PromptCCD: Learning Gaussian Mixture Prompt Pool for Continual Category Discovery

Fernando Julio Cendra[1], Bingchen Zhao[2], and Kai Han[1†]

[1]The University of Hong Kong [2]The University of Edinburgh

Abstract. We tackle the problem of Continual Category Discovery (CCD), which aims to automatically discover novel categories in a continuous stream of unlabeled data while mitigating the challenge of catastrophic forgetting – an open problem that persists even in conventional, fully supervised continual learning. To address this challenge, we propose PromptCCD, a simple yet effective framework that utilizes a Gaussian Mixture Model (GMM) as a prompting method for CCD. At the core of PromptCCD lies the Gaussian Mixture Prompting (GMP) module, which acts as a dynamic pool that updates over time to facilitate representation learning and prevent forgetting during category discovery. Moreover, GMP enables on-the-fly estimation of category numbers, allowing PromptCCD to discover categories in unlabeled data without prior knowledge of the category numbers. We extend the standard evaluation metric for Generalized Category Discovery (GCD) to CCD and benchmark state-of-the-art methods on diverse public datasets. PromptCCD significantly outperforms existing methods, demonstrating its effectiveness. Project page: `https://visual-ai.github.io/promptccd`.

Keywords: Continual Category Discovery · Prompt Learning

1 Introduction

Deep learning models have achieved impressive performance in numerous computer vision tasks. However, the majority of these tasks have traditionally been conducted in a closed-world setting, where the models handle known categories and predefined scenarios. The crux lies in developing systems that can effectively and efficiently operate in the real,open-world we inhabit.

Category discovery, initially studied as Novel Class Discovery (NCD) [17] and subsequently extended to Generalized Category Discovery (GCD) [46], has recently emerged as an important open-world research problem, attracting increasing attention and efforts. NCD tackles the challenge of automatically discovering unseen categories in unlabelled data by leveraging the labelled data from seen categories, bridging the gap between known and novel categories. Meanwhile, GCD extends this challenge by allowing the unknown data to come from both labelled and novel categories. However, existing efforts on NCD and GCD mainly consider only static datasets. Our world, however, is inherently dynamic, necessitating

† Corresponding author: Kai Han (kaihanx@hku.hk)

intelligent systems that are not only able to discover novel categories but also can retain past knowledge while accommodating new information. Therefore, it is desired to develop methods that can discover novel classes from the unlabelled images over time. This task, called Continual Category Discovery (CCD) [55], extends the challenging open-world category discovery problem in a continual learning scenario (see Fig. 1). There are two major challenges in CCD. The first challenge is *catastrophic forgetting*, a well-known issue in continual learning settings [9]. Traditional techniques for mitigating forgetting, such as rehearsal-based [38], distillation-based [31], architecture-based [30], and prompting-based methods [49, 50], assume fully labelled data at each stage, which is incompatible with the CCD framework where the goal is to work with unlabelled data streams. The second challenge is *discovering novel visual concepts*. While GCD is a related task, it operates under the assumption of static sets containing both labelled and unlabelled data. However, this assumption does not hold for the CCD task, where all data in the continuous stream during the discovery stage are unlabelled.

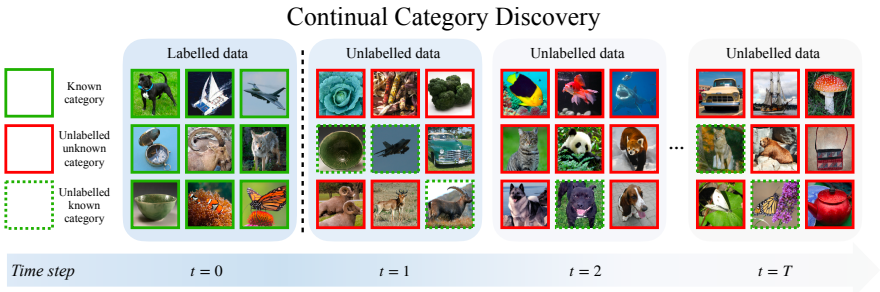


Fig. 1: Overview of the Continual Category Discovery task. In the initial stage, the model learns from labelled data, while in the subsequent stages, the model learns from a continuous data stream containing unlabelled instances from known and novel classes.

Recently, vision foundation models such as [6, 37] have achieved remarkable progress and shown promise in various vision tasks, from image classification and object detection to more complex tasks like scene understanding. Given the capabilities of these foundation models, especially the self-supervised ones, we are interested in unleashing the potential of such models for dynamic environments by repurposing them to effectively tackle the challenging CCD problem. To this end, we propose PromptCCD framework. This framework empowers the model to leverage any prompt pool for solving CCD. Specifically, within our framework, we introduce a *plug-and-play* Gaussian Mixture Prompting (GMP) module. This module utilizes a Gaussian mixture prompt pool to model the data distribution at each discovery stage dynamically. By enhancing the feature representation with our adaptive queried Gaussian mixture prompts, our method excels at identifying novel visual categories across successive stages. Simultaneously, these prompts enable the model to seamlessly adapt to emerging data while preserving its performance on previously discovered categories, thus mitigating catastrophic forgetting. In addition to outperforming existing CCD solutions, our framework provides the unique advantage of enabling *on-the-fly* estimation of the category

number in the unlabelled data, which is often assumed to be predetermined in prior works [55]. In this paper, we make the following contributions: (1) We propose a prompt learning framework for Continual Category Discovery (CCD), named *PromptCCD*. It can effectively repurpose the self-supervised vision foundation model for the challenging task of CCD, only introducing a small amount of extra learnable prompt parameters, and thus possessing strong scalability for practical use. (2) Within the proposed framework, we introduce *Gaussian Mixture Prompting* (GMP) module, a novel prompt learning technique that leverages Gaussian mixture components to enhance the representation learning and effectively address the issue of catastrophic forgetting when dealing with previously learned data. Notably, GMP's prompt serves a dual role, *i.e.*, as a task prompt, guiding the model during training, and as a class prototype, which is essential for CCD as label information is absent during class discovery. Moreover, GMP can be seamlessly integrated with other methods, enhancing their overall performance. An additional distinctive feature of GMP lies in its ability to estimate categories *on-the-fly*, making it well-suited for handling open-world tasks. (3) Finally, to evaluate the performance of the model for CCD, we extend the standard GCD metric to a new metric, called *continual ACC* (*cACC*). Extensive experiments on both generic and fine-grained datasets demonstrate that our method significantly outperforms state-of-the-art CCD methods.

2 Related Work

Semi-supervised learning aims at learning a classifier using both labelled and unlabelled data [7, 36]. Most works assume that the unlabelled data contains instances from the *same* categories in the labelled data [36]. Pseudo-labeling [40], consistency regularization [2, 28, 44, 45], and non-parametric classification [1] are among the popular methods. Some recent works do not assume the categories in the unlabelled and labelled set to be the same, such as [20, 43, 53], yet their focus is still on improving the performance of the categories from the labelled set.

