Automatically Discovering Novel Visual Categories with Adaptive Prototype Learning

Lu Zhang†, Lu Qi†, Xu Yang, Hong Qiao, Ming-Hsuan Yang, Zhiyong Liu

Abstract—This paper tackles the problem of novel category discovery (NCD), which aims to discriminate unknown categories in large-scale image collections. The NCD task is challenging due to the closeness to the real world scenarios, where we have only encountered some partial classes and images. Unlike other works on the NCD, we leverage the prototypes to emphasize the importance of category discrimination and alleviate the issue with missing annotations of novel classes. Concretely, we propose a novel adaptive prototype learning method consisting of two main stages: prototypical representation learning and prototypical self-training. In the first stage, we obtain a robust feature extractor, which could serve for all images with base and novel categories. This ability of instance and category discrimination of the feature extractor is boosted by self-supervised learning and adaptive prototypes. In the second stage, we utilize the prototypes again to rectify offline pseudo labels and train a final parametric classifier for category clustering. We conduct extensive experiments on four benchmark datasets, and demonstrate the effectiveness and robustness of the proposed method with the state-of-the-art performance. The source code and trained models will be made available at this github site.

Index Terms—novel category discovery, image recognition, transfer learning.

1 Introduction

Recently, various computer vision tasks such as image classification [1, 2] and face recognition [3, 4] have obtained significant advances driven by deep learning. With the help of large-scale datasets, *e.g.*, ImageNet [5] and IBUG-300W [6], the trained models of those tasks manifest state-of-the-art recognition ability in new images. However, those tasks are usually in the close-set, requiring classifying images to limited pre-defined categories. It is intrinsically difficult for trained models to expand the learned knowledge to novel concepts [7, 8] as human beings can effortlessly achieve. For example, young children can discover novel shapes and animals [9, 10] (*e.g.*, triangle and bird) and differentiate them based on other seen classes [11, 12] (*e.g.*, circle and dog). This is an innate capability of humans but a great challenge for deep learning models [12, 13]. Making deep models accommodate to the real world has drawn increasing attention in the vision community.

The task of *novel category discovery* (NCD) has been proposed to solve the open-world problem in recent years. Given some labelled data of partial categories, it aims at partitioning unlabelled data into some semantic clusters, which we could regard as anonymous classes. These clusters are open-world without the limitation to the predefined categories. As shown in Figure 1, the NCD task is closer to the real world scenarios, where we can access enormous data but only very few are annotated with limited categories. Thus, the NCD task should require supervised classification and unsupervised clustering methods. Inspired by methods developed for learning to cluster [14, 15] that transfer binary labels from labelled to unlabelled data, recent NCD approaches [12, 16–18] usually utilize pair-wise similarity shared by all categories to produce pseudo-binary labels. This training pipeline could generate robust instance discrimination that helps separate the novel visual concepts. However, we show that both instance and category discrimination are essential to discovering the novel categories in the NCD task. We note that the ability of category discrimination is more relevant to the NCD task and has been proven crucial in traditional classification tasks. In this work, we aim to boost the model ability of both instance and category discrimination for the NCD task.

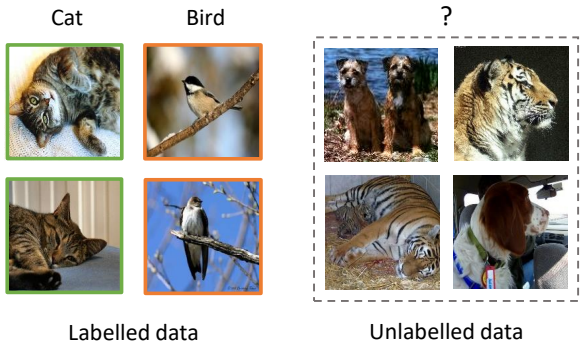


Fig. 1. Illustration of the task of novel category discovery. Given limited labelled images of some known categories, the model needs to automatically separate unlabelled images of novel categories, thus being able to recognize both old and novel categories in testing data. Images are from the ImageNet [5] dataset.

We can easily use a more robust self-supervised method

• *Lu Zhang, Xu Yang, Hong Qiao and Zhiyong Liu are with State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, China. Xu Yang and Zhiyong Liu are also with Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, China*
Lu Qi and Ming-Hsuan Yang are with University of California at Merced, USA
† *Both authors contribution equally to this work.*

to improve instance discrimination. For category discrimination, the standard strategy is cross-entropy loss. However, adopting a similar strategy is not trivial in the NCD task due to lacking novel class labels. Instead, we propose utilizing the adaptive prototypes to encode novel category information, where each prototype represents a class. During training, the class prototypes could be dynamically updated by adopting more discriminative instance features with momentum. Specifically, our method has two stages, including prototypical representation learning (stage I) and prototypical self-training (stage II).

