our method can estimate a more accurate category number with less variances comparing to Vaze et al. (2022a).

F Estimated number of category during training

In Fig. 4, we present the curve of number of categories during the training process on CUB and ImageNet-100. We can see that the estimated category number gradually approaches the ground truth and gets stable towards the end of training, showing that along with the training, the estimated category number is getting more and more accurate.

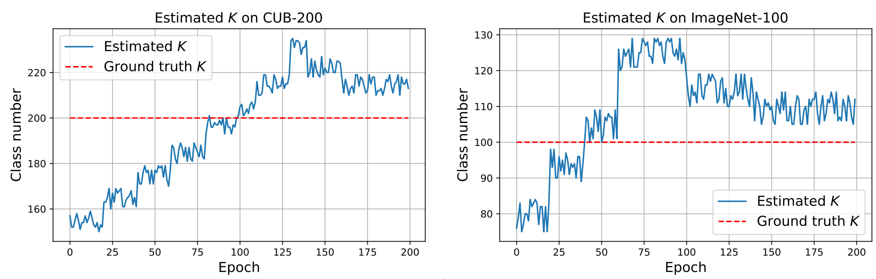


Figure 4: The estimated number of classes during the training process.

G Further comparison with ORCA

ORCA (Cao et al., 2022) is originally pretrained only on the target dataset $\mathcal{D}$, *i.e.*, the data that our model is trained on. We have shown the comparison using ImageNet pretrained features from DINO (Caron et al., 2021) for both ORCA (Cao et al., 2022) and our method in the main paper. In Table 11 and Table 12, we provide additional comparison with ORCA, showing the effects of pretrained models using different data.

Table 11: **Comparison with ORCA (Cao et al., 2022) on generic classification datasets.**

			CIFAR10			CIFAR100			ImageNet-100		
No.	Methods	Pretrain	All	Old	New	All	Old	New	All	Old	New
(1)	ORCA (Cao et al., 2022)	ImageNet	91.4	88.0	**91.2**	68.9	76.1	46.6	**79.8**	93.6	**74.9**
(2)	*Ours* (GPC)	ImageNet	**92.0**	**98.3**	88.7	**77.4**	**84.8**	**62.4**	76.5	**94.0**	68.5
(3)	ORCA (Cao et al., 2022)	Target	90.6	87.2	90.1	64.7	73.2	42.1	78.7	**93.4**	72.4
(4)	*Ours* (GPC)	Target	**91.1**	**87.8**	**90.5**	**65.0**	**74.3**	**42.6**	**79.6**	93.3	**73.1**

H Qualitative results

In this section, we provide the visualization of the images grouped using the DINO features and our GPC trained features. The results are presented in each row of Fig. 5 and Fig. 6. The DINO features are effective to some extent when grouping images, but the results are still not satisfactory as the features are not tuned on the downstream tasks with a clear objective (see Fig. 5). On the other hand, after tuning the representation using our method, images from the same category can be grouped together (see Fig. 6).

We also present t-SNE projections of the learned features on both CUB-200 and ImageNet-100 dataset. From Fig. 7 we can see that on CUB-200, the DINO features can not separate different categories

Table 12: **Comparison with ORCA (Cao et al., 2022) on SSB (Vaze et al., 2022b).**

			CUB			SCars			FGVC-Aircraft		
No.	Methods	Pretrain	All	Old	New	All	Old	New	All	Old	New
(1)	ORCA (Cao et al., 2022)	ImageNet	45.2	**57.2**	29.7	37.0	**68.2**	22.6	**47.1**	**45.3**	42.3
(2)	*Ours* (GPC)	ImageNet	**54.2**	54.9	**50.3**	**41.2**	58.8	**31.6**	46.1	42.4	**47.2**
(3)	ORCA (Cao et al., 2022)	Target	42.9	52.0	28.4	40.3	57.0	31.4	44.4	40.7	44.1
(4)	*Ours* (GPC)	Target	**45.0**	**54.2**	**29.1**	**41.2**	**57.1**	**32.1**	**46.2**	**41.0**	**45.2**

Figure 5: k-means grouping of features of DINO (Caron et al., 2021) on CUB-200 dataset. Notice that the grouping are roughly based on object pose or background, but we would want the clustering to be done to discriminate between different species. The kNN images to the randomly picked prototype (*i.e.*, cluster center) are shown, from left (nearest) to right (furthest).