Continual learning aims to train models that can learn to perform on a sequence of tasks, with the restriction of the model can only see the data for the current task it is trained on [9]. Catastrophic forgetting [35] is a phenomenon that when the model is trained on a new task, it will quickly forget the knowledge on the task it has been trained on before, resulting in a catastrophic reduction of performance on the old tasks. There exists a rich literature on designing methods that enable the model to both learn to do the new task and maintain the knowledge of old tasks [3, 4, 14, 30, 31, 38, 50]. However, these works all assume that the incoming tasks have all labels provided. In contrast, CCD assumes that the new data is fully unlabelled and can have category overlap with previous tasks.

Novel / Generalized category discovery addresses the problem where there are novel categories in the unlabelled data and the goal is to automatically categorize the unlabelled samples, leveraging the labelled samples from the seen categories. Novel Category Discovery (NCD), formalized by DTC [17], assumes no overlap between the unlabelled and labelled data. Several successful NCD methods have emerged, showing promising performance through ranking statistics [15, 16, 21, 56], data augmentation [59], and specialized objective func-

tion [13, 22]. The problem is later extended to Generalized Category Discovery (GCD) [46] by considering that the unlabelled data may contain samples from both known and novel categories. [46] finetunes a pretrained model using both self-supervised [8] and supervised contrastive losses [24] and subsequently obtains the label assignment using a semi-supervised k-means algorithm. SimGCD [51] introduces a strong parametric baseline based on [46] for GCD, obtaining strong performance. Other GCD methods focus on fine-grained categories [11], automatic category estimation [18, 58], and prompt learning [48, 54].

Continual category discovery is a challenging but relatively under-explored problem. NCDwF [23] studies NCD under the continual learning setting, where the model first learns from labelled data and subsequently focuses on novel category discovery solely from unlabelled data. NCDwF shows that feature distillation and mutual information-based regularizers are effective for this problem. Concurrent to NCDwF, FRoST [41] introduces a replay-based method that stores feature prototypes from labelled data during the discovery phase. MSc-iNCD [32] leverages pretrained self-supervised learning models to address this problem. Grow & Merge [55] studies GCD under the continual learning setting, where the model has access to the labelled data in the initial stage and the unlabelled data in sequences in the subsequent stages. This method utilizes a growing phase to detect novel categories and a merging phase to distil knowledge from both novel and previously learned categories into a single model. Other methods addressing the GCD problem under the continual learning setting include PA-CGCD [25], which prevents forgetting using a proxy-anchor-based method, and MetaGCD [52], which balances class discovery and prevents forgetting using a meta-learning framework. Another method, IGCD [57], studies GCD under the continual learning setting in a slightly different way with an emphasis on the iNaturalist dataset for plant and animal species discovery. In each stage, IGCD takes a partially labelled set of images as input, rather than a set of fully unlabelled data like [25, 52, 55]. In this paper, we consider Continual Category Discovery (CCD) as the setting studied in [25, 52, 55]. In CCD, the model receives the labelled set at the initial stage and is tasked to discover categories from the unlabelled data in the subsequent stages.

3 Method

Problem statement. The dataset in CCD contains both labelled data D^l and unlabelled data D^u. The labelled data $D^l = \{(x_i, y_i)\}_{i=1}^{N}$ contains tuples of the input $x_i \in \mathcal{X}$ and its corresponding labels $y_i \in \mathcal{Y}$ and it is only used in the initial stage for the model to learn useful features for the category discovery. In the following T discovery stages, at each stage, we receive a part of the unlabelled data $D_t^u \subset D^u$ that can be used to train the model. The unlabelled data D_t^u at each stage does not contain the labels, and it contains both known categories from previous stages and also novel categories. The goal of CCD is to train a model $\mathcal{H}_\theta : \mathcal{X} \to \mathcal{Z}$ parameterized by θ that first learns from labelled D^l and then in the following T discovery stages, learns from unlabelled data D_t^u such that $\mathcal{H}_\theta$ can be used to discover novel classes and assign class labels to all unlabelled instances utilizing representative feature without forgetting previous knowledge.

3.1 PromptCCD-B (*Baseline*): Learning Prompt Pool for CCD

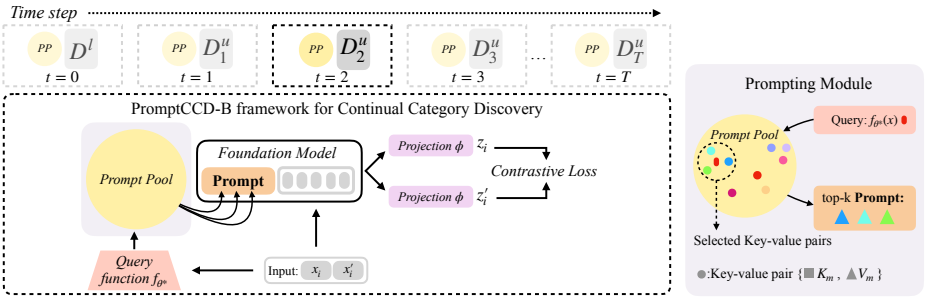


Fig. 2: Our baseline CCD framework adopts a prompt-based continual learning technique by utilizing a prompt pool module to adapt the vision foundation model for CCD.

Prompt learning [49, 50] has been shown effective for supervised continual learning. With properly designed prompts, the necessity of extensive modification for the model when handling the growing data stream can be greatly reduced. However, these methods can not be directly applied to the CCD task, as they assume the data stream to be fully annotated, which is not the case for CCD.

To address this gap, we propose a novel baseline prompt learning framework for CCD denoted as PromptCCD-B, taking inspiration from [49,50] which learn a pool of prompts to adopt a large-scale pretrained model on ImageNet-21K [39] (in a supervised manner) for supervised continual learning. This baseline is designed to learn a shared pool of prompts that can effectively adapt the self-supervised foundation model to tackle the CCD challenge. Specifically, the model extracts a feature from a query example using a frozen pretrained model, and the feature will be used to retrieve the top-k most relevant prompts from the fixed-size M prompts in the shared pool. These prompts are then used to guide the representation learning process by prepending them with the input embeddings, optimised with contrastive learning at each learning stage.