The aim of stage I is to obtain a robust feature extractor to serve all images, irrespective of base or novel categories. We achieve it by the self-supervised learning method DINO [19] and the module of online prototype learning (OPL). On the one hand, DINO is a more robust self-supervised learning method without requiring negative samples. Meanwhile, we present a new data augmentation strategy named restricted rotation for multi-view construction of symbolic data (*e.g.*, shape and character). On the other hand, the OPL is the critical part of stage I to excavate the inherent category discrimination ability. It can maintain adaptive prototypes for novel categories, allowing prototype online updates and then assigning class-level pseudo labels on-the-fly.

In stage II, we empirically find that online pseudo labels generated in stage I are less effective (*i.e.*, they are unreliable for training a classifier over the whole dataset). As such, in stage II we retrain a parametric classifier for base and novel categories with three main components: pseudo labelling, prototypical pseudo label rectification, and joint optimization. The pseudo labelling leverages the discriminative feature extracted by the model trained in stage I and then generates pseudo labels by clustering. We then reuse the angular similarity of prototypes to rectify pseudo labels further. Finally, we optimize the loss of base data with human labels and novel data with offline pseudo labels. The stage II is optionally iterated to refine the classifier decision boundary and improve recognition accuracy.

Overall, the prototypical representation learning method in stage I builds a strong feature extractor for non-parametric classification via clustering. In stage II, a parametric classifier is trained by prototypical self-training. The proposed prototypical learning method facilitates improving the discrimination ability with online and offline pseudo labels. To the best of our knowledge, this is the first approach to focus more on category discrimination and effectively make use of prototypes in the NCD task. Extensive experiments on benchmark datasets including CIFAR10 [20], CIFAR100 [20], OmniGlot [21], and ImageNet [5] demonstrate the effectiveness of our method in different settings. For novel category discovery, we achieve state-of-the-art performance on the unlabelled set of all datasets. In addition, our method generalizes better than existing schemes in the testing set. With the labelled data and "pseudo-labelled" unlabelled data, our method can recognize new categories without forgetting the old ones.

2 Related Work

In this section, we first review semi-supervised and self-supervised learning methods due to close relevance to the usage of labelled and unlabelled data. Then, we introduce some methods related to transfer clustering, which motivates our designed two stages and core module of prototypical learning.

2.1 Semi-supervised Learning

Semi-supervised learning is typically used to train a model with a small amount of labelled data and a large amount of unlabelled data. Its main challenge is effectively leveraging unlabelled data to improve the model performance.

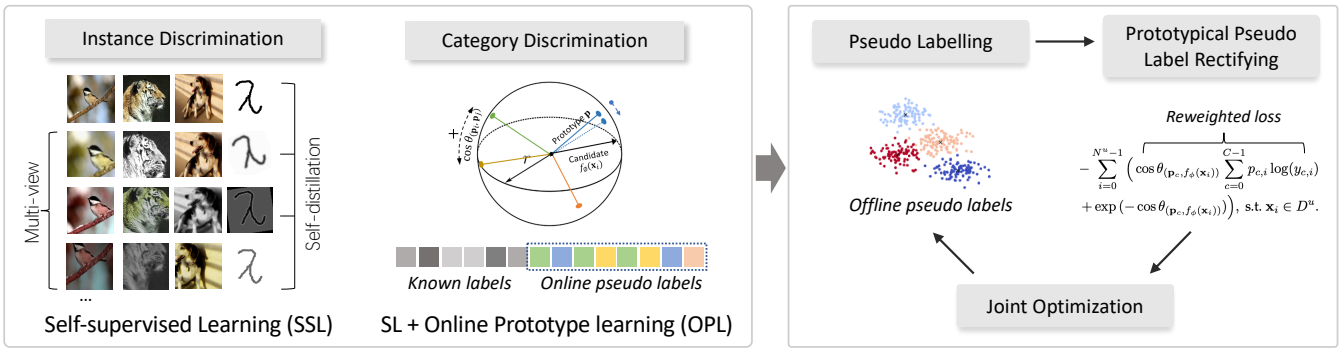
In the era of deep learning, semi-supervised learning methods can be broadly categorized as: consistency regularization [22–25] and self-training [26, 27] (*i.e.*, pseudo-labelling [28–30]). Consistency regularization methods assume that the model should be less sensitive to the different perturbations imposed on the inputs. Thus, the model predictions for the unlabelled data can be utilized as artificial labels to enforce consistency. Self-training methods first train the model on the labelled data, and then utilize it to generate pseudo labels for the unlabelled data. This pseudo-labelling process may iterate to produce better results. The idea behind self-training is direct and can be traced back to decades ago [31, 32] before the emergence of deep learning. There are also some mixed ideas between consistency regularization and self-training, such as FixMatch [33] and ISMT [34]. As the intermediate zone of unsupervised and supervised learning, semi-supervised learning has recently witnessed success in combining self-supervised pre-training and self-training [35–37].