Figure 6: The prototype (*i.e.*, the Gaussian mean vector in our method) and the retrieved nearest neighbor on GPC representations in the CUB-200 dataset. Images are grouped by different bird species. The kNN images to the randomly picked prototype (*i.e.*, cluster center) are shown, from left (nearest) to right (furthest).

very well, while our method and Vaze et al. (2022a) can have a clear category boundary. From Fig. 8, we found that although the t-SNE projections on ImageNet-100 appear to be similarly discriminative among DINO, Vaze et al. (2022a), and our method, while further finetuning the representation from DINO can significantly improve the performance for the task of GCD.

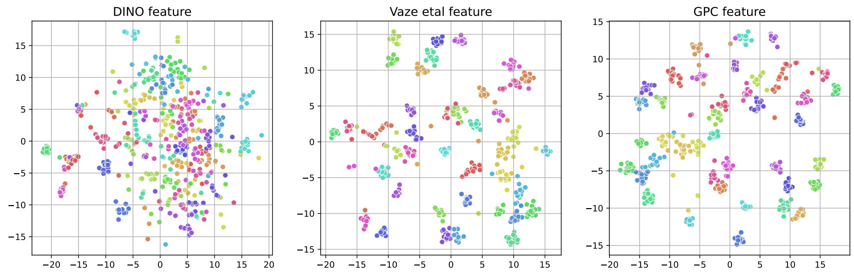


Figure 7: The t-SNE plot of the features on the CUB-200 dataset.

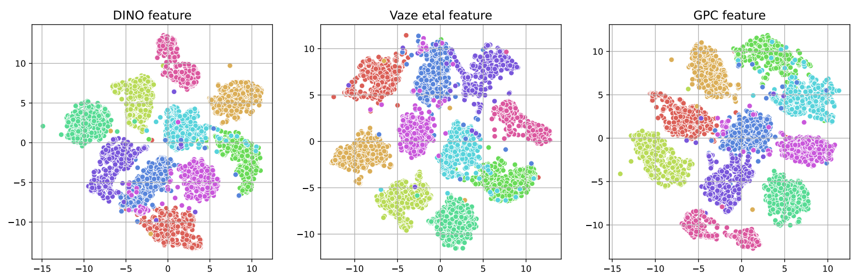


Figure 8: The t-SNE plot of the features on the ImageNet-100 dataset.

I LIMITATION AND NEGATIVE SOCIETAL IMPACT

It should noted that although our method achieves the state-of-the-art results on the task of generalized category discovery, the classification performance is still far from those models trained with full human supervision. Furthermore, when the class number is unknown, there is still a noticeable performance gap w.r.t. the unknown category number case. Besides, real-world data is much more complex and difficult than the curated data we used. Therefore, careful validation and adaptation to specific application scenarios should be tested before deploying the model for any real-world use.

J LICENSE OF USED DATASETS

All the datasets used in this paper are permitted for research use. CIFAR-10 and CIFAR-100 datasets (Krizhevsky & Hinton, 2009) are released under the MIT license, allowing use for research purposes. The terms of access of the ImageNet dataset (Deng et al., 2009) allow the use for non-commercial research and educational purposes. Similar to ImageNet, the Stanford Cars (Krause et al., 2013) allows the use for research purposes. The FGVC aircraft (Maji et al., 2013) dataset was made available exclusively for non-commercial research purposes by the authors. The CUB-200 (Wah et al., 2010) dataset also allows research use.