The overall framework of our baseline is shown in Fig. 2. Given a model $\mathcal{H}_\theta : \{\phi, f_\theta\}$, where ϕ is a projection head, and $f_\theta = \{f_e, f_b\}$ is the transformer-based feature backbone which consists of input embedding layer f_e and self-attention blocks f_b. An input image $x \in \mathbb{R}^{H\times W\times 3}$ where H, W represent the height and width of the image, is first split into L tokens (patches) such that $x_q \in \mathbb{R}^{L\times(h\times w\times 3)}$ where h, w represent the height and width of the image patches. These patches are then projected by the input embedding layer $x_e = f_e(x_q) \in \mathbb{R}^{L\times z}$. A learnable prompt pool with M prompts is denoted as $\mathbb{V} = \{(K_m, V_m)\}_{m=1}^{M}$ where $K_m \in \mathbb{R}^z$ and $V_m \in \mathbb{R}^{L_{pp}\times z}$ are the key-value learnable pairs and L_{pp} is the prompt pool's token length. We define a query function f_{θ^*} (*non-trainable*) to map the input image x to the feature space. The query process on the prompt pool operates in a key-value fashion. For a given query $f_{\theta^*}(x)$, we find the top-k most similar keys in the prompt pool and retrieve the associated value by:

$$\mathcal{V}_{\text{top-k}} = \{V_i | K_i \in \mathcal{T}_{\mathbb{V}}^k(f_{\theta^*}(x))\}, \tag{1}$$

where $\mathcal{T}_{\mathbb{V}}^k$ is a set of the top-k similar keys in $\mathbb{V}$. These retrieved prompts are then prepended to the patch embeddings to aid the learning process $x_{total} = [\mathcal{V}_{\text{top-k}}; x_e]$.

The baseline method is trained with contrastive learning, let $\{x_i, x_i'\}$ be two randomly augmented views of the same image x_i. We obtain their representations as $z_i = \phi(f_\theta(x_i))$ and $z_i' = \phi(f_\theta(x_i'))$. To optimize the prompt pool, we pull the selected keys closer to the corresponding query features by making use of a cosine distance loss:

$$\mathcal{L}_i^{\cos} = \sum_{K_m \in \mathcal{T}_{\mathbb{V}}^k(f_{\theta^*}(x))} \gamma(f_{\theta^*}(x_i), K_m), \tag{2}$$

where γ is the cosine distance function. Finally, when the training of stage t is finished, we transfer the current prompt pool $\mathbb{V}$ to the next stage.

Model optimization. To optimize the model's representation, we follow the GCD literature to adopt the contrastive loss:

$$\mathcal{L}_i^{\text{rep}} = -\frac{1}{\mid \mathbb{N}(i) \mid} \sum_{p \in \mathbb{N}(i)} \log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_n \mathbb{1}_{[n \neq i]} \exp(z_i \cdot z_n/\tau)}, \tag{3}$$

where $\mathbb{1}_{[n \neq i]}$ is an indicator function such that it equals to 1 *iff* $n \neq i$, and τ is the temperature value. If x_i is a labelled image, $\mathbb{N}(i)$ corresponds to images with the same label y in the mini-batch B. While if x_i is an unlabelled image, $\mathbb{N}(i)$ contains only the index of the other augmented view x_i' of the image, *i.e.*, $z_p = z_i'$. For the baseline model optimization, at the initial stage, *i.e.*, $t = 0$, we have our initial labelled set D^l, and each image may have more than one positive sample; while at the subsequent stages, *i.e.*, $t > 0$, we only have access to the unlabelled data D_t^u, and each image has only one positive sample, *i.e.*, its another augmented view. The prompt parameters are also simultaneously optimized during the model optimization process.

Limitations of baseline. Our baseline can achieve reasonably good performance, as can be seen in Sec. 4. It has several limitations. First, our baseline lacks an explicit mechanism to prevent forgetting. Without label information to guide it, the model may inadvertently bias its representation learning towards the current unlabelled data during fine-tuning, resulting in representation bias and forgetting. Another limitation arises from the fixed size of the prompt pool in our baseline framework. We rely on a predefined prompt pool size, which restricts the model's scalability. Consequently, the prompt pool's parameters may hinder the model's ability to discover a growing number of new categories. Lastly, our baseline framework lacks an efficient mechanism to estimate the number of categories dynamically, which is a crucial challenge for category discovery as per the GCD literature, but remains an open challenge under-explored in CCD.

3.2 PromptCCD: Learning Gaussian Mixture Prompt Pool for CCD

To address the aforementioned limitations in our baseline framework PromptCCD-B, here, we propose a novel Gaussian Mixture Prompting (GMP) module, which learns a parameter-efficient Gaussian Mixture Model (GMM) as the prompt pool, leading to a new framework, called PromptCCD (see Fig. 3).

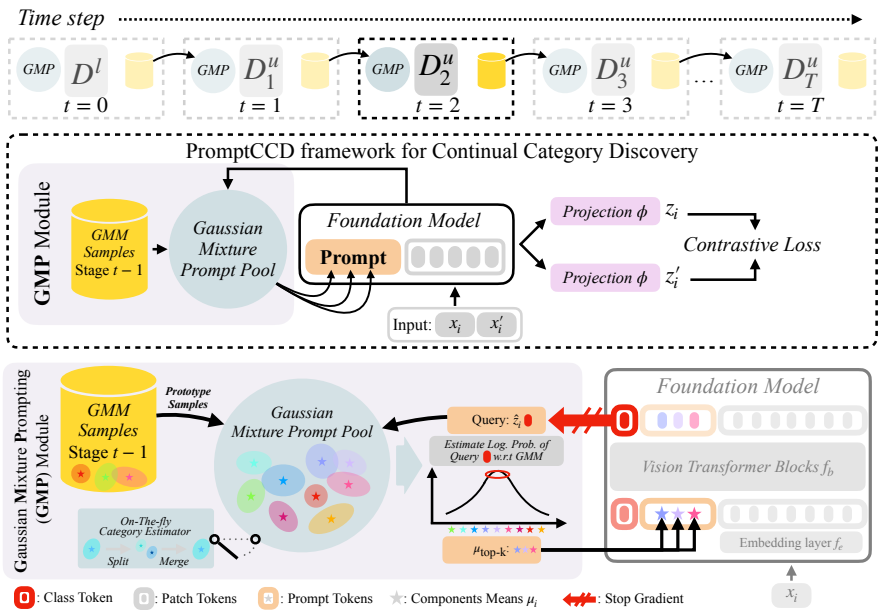


Fig. 3: Overview of our proposed PromptCCD framework and Gaussian Mixture Prompting (GMP) module. PromptCCD continually discovers new categories while retaining previously discovered ones by learning a dynamic GMP pool to adapt the vision foundation model for CCD. Specifically, we address CCD by making use of GMP modules to estimate the probability of input $\hat{z}_i$ by calculating the log-likelihood and use the top-k mean of components μ_i as prompts to guide the foundation model. Lastly, to retain previously learned prompts, we generate prototype samples from the fitted GMM at time step $t-1$ and fit the current GMM with these samples at time step t.