In general, the labelled and unlabelled data in semi-supervised learning contain the same object categories. However, unlike semi-supervised learning, the NCD task requires recognizing unlabelled novel categories that are not observed in the labelled data. Thus, we need some potential classes by unsupervised clustering before applying appropriate semi-supervised methods.

2.2 Self-supervised Learning

Self-supervised learning has recently achieved significant success in natural language and computer vision without requiring expensive target labels. The core of self-supervised learning is designing some pretext tasks to obtain better representations. For example, generating future tokens [38], predicting masked tokens [39], and denoising corrupted tokens [40] are common pretext tasks in the area of natural language. For computer vision, some works use pretexts such as colorization [41], rotation prediction [42], and patch position prediction [43, 44] to learn representative features for image data.

Most recently, contrastive self-supervised learning has shown great potential by leveraging both negative and positive samples, such as InstDis [45], contrastive predictive coding (CPC) [46], AMDIM [47], MoCo [48, 49], SimCLR [35, 50], and InfoMin [51]. These methods usually pull together two augmented views of the same object (positive samples) to encourage local invariance while pushing apart those of different objects (negative samples). This strategy could prevent the model from mapping all instances to the same representation, *i.e.*, representational collapse. However, the contrastive pairs are not easy to be appropriately



(a) Stage I: Prototypical Representation Learning

(b) Stage II: Prototypical Self-training

Fig. 2. Overview of the proposed framework. It contains two training stages: prototypical representation learning (stage I) and prototypical self-training (stage II). First, stage I obtains a robust feature extractor, which could serve for all images with base and novel categories. This feature extractor is boosted in the ability of instance and category discrimination by self-supervised learning and adaptive prototypes. After that, stage II utilizes the prototypes again to rectify offline pseudo labels and then train a final parametric classifier for category clustering.

constructed and need a large batch size or memory bank for storage [52]. To solve this problem, some non-contrastive approaches are developed with only using positive pairs but achieving remarkable performance, such as BYOL [53], SwAV [54], SimSiam [55], DirectPred [52], and DINO [19]. The non-contrastive methods use a siamese-like network to match two augmented views of the same object. Typically, one network is updated online, and another is directly constructed using the online one. Without contrasting negative instances, the training process of non-contrastive methods is more efficient and conceptually simple [52, 55].

With this in mind, we choose the non-contrastive direction and specify appropriate augmentations for different types of data domains to enhance the instance discrimination in our proposed method.

2.3 Transfer Clustering

The NCD task can also be considered as a "transfer clustering" problem in [12, 16, 17]. Different from the classic or deep learning clustering problem, transfer clustering first learns the appropriate criterion by using the labelled data, then transfers such knowledge to partition the unlabelled data with novel categories. Specifically, Hsu *et al.* [15] propose to learn the category-agnostic pairwise similarity with the Kullback–Leibler divergence based contrastive loss (KCL) on the labelled data. This semantic information is then transferred to the unlabelled data by training the model with binary pseudo labels. The [14] improves KCL with a novel meta classification likelihood (MCL) loss. In [12], the deep embedded clustering (DEC) [56] is extended to conduct joint transfer clustering and representation learning. On the other hand, AutoNovel [16, 17] utilizes the pretext task of rotation predictions for self-supervised learning and transfers knowledge of labelled classes to the clustering of unlabelled data by using ranking statistics. Jia *et al.* [18] extend the noise-contrastive estimation in self-supervised representation learning to jointly handle labelled and unlabelled data.

In contrast to existing works that mainly use instance discrimination to help separate the novel visual concepts, we show that both instance and category discrimination are essential. To the best of our knowledge, we are the first to focus more on category discrimination and creatively make use of prototypes to alleviate the issues with missing labels for the NCD task.

3 Proposed Method

Given some labelled images from *base* categories, the goal of novel category discovery is automatically discovering *novel* categories in the test image collection [12, 16]. In the training stage, we have access to the labelled data D^l and the unlabelled data D^u. The images in D^l are annotated by a set of base categories C^l. $D^l = \{(\mathbf{x}_i^l, y_i^l), i = 1, ..., N^l\}$, where y_i^l is the corresponding class label for image $\mathbf{x}_i^l$ and N^l is the number of labelled data. For the images in unlabelled data D^u, they belong to the novel categories C^u. We are only aware of the number of novel categories but not the concrete meaning of each one. Note that the base and novel categories are *disjoint*, *i.e.*, $C^l \cap C^u = \varnothing$. In the inference stage, we should assign one of all categories $C^l \cup C^u$ to each image in the test split. Overall, the NCD task targets to transfer the knowledge learned from the labelled data D^l and the unlabelled one D^u to recognize novel categories C^u. This process is similar to human beings, where we could automatically differentiate some new concepts from the learned knowledge in the past.