Gaussian mixtures prompting (GMP) module. The GMM is formulated as:

$$p(z) = \sum_{c=1}^{C} \pi_c \mathcal{N}(z|\mu_c, \Sigma_c) \quad \text{s.t.} \quad \sum_{c=1}^{C} \pi_c = 1, \tag{4}$$

where C is the number of Gaussian components, π_i is the learnable mixture weight, μ_i is the mean of each components, and Σ_i is the covariance of each component. Given the feature $\hat{z}_i = f_\theta(x_i)$ corresponding to the `[CLS]` token in the backbone, we calculate the log probability density value of each of the mixture components with the queried feature $\hat{z}_i$ and obtain a set of log-likelihood values for different GMM components. Finally, we find the top-k components in GMM with the highest log-likelihood values and retrieve the associated components' means:

$$\mu_{\text{top-k}} = \{\mu_c | c \in \mathcal{T}^k_{\text{GMM}}(p(\hat{z}_i))\}, \tag{5}$$

where $\mathcal{T}^k_{\text{GMM}}$ is a set of the top-k component(s) c. Similar to our baseline framework, PromptCCD-B, a set of embeddings $x_{total} = [\mu_{\text{top-k}}; x_e]$ is formed by prepending the selected prompts with the patch embeddings. We then apply the same contrastive learning objective as in Sec. 3.1 to optimize our PromptCCD framework. The Gaussian mixture prompt pool serves as the core

component that supports continuous category discovery across stages. After the training at stage t is done, we use the fitted GMM to sample a set of samples $\mathcal{Z}_t^s$ with S samples for each component c in the GMM. These samples are used to prevent forgetting previously learned knowledge, we achieve this by using these samples $\mathcal{Z}_t^s$ to fit the GMM of the next stage $t+1$. A pseudocode of the training procedure is provided in Sec. S1, supplementary material.

Our GMP possesses several unique strengths over the existing prompting techniques for supervised continual learning [49, 50]. First, GMP's prompt serves a dual role, namely (1) as a task prompt to instruct the model and (2) as class prototypes to act as parametric replay sample distribution for discovered classes. The second role, which is unique and important for CCD/GCD, not only allows the model to draw unlimited replay samples to facilitate the representation tuning and class discovery in the next time step but also allows the model to transfer knowledge of previously discovered and novel categories and incorporate this information when making the decision to discover a novel category. Second, our GMP module enables easy adjustment of parameters and efficient dynamic expansion across stages. This allows our model to enjoy great scalability which is especially important when handling a growing number of categories. Finally, the GMM-based design of our GMP module allows us to equip it with an automatic split-and-merge mechanism, allowing our model to estimate the unknown number of categories in the unlabelled data stream.

3.3 PromptCCD-U: Unknown Number of Classes in Unlabelled Data

When the class numbers in the unlabelled data are unknown, one way to approach this problem is to estimate it offline using the non-parametric clustering method introduced in [46] at each time step. In CCD, considering the continual learning nature of the problem, it would be more plausible to estimate the class numbers on-the-fly without introducing extra models or an offline process. Inspired by GPC [58], which introduces a GMM-based category number estimation method for GCD, by automatically splitting and merging clusters during learning through assessing the cluster's compactness and separability using a Markov chain Monte Carlo (MCMC) algorithm. We incorporate this key idea into our CCD framework, making use of our learned prompt pool in our GMP module, further enabling the capability of our framework for automatic category number estimation. We denote this extended variant of our framework as PromptCCD-U.

Specifically, consider stage $t = 1$ of CCD. We first extract the features for all the unlabelled samples D_t^u and also use our GMM in our GMP fitted in the previous stage $t = 0$ to generate a set of pseudo features (as a replay for previously learned classes from the labelled data D^l). Let the combined features be $\mathcal{Z}$. We then fit them into the GMM in our GMP. As the class number in D_t^u is unknown, we start the fitting by setting an initial class number of the known class number in D^l and incorporate a *split-and-merge* mechanism as in [58] to allow for the dynamic adjustment of the GMM. Particularly, for each of the Gaussian components of the GMM and we further decompose it into two sub-components, *i.e.*, $\mu_{c,1}, \mu_{c,2}$ and $\Sigma_{c,1}, \Sigma_{c,2}$. We then calculate the Hastings ratio which measures the compactness and separability of the clusters during the fitting iteration. The

Hastings ratio for splitting a cluster is defined as:

$$H_s = \frac{\Gamma(N_{c,1})h(\mathcal{Z}_{c,1})\Gamma(N_{c,2})h(\mathcal{Z}_{c,2})}{\Gamma(N_c)h(\mathcal{Z}_c)}, \tag{6}$$

where Γ is the factorial function, h is the marginal likelihood function of the observed data $\mathcal{Z}$, $\mathcal{Z}_{c,1}$ denotes the data points assigned to the subcluster $\{c,1\}$, and $N_{c,1}$ is the number of data points in the subcluster $\{c,1\}$. Note that H_s is in the range of $(0,+\infty)$, thus we will use $p_s = min(1, H_s)$ as a valid probability for performing the splitting operation. When the fitting of the GMM is converged, the number of the resulting GMM components is then the class number of all classes seen so far. The number of new classes in D_t^u can be obtained by simply subtracting the previously learned class number.

4 Experiments

In this section, we describe our experimental setups in Sec. 4.1. Next, we present our main experimental results in Sec. 4.2. Finally, in Sec. 4.3 we analyze the effectiveness of our model's components and design choices.

4.1 Experimental Setups

Datasets. We conduct our experiments on various benchmark datasets, namely CIFAR100 (C100) [27], ImageNet-100 (IN-100) [42], TinyImageNet (Tiny) [29], Caltech-UCSD Birds-200-2011 (CUB) [47], FGVC-Aircraft [34], Stanford-Cars (SCars) [26], and Caltech-101 (C-101) [12]. Statistics of the benchmark datasets are shown in Tab. 1. CCD task consists of several stages. We set the number of stages to 4 with data splits presented in Tab. 2 following [55].

Table 1: Statistics of the CCD benchmark datasets following the splits in Tab. 2.

Stages	C100 [27]		IN-100 [42]		Tiny [29]		C-101 [12]		Aircraft [34]		SCars [26]		CUB [47]	
	C	#	C	#	C	#	C	#	C	#	C	#	C	#
Stage 0 (D^l)	70	$30.45K$	70	$77.46K$	140	$60.90K$	71	$4.70K$	70	$1.98K$	130	$4.62K$	140	$3.65K$
Stage 1 (D_1^u)	80	$5.95K$	80	$15.14K$	160	$11.90K$	81	$0.73K$	80	$0.37K$	152	$0.98K$	160	$0.71K$
Stage 2 (D_2^u)	90	$6.55K$	90	$16.66K$	180	$13.10K$	91	$0.65K$	90	$0.43K$	174	$1.13K$	180	$0.79K$
Stage 3 (D_3^u)	100	$7.05K$	100	$17.94K$	200	$14.10K$	101	$1.12K$	100	$0.55K$	196	$1.38K$	200	$0.85K$

Implementation details. We use ViT-B/16 backbone [10] pretrained with DINO [6, 37] for all experiments. Please note that [49, 50] utilized a pretrained model with supervision, which is suitable for the standard supervised continual learning task. However, it is not well-suited to use such pretrained models for CCD task due to label information leakage. During training, only the final block of the vision transformer is finetuned for 200 epochs with a batch size of 128, using SGD optimizer and cosine decay learning rate scheduler with an initial learning rate of 0.1 and minimum learning rate of 0.0001, and weight decay of 0.00005. For the GMP module, we optimize the GMM every 30 epochs and start the prompt learning when the epoch is greater than 30. We set top-k to be 5, and the number of GMM samples to 100. We pick

Table 2: Data splits.

Class splits	D^l	D_1^u	D_2^u	D_3^u
$\{y_i \mid y_i \leq 0.7 * \lvert\mathcal{Y}\rvert\}$	87%	7%	3%	3%
$\{y_i \mid 0.7 * \lvert\mathcal{Y}\rvert < y_i \leq 0.8 * \lvert\mathcal{Y}\rvert\}$	0%	70%	20%	10%
$\{y_i \mid 0.8 * \lvert\mathcal{Y}\rvert < y_i \leq 0.9 * \lvert\mathcal{Y}\rvert\}$	0%	0%	90%	10%
$\{y_i \mid 0.9 * \lvert\mathcal{Y}\rvert < y_i \leq \lvert\mathcal{Y}\rvert\}$	0%	0%	0%	100%

the final model by selecting the best performing model on *'Old' ACC* using the validation set (evaluated every 10 epochs). All input images are resized to 224×224 and augmented to match the DINO pretrained model settings. For our method, we finetune the last transformer block of the model f_b and the projection head ϕ (Sec. S8 for details) using the loss introduced in Sec. 3. For other compared methods, we carefully chose the right hyper-parameters following their original papers. Finally, we dynamically estimate the class number using the method described in Sec. 3.3, following a procedure similar to [58]. We build our framework with PyTorch on a single NVIDIA RTX 3090 GPU.

Evaluation metric. The model is finetuned at each stage. At test time, the classification token `[CLS]` features are used for clustering. For the clustering algorithm and label assignment, we use semi-supervised k-means (SS-k-means) [46] on the unlabelled sets D_t^u and measure the accuracy given the ground truth y_i and the clustering prediction $\hat{y}_i$ such that:

Algorithm 1 *Continual ACC* (*cACC*) evaluation metric

Input: Models $\{f_\theta^t \mid t = 1, \ldots, T\}$ and datasets $\{D^l, D^u\}$.
Output: *cACC* value.
Require: SS-k-MEANS(Model, Labelled set, Unlabelled set).
Require: Initialize set $\mathbb{A}^L \leftarrow D^l$.
for $t \in \{1, \cdots, T\}$ **do**
  ACC_t , $D_t^{u^*} \leftarrow$ SS-k-MEANS(f_θ^t, $\mathbb{A}^L$, D_t^u)
  $\mathbb{A}^L \leftarrow \mathbb{A}^L \cup D_t^{u^*}$ *// append $D_t^{u^*}$ (w/ assigned labels) to $\mathbb{A}^L$*
$ACCs \leftarrow \{ACC_t \mid t = 1, \ldots, T\}$
$cACC \leftarrow$ AVERAGE($ACCs$)
return *cACC*

$$ACC = \max_{g \in \mathcal{G}(\mathcal{Y}_U)} \frac{1}{|D_t^u|} \sum_{i=1}^{|D_t^u|} \mathbb{1}\{y_i = g(\hat{y}_i)\}, \tag{7}$$

where $\mathcal{G}(\mathcal{Y}_\mathcal{U})$ represents a set of all permutations of class labels in the unlabelled set D_t^u. For the evaluation across stages in CCD, based on the standard clustering accuracy *ACC* for GCD, we introduce a new metric, called *continual ACC* (*cACC*), for the continual setting considering the sequential data stream. Commonly, in GCD, the ACC values are evaluated for *'All'*, *'Old'*, and *'New'* splits of the dataset. In CCD, for one time step t, *'All'* indicates the overall accuracy on the entire set D_t^u. *'Old'* and *'New'* indicate the accuracy from instances of unlabelled data from D_t^{uo} and D_t^{un} respectively. The evaluation protocol of cACC is summarized in Alg. 1. Instead of relying solely on the labelled data D^l to guide the SS-k-means clustering algorithm, *cACC* incorporates labelled data from $\{D^l, D_1^{u^*}, \ldots, D_{t-1}^{u^*}\}$, where $D_i^{u^*}$ represents data with assigned labels from previously unlabelled data D_i^u. High-quality label assignments facilitate the subsequent category discovery while low-quality label assignments accumulate errors for the subsequent category discovery.

Comparison with other methods. We compare our method with the other representative CCD methods: 1) Grow & Merge (G&M) [55]; 2) MetaGCD [52]; 3) PA-CGCD [25]; and re-implement GCD methods for CCD task, including 4) ORCA [5]; 5) GCD [46]; 6) SimGCD [51]. As G&M's encoder is based on ResNet18 network [19], we re-implement their dynamic branch mechanism with the ViT backbone and observe improved performance for their method compared to their original results (see Sec. S5). We also re-implement GCD and SimGCD for CCD settings by incorporating a replay-based method. At each stage, the model saves samples for discovered classes and mixes them with incoming streamed

images. Lastly, we adopt L2P's [50] and DualPrompt's [49] prompt pool modules, following their original prompt pool hyperparameter choices, and integrate them with PromptCCD-B framework as our baselines.

4.2 Main Results

We evaluate our method in two scenarios: when the class number C, is known (Tab. 3, 4, and 5) in each unlabelled set at different stages, and when C is unknown (Tab. 6). We report the $cACC$ by averaging the results across all stages. We also provide the breakdown results for each stage in Sec. S2. In addition, we also report results using other metrics in Sec. S4 and Sec. S6.

Table 3: The $cACC$ results of our method with different prompt pool designs for CCD on generic and fine-grained benchmark datasets where C is *known* in each unlabelled set. The experiments are conducted five times with different random seeds.

		CIFAR100			ImageNet-100		
Method	Prompt Pool	*All*	*Old*	*New*	*All*	*Old*	*New*
PromptCCD-B (Ours)	L2P [50]	51.59 ± 6.3	67.27 ± 8.7	46.14 ± 6.1	66.14 ± 2.3	81.05 ± 1.5	61.36 ± 3.2
PromptCCD-B (Ours)	DP [49]	59.60 ± 1.2	78.93 ± 1.3	54.14 ± 1.6	70.64 ± 1.3	83.46 ± 0.4	67.24 ± 1.8
PromptCCD (Ours)	GMP (Ours)	**63.97** ± 1.4	76.67 ± 2.6	60.01 ± 1.7	**75.38** ± 0.7	81.16 ± 0.7	73.71 ± 0.8
		TinyImageNet			CUB		
Method	Prompt Pool	*All*	*Old*	*New*	*All*	*Old*	*New*
PromptCCD-B (Ours)	L2P [50]	56.66 ± 0.4	66.05 ± 0.8	53.69 ± 0.4	51.31 ± 1.0	72.43 ± 1.0	44.27 ± 1.4
PromptCCD-B (Ours)	DP [49]	58.61 ± 1.5	66.61 ± 0.6	55.84 ± 1.7	56.30 ± 1.1	78.64 ± 1.7	48.91 ± 1.1
PromptCCD (Ours)	GMP (Ours)	**61.15** ± 1.0	66.29 ± 2.0	58.83 ± 1.0	**56.65** ± 1.0	79.88 ± 2.5	48.96 ± 0.8

Table 4: Comparison with other methods for CCD leveraging pretrained DINO and DINOv2 models on generic datasets with the *known* C in each unlabelled set.