Similar to existing schemes, our method mainly consists of two stages to solve the NCD problem, including forming a robust feature extractor and an accurate classifier. The former stage obtains effective features of images in both base and novel categories, whereas the latter aims to precise recognition. Instead of using the binary similarity [12, 14–17] in other NCD works, we are the first to exploit prototypes to enhance both stages. This prototype could help our model obtain statistics across each category whatever the base or novel. For brevity, we name our two stages prototypical representation learning and prototypical self-training.

3.1 Stage I: Prototypical Representation Learning

The goal of stage I is to obtain a robust feature extractor serving all images, irrespective of the corresponding categories

(base or novel). The extracted feature should perform well in both instance and category discrimination. We adopt the DINO model and prototype learning to improve discrimination abilities. The loss function L_{s1} of stage I is:

$$L_{\text{s1}} = L_{\text{ins}} + L_{\text{cat}}, \tag{1}$$

where the L_{ins} and L_{cat} are the losses for instance and category discrimination.

3.1.1 Instance Discrimination

We boost the model ability in instance discrimination by self-supervised representation learning. In this work, we train a model using the self-distillation with no labels (DINO) [19] scheme on labelled and unlabelled data with uniform sampling by exploiting two properties. First, the DINO model could be used as the nearest neighbour classifier for the non-parametric clustering. This property is consistent with the NCD task, which also requires clustering. Second, the DINO method converges fast by self-distillation in the training period. The parameters of the teacher are momentum updated by weighted averaging several student models in different training iterations. This procedure is like the popular AdaBoost, which obtains the most convincing results by voting various weak classifiers. Therefore, the DINO model could present a high-quality instance representation even without annotations.

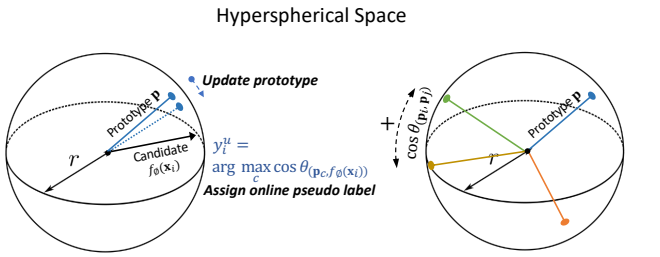
In contrast to the original DINO scheme, we propose the *restricted rotation*, a new data augmentation strategy, to satisfy different data types. In DINO or other self-supervised approaches [50, 54, 57], random cropping is widely utilized to extract intrinsic information of the image since it establishes a part-based invariance assumption which is valid for natural object-centric images like ImageNet. However, this assumption is not reasonable for symbolic data like the OmniGlot. Compared to nature images, the symbolic object has a minor appearance like the texture and color. Thus it should require a more abstract understanding of self-supervised models. In our experiments, we find that random cropping will destroy the structural information of symbolic data, leading to a dramatic performance decrease, see Sec. 4.4.1. To tackle this problem, we design an augmentation strategy named *restricted rotation*, which constructs different random rotated views for a given image in a restricted degree θ. The proposed approach injects randomness while keeping the semantic information well for the symbolic data.

We construct an image set V for each input image x. The set V contains two augmented global views x_1^g and x_2^g without any rotation, and several rotated and local augmented views x'. Then we encourage the model to learn the "local-to-global" [19] and "rotation-to-upright" correspondences from the image itself by minimizing the following loss:

$$L_{\text{ins}} = \frac{1}{N} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x')), \tag{2}$$

where N is the number of samples in a batch, P is the output distribution of the DINO head, and $H(a, b) = -a \log b$.

With the above essential representation learning, a strong baseline for NCD is established (see Sec. 4.4.1). Meanwhile, it can facilitate the category discrimination.



(a) Online Prototype Learning (b) Pair-wise Angular Separation

Fig. 3. Illustration of the online prototype learning (OPL) and pair-wise angular separation (PAS) in stage I. The prototype $\mathbf{p}$ and candidate feature $f_\phi(\mathbf{x}_i)$ are re-projected onto the hypersphere. During training, OPL provides online pseudo labels for the category discrimination and gradually updates the prototype with its corresponding candidate feature. Meanwhile, PAS pushes the prototypes far away from each other based on the angular similarity to guarantee effective discrimination.

3.1.2 Category Discrimination

Inspired by semi-supervised learning on classification or detection, the pseudo-training signals on unlabelled data could improve the recognition performance of the models due to the consistency with labelled data. Thus we aim to build a unified classifier for both base and novel categories on labelled and unlabelled data. For labelled data D^l, we can easily use cross-entropy loss to encode category information directly. However, this encoding process is impractical for the unlabelled data D^u without knowing classes. As such, we leverage prototype learning to encode the information of novel classes implicitly, thereby generating the pseudo labels online. In this way, the labelled and unlabelled data in the same label system would be used in training together.

3.1.2.1 Online Prototype Learning

The online prototype learning (OPL) module is proposed to provide online pseudo labels for unlabelled data D^u, which can be used to train a cross-entropy criterion simultaneously with labelled data D^l. Specifically, the OPL constructs a feature prototype for each novel category and generates pseudo labels for discriminative training in an online manner. During training, OPL contains the following three steps:
(1) Initializing the prototypes. At the beginning, there are no well-established classification heads and prototypes for novel categories which need some initialization. Specifically, the classification heads of novel categories are initialized with a uniform distribution weight $\mathbf{w}_c$ and no bias value $\mathbf{b}_c = 0$ where c denotes the c-th category. For ease of measuring the cosine similarity, class prototypes are initialized as the L2-normed classifier weights:

$$\mathbf{p}_c^{init} = \mathbf{w}_c^{init} / \|\mathbf{w}_c^{init}\|_2. \tag{3}$$

(2) Assigning online pseudo labels. For each batch, we calculate the cosine similarity between the c-th class prototype $\mathbf{p}_c$ and features of the i-th unlabelled data $f_\phi(\mathbf{x}_i)$ to assign pseudo labels in an online manner:

$$y_i^u = \arg\max_c \ \cos\theta_{(\mathbf{p}_c, f_\phi(\mathbf{x}_i))}. \tag{4}$$

Then, the pseudo labels y_i^u in the novel category are included to train a classifier responsible for both base and

novel categories. We minimize the standard cross-entropy loss for classification:

$$L_{cls} = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{c=0}^{C-1} p_{c,i}\log(y_{c,i}), \tag{5}$$

where C is the number of all categories in $C^l \cup C^u$, and $p_{c,i}$ is the probability of the i-th sample for the c-th class, $y_{c,i}$ is a binary value that denotes if the i-th example belongs to the c-th class.

(3) Updating online prototypes. After assigning the pseudo labels to current batch, we can update the corresponding prototypes using the output features $f_\phi(\mathbf{x}_i)$. In particular, the prototype $\mathbf{p}_c$ for the novel category c is updated using the exponential moving average:

$$\mathbf{p}_c \leftarrow \beta \cdot \mathbf{p}_c + (1-\beta)\cdot f_\phi(\mathbf{x}_i),\ \text{s.t.}\ \ \mathbf{x}_i \in D^u, \tag{6}$$

where $\beta \in [0, 1)$ is a rate parameter. As the model and its assignments become better, the old error-prone features fade, and recent data batch gradually arrives and plays a more important role. However, the norm of the prototype may no longer be unit with this modification. To facilitate the calculation of cosine similarity, we re-project the prototype onto a hypersphere with the L2-normalization (as in Eq. 3) after the update.

Avoid trivial solutions. The methods that jointly learn a discriminative classifier and assign pseudo labels would suffer from the problem of trivial solutions [58]. A similar phenomenon occurs in directly training our model with OPL. The assignments are collapsed into a single prototype, thereby leading to the classifier predicting a single class. Considering the situation, we use a uniform class distribution for initial pseudo labels at the first iteration of the training epoch.

Pair-wise Angular Separation. With the help of the L2-normalization (Eq. 3) in the prototype initialization and update, we could project prototypes onto a hypersphere. In this non-Euclidean output space, the distance is evaluated by the *angular* similarity (*i.e.*, cosine similarity) between outputs and class prototypes. Recall the goal of obtaining a discriminative feature extractor in stage I. We encourage class prototypes to have significant angular separation during prototype learning. This idea is intuitive and effective [59]: the distance between two semantically-unrelated classes would be pushed away if their corresponding class prototypes were positioned separately on the hypersphere.

Since there is no optimal separation algorithm for three- or higher-dimensional unit-hypersphere (known as the Tammes problem [60]), we optionally approximate the separation by maximizing the cosine distances of prototypes. Following [59], we define a cosine similarities loss over each pairwise prototypes:

$$L_{pas} = \frac{1}{K}\sum_{i=1}^{K}\max_{j\in C^u}\mathbf{M}_{i,j},\ \ \mathbf{M} = \mathbf{P}\mathbf{P}^\top - 2\mathbf{I} \tag{7}$$

where $\mathbf{P} \in \mathbb{R}^{K\times D}$ is the matrix of prototypes, $\mathbf{I}$ denotes the identity matrix in case of self-selection, and $\mathbf{M}$ is the final pairwise prototypes similarities. Different from [59] which defines class prototypes *a priori* with large margin separation, we use data-dependent class prototypes that are updated by instreaming novel instances. Hence the loss function can be simultaneously optimized with the learning process in stage I.