		CIFAR100			ImageNet-100			TinyImageNet			Caltech-101		
Method	Pretrained Model	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
ORCA [5]	DINO	60.91	66.61	58.33	40.29	45.85	35.40	54.71	63.13	51.93	76.77	82.80	73.20
GCD [46]	DINO	58.18	72.27	52.83	69.41	81.56	65.65	55.20	65.87	51.61	78.27	86.60	72.92
SimGCD [51]	DINO	25.56	38.76	20.43	31.38	40.47	27.44	33.40	29.11	34.74	33.65	37.53	31.62
GCD w/replay	DINO	49.93	73.15	41.47	72.04	83.75	69.01	56.33	67.54	52.60	76.51	86.14	72.48
SimGCD w/replay	DINO	40.13	66.72	30.91	47.53	67.86	39.18	37.45	58.15	30.36	49.38	52.72	47.99
Grow & Merge [55]	DINO	57.43	63.68	55.31	67.84	75.10	66.60	52.14	59.68	49.96	75.75	83.66	71.59
MetaGCD [52]	DINO	55.49	69.38	48.98	66.41	80.54	60.65	55.26	66.12	50.79	80.75	89.02	75.86
PA-CGCD [25]	DINO	58.25	87.11	49.04	64.79	91.15	57.83	51.13	74.95	43.52	77.96	94.75	69.66
PromptCCD w/GMP (Ours)	DINO	**64.17**	75.57	60.34	**76.16**	81.76	74.35	**61.84**	66.54	60.26	**82.44**	89.08	79.72
GCD [46]	DINOv2	65.35	77.06	60.46	71.58	83.02	68.05	59.05	77.44	53.41	83.00	88.65	79.80
MetaGCD [52]	DINOv2	52.10	79.64	43.13	70.20	82.62	64.66	56.15	74.69	49.37	83.05	88.08	80.89
PA-CGCD [25]	DINOv2	54.36	79.19	45.65	74.82	88.20	72.02	52.10	68.07	46.32	83.06	94.07	77.55
PromptCCD w/GMP (Ours)	DINOv2	**69.73**	78.01	66.16	**76.28**	82.61	74.53	**68.20**	75.56	65.23	**83.86**	87.93	81.42

Variants of PromptCCD. In our comparison with our baseline PromptCCD-B, as shown in Tab. 3, our PromptCCD w/GMP demonstrates superior performance for CCD. Specifically, our model outperforms the baselines across all *'All'* accuracy, while the baselines experience performance degradation in later stages (see Sec. S2). We attribute this decline to the baselines' non-scalable prompt pool parameters, which restrict their ability to *instruct* the model as the parameter count grows. In contrast, our scalable prompting technique leverages Gaussian mixture models to construct a flexible pool of prompts. Additionally, we ensure knowledge retention by sampling learned mixture components for fitting subsequent GMM.

Table 5: Comparison with other methods for CCD leveraging pretrained DINO and DINOv2 models on fine-grained datasets with the *known* C in each unlabelled set.

Method	Pretrained Model	Aircraft			Stanford Cars			CUB		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
ORCA [5]	DINO	30.77	25.71	32.44	20.79	33.40	17.60	41.73	66.19	34.14
GCD [46]	DINO	47.37	61.43	42.53	39.21	58.29	33.45	54.98	75.47	48.15
SimGCD [51]	DINO	29.03	35.72	25.61	21.01	40.93	16.48	39.89	59.25	33.75
GCD w/replay	DINO	45.63	62.38	39.89	39.87	58.18	33.89	54.66	74.64	47.81
SimGCD w/replay	DINO	37.44	61.43	28.96	22.76	49.04	16.65	42.08	72.65	31.92
Grow & Merge [55]	DINO	31.06	33.33	30.78	21.90	35.29	18.17	38.87	65.00	30.29
MetaGCD [52]	DINO	44.63	59.05	39.39	35.98	56.97	29.96	44.59	74.40	35.40
PA-CGCD [25]	DINO	48.24	73.09	40.60	43.88	80.43	33.54	52.48	77.26	44.74
PromptCCD w/GMP (Ours)	DINO	**52.64**	60.48	50.23	**44.07**	66.36	36.83	**55.45**	75.48	48.56
GCD [46]	DINOv2	57.87	63.80	55.39	58.52	71.65	53.80	66.70	83.33	60.81
MetaGCD [52]	DINOv2	54.90	64.29	52.08	57.16	71.87	52.01	62.19	82.50	55.13
PA-CGCD [25]	DINOv2	58.15	77.62	51.08	64.91	89.64	57.84	66.88	92.62	58.48
PromptCCD w/GMP (Ours)	DINOv2	**62.71**	68.33	60.82	**65.08**	76.60	60.75	**67.81**	81.55	62.81

Comparison with known class numbers. The CCD benchmark results for generic and fine-grained datasets are presented in Tab. 4 and 5. Remarkably, our PromptCCD w/GMP outperforms other approaches across all datasets in terms of overall accuracy *'All'*. Notably, our approach maintains the balance between the *'Old'* and *'New'* accuracy compared to existing methods and achieving improved performance. As our model is based on GCD [46], we show that simply integrating our Gaussian mixture prompt module enables effective adaptation to the CCD settings. This highlights the robustness and versatility of our approach in handling CCD.

Table 6: Comparison with other methods for CCD leveraging the pretrained DINO model when the class number C in each unlabelled set is *unknown*.

	Est. method	CIFAR100			ImageNet-100			TinyImageNet			CUB		
Category discovery at stage --→		1	2	3	1	2	3	1	2	3	1	2	3
Estimated category C	GPC	77	78	81	73	73	83	158	164	168	163	172	175
Ground truth category C	–	80	90	100	80	90	100	160	180	200	160	180	200
Methods		*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
GCD [46]	GPC	57.20	68.23	52.87	56.60	77.75	48.28	53.68	64.71	50.00	50.02	71.31	42.82
Grow & Merge [55]	GPC	56.59	60.59	54.02	52.83	72.34	45.46	53.44	56.49	51.94	36.88	63.81	28.10
MetaGCD [52]	GPC	53.20	66.22	47.09	51.40	75.73	41.53	57.92	62.82	54.64	44.56	70.24	36.57
PA-CGCD [25]	GPC	56.46	82.86	47.68	48.63	84.51	36.98	50.40	70.17	43.87	50.61	74.53	42.60
PromptCCD-U w/GMP (Ours)	GPC	**63.20**	71.19	59.90	**63.14**	76.68	56.87	**60.81**	71.07	56.81	**51.20**	73.33	43.51

Comparison with unknown class numbers. To show the performance comparison for each model in a more realistic setting where C is unknown, we also report the benchmark results in Tab. 6, where we show 5 representative methods, *i.e.*, [25,46,52,55], and ours. Our method consistently outperforms all other methods by a large margin across the board, demonstrating the superior performance of our approach in the more realistic case when the class number is unknown.

Qualitative analysis. Lastly, to visualize the feature representation generated by our method, we use t-SNE algorithm [33] to visualize the high-dimensional features of $\{D^l, D_t^u\}$ on each stage. For the sake of comparison, we also provide the visualization for the feature representation generated by Grow & Merge [55]. The qualitative visualization can be seen in Fig. 4; nodes of the same colour indicate that the instances belong to the same category. Moreover, for stage $t >$

0, we only highlight the feature's node belonging to unknown novel categories. It is observed that across stages, our cluster features are more discriminative.

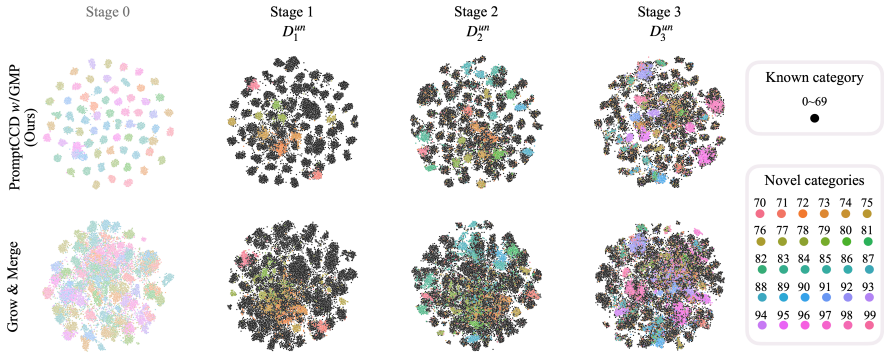


Fig. 4: t-SNE visualization of CIFAR100 with features from our model PromptCCD w/GMP and Grow & Merge on each stage.

4.3 Model Component Analysis

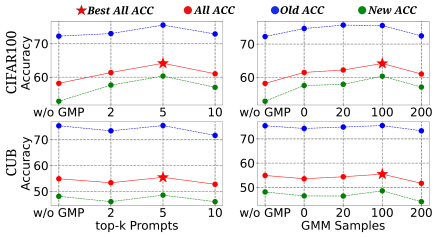
Top-k vs random prompts. In Tab. 8, to validate the effectiveness of using top-k prompts, we compare the results by using top-k and random-k prompts. We observe that using random-k prompts hurts the performance, as evidenced by that the performance using random-k is worse than without using any prompts. In contrast, our top-k strategy leads to significantly improved performance, especially for the *'New' ACC*. This observation suggests that prompting with top-k class prototypes indeed aids in the discovery of novel classes.

The choices of numbers for top-k and GMM samples in GMP. To investigate the effectiveness of our GMP module, we analyzed each component in our prompt module and present the results in Tab. 7. The results show a clear advantage of adopting the GMP into our framework. The number of top-k prompts and the number of GMM samples are identified as important factors. The optimal configuration is top-5 for prompt selection, and 100 samples for sampling (Fig. 5), which appears to be a good trade-off.

Table 7: Ablation study on different components of our GMP on C100 and CUB.

top-k Prompts	GMM Samples	C100 Avg. *ACC* *All*	*Old*	*New*	CUB Avg. *ACC* *All*	*Old*	*New*
0	0	58.18	72.27	52.83	54.98	75.47	48.15
5	0	61.48	74.68	57.55	53.54	74.28	46.47
5	20	62.21	75.71	57.90	54.37	74.88	46.41
5	200	61.00	72.46	57.08	51.67	73.33	44.08
2	100	61.39	73.04	57.64	53.36	73.45	46.04
5	100	**64.17**	75.57	60.34	**55.45**	75.48	48.56
10	100	61.03	72.91	56.97	52.76	71.67	46.02

Fig. 5: Performance curves depicted from Tab. 7 ablation results.



Improving other CCD methods with our GMP. Thanks to the great flexibility of our GMP, it can serve as a *plug-and-play* module and be seamlessly integrated with other methods. Table 9 presents the results of integrating GMP with

G&M and MetaGCD methods, showcasing significant improvements and highlighting the effectiveness of our GMP module. Nevertheless, our PromptCCD w/GMP substantially outperforms these enhanced CCD methods.

Table 8: Study on the effectiveness of top-k prompts compared with randomly picked prompts from GMM.

	CIFAR100 Avg. *ACC*			ImageNet-100 Avg. *ACC*		
PromptCCD	*All*	*Old*	*New*	*All*	*Old*	*New*
w/o GMP	58.18	72.27	52.83	69.41	81.56	65.65
GMP (random-k)	59.98	73.81$^{+1.54}$	55.68$^{+2.85}$	68.30	80.09$^{-1.47}$	63.50$^{-2.15}$
GMP (top-k) (Ours)	**64.17**	75.57$^{+3.30}$	60.34$^{+7.51}$	**76.16**	81.76$^{+0.20}$	74.35$^{+8.70}$
	TinyImageNet Avg. *ACC*			CUB Avg. *ACC*		
PromptCCD	*All*	*Old*	*New*	*All*	*Old*	*New*
w/o GMP	55.20	65.87	51.61	54.98	75.47	48.15
GMP (random-k)	55.69	63.95$^{-1.92}$	52.52$^{+0.91}$	51.46	73.10$^{-2.37}$	43.90$^{-4.25}$
GMP (top-k) (Ours)	**61.84**	66.54$^{+0.67}$	60.26$^{+8.65}$	**55.45**	75.48$^{+0.01}$	48.56$^{+0.41}$

Table 9: Study on the effectiveness of GMP module on different models. Here, we show the performances on CIFAR100.

Method	Prompt	*All*	*Old*	*New*
G&M	*w/o* GMP	57.43	63.68	55.31
G&M	*w*/GMP	**61.14**	**64.94**	**59.10**
MetaGCD	*w/o* GMP	55.49	69.38	48.98
MetaGCD	*w*/GMP	**58.99**	**69.69**	**54.29**
PromptCCD	*w/o* GMP	58.18	72.27	52.83
PromptCCD	*w*/GMP	**64.17**	**75.57**	**60.34**

Further study on the baseline prompts. We also explore the impact of prompt pool size for PromptCCD-B w/{L2P, DP} as indicated in Tab. 10, by experimenting with different prompt sizes. We find that varying the number of prompts does not significantly affect performance, even when the size aligns with the total number of classes in the CUB. In all cases, the results are significantly worse than the results obtained by our GMP above. This reveals the limitations of our baseline prompt method, particularly its reliance on fixed-size prompts and lack of scalability. Moreover, this also highlights the importance and effectiveness of our GMP design, the superior effectiveness of which can not be achieved by simply changing the prompt pool size of the baseline L2P and DP prompt pool size.