3.1.2.2 Joint Optimization

Finally, the category discrimination loss L_{cat} is computed by:

$$L_{cat} = L_{cls} + \lambda L_{pas}, \tag{8}$$

where the L_{cls} is the cross-entropy loss for classification on both D^l with human labels and D^u with pseudo labels generated from the prototype learning; and L_{pas} is the loss for pair-wise angular separation.

3.2 Stage II: Prototypical Self-training

While a robust feature extractor has been learned in stage I, we empirically find the online pseudo labels are of less quality than offline ones (see Sec. 4.4.2). Therefore, in this stage, we discard the online classifier of stage I and retrain a parametric classifier that recognizes both base and novel categories. Specifically, we utilize offline pseudo labels to conduct prototypical self-training. The reasons for using self-training in NCD are two-fold: (1) we do not have annotations of novel data D^u, while we can generate offline pseudo labels via non-parametric recognition based on stage I; (2) recent approaches on classification, detection or segmentation [27, 36] achieve improvements in self-training even after using popular self-supervised pre-training and supervised learning.

Based on the simple yet effective self-training methods [27, 28], our prototypical self-training includes three steps. First, we use the model trained in stage I to generate pseudo labels on unlabelled data (Sec. 3.2.1). Then, we note that the loss for pseudo labels would be rectified based on class prototypes (Sec. 3.2.2). Finally, we retrain the model by optimizing the classification loss on both human and offline pseudo labels (Sec. 3.2.3).

3.2.1 Pseudo labelling

Based on the well-trained feature extractor in stage I, we collect all novel images and use the k-means [61] clustering method to generate offline pseudo labels at the start of stage II. In the pseudo labelling on symbolic data, we observe features typically in the non-flat regions, such as the alphabets in OmniGlot [21]. Hence we choose the spectral clustering [62] to separate symbolic data. After clustering, we obtain cluster labels as offline pseudo labels for all novel images D^u. In addition, the central feature of clusters is re-positioned as new class prototypes.

3.2.2 Prototypical Pseudo Label Rectification

Next, we use prototypes and angular/cosine similarity to rectify the self-training from noisy pseudo labels on D^u. Instead of the hard filtering, *i.e.*, only images whose prediction confidence of pseudo label higher than a given threshold are considered in the training, we use the soft weighting for

TABLE 1
Dataset splits for novel category discovery experiments.

Dataset	# labelled cls	# unlabelled cls
OmniGlot	964	659
CIFAR10	5	5
CIFAR100	80	20
ImageNet	882	118

label rectification. We formulate the rectified objective with the angular/cosine similarity as:

$$L_{rect} = -\sum_{i=0}^{N^u-1} \cos\theta_{(\mathbf{p}_c, f_\phi(\mathbf{x}_i))} \sum_{c=0}^{C-1} p_{c,i}\log(y_{c,i}), \quad \text{s.t. } \mathbf{x}_i \in D^u. \tag{9}$$

Note that the class prototype $\mathbf{p}_c$ is fixed during the training epoch.

3.2.3 Joint Optimization

We jointly optimize the loss of both base data with human labels and novel data with offline pseudo labels. For the base data, we utilize the standard cross-entropy loss:

$$L_{ce} = -\sum_{i=0}^{N^l-1}\sum_{c=0}^{C-1} p_{c,i}\log(y_{c,i}), \text{ s.t. } \mathbf{x}_i \in D^l. \tag{10}$$

This loss is similar to Eq. 5, but here we only use instances in the labelled data D^l. Together with the rectified loss L_{rect} in Eq. 10 that uses novel categories in the unlabelled data D^u, the overall objective is

$$L_{s2} = \frac{1}{N^u + N^l}(L_{ce} + L_{rect}). \tag{11}$$

After joint optimization, we obtain the enhanced model with an explicit classification layer. This model can be reused to generate new offline pseudo labels. That is, the prototypical self-training procedure can optionally iterate to further refine the decision boundary of classifiers.

4 Experiments

4.1 Datasets and Implementation Details

4.1.1 Datasets

The proposed method is extensively evaluated on four benchmark datasets: CIFAR10 [20], CIFAR100 [20], OmniGlot [21], and ImageNet [5]. Following previous works [16, 17], the dataset splits for novel category discovery experiments are shown in Table 1. Next, we briefly introduce the datasets and describe experimental setups.