Table 10: Study on the PromptCCD-B w/{L2P, DP} pool size on CUB.

	w/ L2P			*w/* DP		
Pool Size	*All*	*Old*	*New*	*All*	*Old*	*New*
5	48.69	70.12	41.51	55.54	77.78	48.21
10	50.57	73.22	43.28	55.21	77.24	48.04
20	49.32	70.60	42.29	55.41	76.31	48.18
40	48.59	69.26	41.43	53.97	77.38	46.26
100	51.84	73.09	44.39	55.12	77.26	47.82
200	49.40	71.79	41.94	54.54	79.05	46.29

Please refer to the supplementary material for additional details and results.

5 Conclusion

In this paper, we have introduced PromptCCD, a simple yet effective framework for Continual Category Discovery (CCD) that tackles the challenge of discovering novel categories in the continuous stream of unlabelled data without catastrophic forgetting. By introducing the GMP module, PromptCCD dynamically updates the data representation and prevents forgetting during category discovery. Additionally, GMP enables on-the-fly estimation of category numbers, eliminating the need for prior knowledge of the category numbers. Our extensive evaluations on diverse datasets, along with our extended evaluation metric *cACC*, show that PromptCCD outperforms existing methods, highlighting its effectiveness in CCD.

Acknowledgments

This work is supported by the Hong Kong Research Grants Council - General Research Fund (Grant No.: 17211024).

References

1. Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., Rabbat, M.: Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In: ICCV (2021)
2. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. In: NeurIPS (2019)
3. Boschini, M., Bonicelli, L., Buzzega, P., Porrello, A., Calderara, S.: Class-incremental continual learning into the extended der-verse. IEEE TPAMI (2022)
4. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. In: NeurIPS (2020)
5. Cao, K., Brbić, M., Leskovec, J.: Open-world semi-supervised learning. In: ICLR (2022)
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
7. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press (2006)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
9. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE TPAMI (2021)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
11. Fei, Y., Zhao, Z., Yang, S., Zhao, B.: Xcon: Learning with experts for fine-grained category discovery. In: BMVC (2022)
12. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE TPAMI (2006)
13. Fini, E., Sangineto, E., Lathuilière, S., Zhong, Z., Nabi, M., Ricci, E.: A unified objective for novel class discovery. In: ICCV (2021)
14. Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S.G., Grefenstette, E., Ramalho, T., Agapiou, J., et al.: Hybrid computing using a neural network with dynamic external memory. Nature (2016)
15. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics. In: ICLR (2020)
16. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Autonovel: Automatically discovering and learning novel visual categories. IEEE TPAMI (2021)
17. Han, K., Vedaldi, A., Zisserman, A.: Learning to discover novel visual categories via deep transfer clustering. In: ICCV (2019)
18. Hao, S., Han, K., Wong, K.Y.K.: Cipr: An efficient framework with cross-instance positive relations for generalized category discovery. TMLR (2024)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
20. Huang, J., Fang, C., Chen, W., Chai, Z., Wei, X., Wei, P., Lin, L., Li, G.: Trash to treasure: harvesting ood data with cross-modal matching for open-set semi-supervised learning. In: ICCV (2021)
21. Jia, X., Han, K., Zhu, Y., Green, B.: Joint representation learning and novel category discovery on single- and multi-modal data. In: ICCV (2021)
22. Joseph, K.J., Paul, S., Aggarwal, G., Biswas, S., Rai, P., Han, K., Balasubramanian, V.N.: Spacing loss for discovering novel categories. In: CVPR Workshop (2022)

23. Joseph, K., Paul, S., Aggarwal, G., Biswas, S., Rai, P., Han, K., Balasubramanian, V.N.: Novel class discovery without forgetting. In: ECCV (2022)
24. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: NeurIPS (2020)
25. Kim, H., Suh, S., Kim, D., Jeong, D., Cho, H., Kim, J.: Proxy anchor-based unsupervised learning for continuous generalized category discovery. In: ICCV (2023)
26. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: ICCV workshop (2013)
27. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)
28. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: ICLR (2017)
29. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
30. Li, X., Zhou, Y., Wu, T., Socher, R., Xiong, C.: Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In: ICML (2019)
31. Li, Z., Hoiem, D.: Learning without forgetting. IEEE TPAMI (2017)
32. Liu, M., Roy, S., Zhong, Z., Sebe, N., Ricci, E.: Large-scale pre-trained models are surprisingly strong in incremental novel class discovery. In: ICPR (2024)
33. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR (2008)
34. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
35. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation. Elsevier (1989)
36. Oliver, A., Odena, A., Raffel, C.A., Cubuk, E.D., Goodfellow, I.: Realistic evaluation of deep semi-supervised learning algorithms. In: NeurIPS (2018)
37. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. TMLR (2023)
38. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR (2017)
39. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. In: NeurIPS (2021)
40. Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In: ICLR (2021)
41. Roy, S., Liu, M., Zhong, Z., Sebe, N., Ricci, E.: Class-incremental novel class discovery. In: ECCV (2022)
42. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
43. Saito, K., Kim, D., Saenko, K.: Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. In: NeurIPS (2021)
44. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: NeurIPS (2020)
45. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS (2017)

46. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. In: CVPR (2022)
47. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. California Institute of Technology (2011)
48. Wang, H., Vaze, S., Han, K.: Sptnet: An efficient alternative framework for generalized category discovery with spatial prompt tuning. In: ICLR (2024)
49. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: ECCV (2022)
50. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: CVPR (2022)
51. Wen, X., Zhao, B., Qi, X.: Parametric classification for generalized category discovery: A baseline study. In: ICCV (2023)
52. Wu, Y., Chi, Z., Wang, Y., Feng, S.: Metagcd: Learning to continually learn in generalized category discovery. In: ICCV (2023)
53. Yu, Q., Ikami, D., Irie, G., Aizawa, K.: Multi-task curriculum framework for open-set semi-supervised learning. In: ECCV (2020)
54. Zhang, S., Khan, S., Shen, Z., Naseer, M., Chen, G., Khan, F.S.: Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In: CVPR (2023)
55. Zhang, X., Jiang, J., Feng, Y., Wu, Z.F., Zhao, X., Wan, H., Tang, M., Jin, R., Gao, Y.: Grow and merge: A unified framework for continuous categories discovery. In: NeurIPS (2022)
56. Zhao, B., Han, K.: Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In: NeurIPS (2021)
57. Zhao, B., Mac Aodha, O.: Incremental generalized category discovery. In: ICCV (2023)
58. Zhao, B., Wen, X., Han, K.: Learning semi-supervised gaussian mixture models for generalized category discovery. In: ICCV (2023)
59. Zhong, Z., Zhu, L., Luo, Z., Li, S., Yang, Y., Sebe, N.: Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In: CVPR (2021)