CIFAR10 and CIFAR100. There are 10 object classes in CIFAR10, and each class has 5,000 training and 1,000 testing images of 32×32 resolution. Following [16, 17], CIFAR-10 is separated into labelled and unlabelled subsets. The first 5 categories (*i.e.*, airplane, automobile, bird, cat, deer) are the labelled set, and the last 5 categories (*i.e.*, dog, frog, horse, ship, truck) are the unlabelled set. CIFAR100 also contains 50,000 training and 10,000 testing images, but with a total of 100 classes. Each class includes 500 training and 100 testing images in 32×32 resolution. For NCD, the first 80 classes are selected as labelled data, and the remaining 20 classes are used as unlabelled data.

OmniGlot. OmniGlot is a challenging dataset of handwritten characters. It contains a total of 1,623 characters from 50 different alphabets, and each alphabet has 20∼47 characters. OmniGlot splits 30 alphabets as the "background" set and 20 alphabets as the "evaluation" set. Our experimental setting for NCD follows [14] and [16]. Specifically, the 30 alphabets in the "background" are set as labelled data, including 969 characters (classes). The 20 alphabets in the "evaluation" are set as unlabelled data, which contain 659 characters. The results of the OmniGlot dataset are averaged across the 20 alphabets in the "evaluation" set.

ImageNet. ImageNet is a large-scale visual dataset that contains 1,000 classes with about 1,000 images per class. As in [63], the ImageNet dataset is randomly split into the 882-class and 118-class subsets. Following previous works [12, 14, 16, 17], we use the 882-class ImageNet as labelled data, then use three 30-class subsets (∼39k images each subset) randomly sampled from the 118-class ImageNet as unlabelled sets. As in [12, 17], the results are averaged over three 30-class subsets.

4.1.2 Implementation Details

Following previous works [12, 16–18], we use the *clustering accuracy* (ACC) that denotes the matching accuracy between ground-truth labels and clustering assignments to evaluate the performance of our method. The results are averaged over 10 runs for all datasets except the ImageNet. Our method is implemented with PyTorch 1.7.1 and runs on the NVIDIA 2080Ti GPUs.

In stage I, our method jointly performs instance and category discrimination on the labelled training data D^l and unlabelled training data D^u with online pseudo labels. Specifically, we train our model for 100 epochs on the CIFAR10, CIFAR100, OmniGlot, and ImageNet datasets with the AdamW optimizer [65]. During the first 10 epochs, the learning rate is linearly warmed up to the base value determined with the linear scaling rule: $lr = 0.0005 * \text{batchsize}/256$, in which we set $\text{batchsize} = 256$. After that, the learning rate is decayed with a cosine schedule [19, 66]. For fair comparisons to previous works, we use ResNet-18 [2] as our backbone. In Eq. 8, we set $\lambda = 0.1$ for all datasets. The embedding dimension of prototypes is set to 512. The data augmentation strategies for different domains are described in detail in Section 4.4.1.

For stage II, we use two iterations (two epochs for each iteration) of the self-training for all datasets to further improve the performance. The learning rate is set to 0.05 and decayed with a cosine schedule as the same in stage I.

4.2 Novel Category Discovery

4.2.1 Comparison with State-of-the-art Methods

We first evaluate the proposed method and compare it with other state-of-the-art methods for novel category discovery on the *unlabelled training set* of CIFAR10 and CIFAR100. Table 2 shows that our method achieves 96.0% and 78.9% ACC on CIFAR10 and CIFAR100, outperforming the previous works. Note that "k-means", shown in the first row of Table

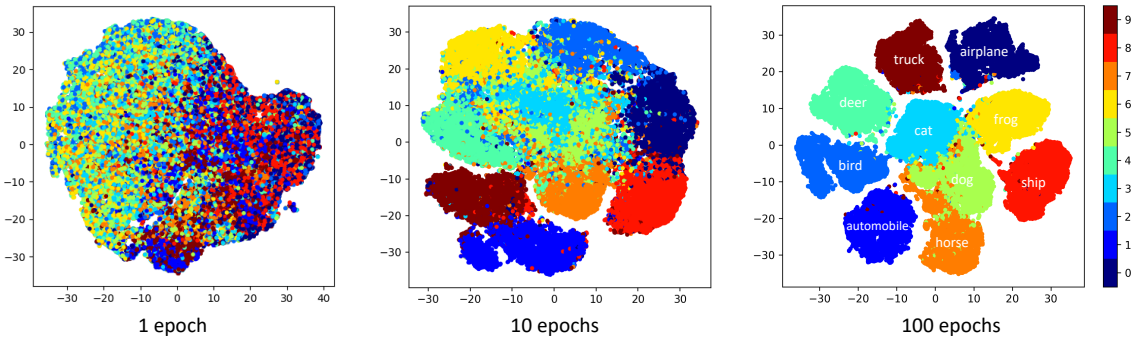


Fig. 4. Visualized features of unlabelled data in CIFAR10 using the t-SNE [64] projection. The 1, 10, and 100 epochs denote the evolution during the training phase. Colors for class numbers 5-9 refer to ground truths of five novel categories: dog, frog, horse, ship, and truck.

TABLE 2
Comparative performance for the novel category discovery on the *unlabelled training set* of CIFAR10 and CIFAR100. "w/ S.S." means with self-supervision, "w/ I.L." denotes with incremental learning.

Method	CIFAR10	CIFAR100
k-means [61]	65.5±0.0%	56.6±1.6%
KCL [15]	66.5±3.9%	14.3±1.3%
MCL [14]	64.2±0.1%	21.3±3.4%
DTC [12]	87.5±0.3%	56.7±1.2%
DTC [12] w/ S.S. [16]	88.7±0.3%	67.3±1.2%
AutoNovel [17]	90.4±0.5%	73.2±2.1%
AutoNovel [17] w/ I.L.	91.7±0.9%	75.2±4.2%
WTA-NCD [18]	93.4±0.6%	76.4±2.8%
Ours	**96.0±0.4%**	**78.9±0.9%**

2, is our baseline. It represents directly training a model using D^l and applying clustering on novel categories in D^u, which only obtains 65.5% and 56.6% ACC on CIFAR10 and CIFAR100, respectively. We also show evaluation results on the *testing set* of CIFAR10 and CIFAR100 in Table 3. This experimental setting contains old and new (*i.e.*, base and novel) categories that need to be simultaneously recognized without forgetting. As shown in Table 3, the proposed method stands out by achieving 97.1%, 93.6%, and 95.3% ACC on the old, new, and all categories of CIFAR10. In addiiton, it achieves the best 76.9%, 63.3%, and 74.2% ACC on the same setting in CIFAR100. Especially, the proposed method shows greater advantages (about 5%∼7% ACC improvements) on the testing set than the unlabelled training set, demonstrating its robust generalization ability.

Furthermore, we evaluate our method on the more challenging OmniGlot and ImageNet datasets. The results of OmniGlot are averaged over the 20 alphabets in the evaluation set, and the results of ImageNet are average over three random 30-class unlabelled subsets as in previous works [12, 14, 16, 17]. As shown in Table 4, the proposed method achieves the best performance with 93.4% and 88.8% ACC on Omniglot and ImageNet, demonstrating the effectiveness of our approach.

4.2.2 Qualitative Analysis

Next, we qualitatively analyze the proposed method for novel category discovery. Figure 4 illustrates the evolution of instance features of base (0-4) and novel (5-9) categories in CIFAR10 using the t-SNE [64]. The instance features of both base and novel categories become more separable and gradually gather into clusters. Meanwhile, some instances of the dog (5) and horse (7) are close since they maintain similar appearances as four-legged mammals. We note that mining the discrimination of such similar categories is an interesting problem and worthy of further study.

4.3 Ablation Study

To analyze the contribution of proposed components in two stages, we conduct several ablation studies on the CIFAR10, CIFAR100, and OmniGlot datasets.

Instance Discrimination. We compare performances with and without the instance discrimination module (InstDis) to validate its effectiveness for the novel category discovery. As shown in Table 5, InstDis is a basic component for learning semantic representations of the unlabelled data. For the CIFAR10, CIFAR100, and OmniGlot datasets, InstDis significantly boosts the cluster accuracy by 5.0%, 14.2%, and 3.9%, respectively.

Category Discrimination. Table 5 illustrates the effectiveness of category discrimination (CatDis). As a crucial component of the proposed method, CatDis enhances the class-level discrimination of features and consistently improves performance for the novel category discovery, *i.e.*, introducing 3.1%, 6.1%, and 2.2% ACC improvements on CIFAR10, CIFAR100, OmniGlot respectively, demonstrating the effectiveness of online prototype learning.

Prototypical Self-training. Then we ablate the training of explicit classifier in prototypical self-training (PST). It can be observed that PST is a good strategy to further boost the model's performance for novel category discovery, consistently improving 1∼3% (*i.e.*, 96.0% *v.s.* 93.6%, 78.9% *v.s.* 75.6%, 93.4% *v.s.* 92.0%) ACC on all three datasets.

4.4 Analysis and Discussion

4.4.1 Instance Discrimination

First, we analyze the performance of other state-of-the-art self-supervised methods for the instance discrimination in our model. Then we study how the augmentations used in self-supervised learning affect different image domains, *i.e.*, the natural and symbolic images.

TABLE 3
Comparative performance for recognizing both old and new categories on the *testing set* of CIFAR10 and CIFAR100.

Method	CIFAR10			CIFAR100		
	old	new	all	old	new	all
KCL [15] w/ S.S.	79.4±0.6%	60.1±0.6%	69.8±0.1%	23.4±0.3%	29.4±0.3%	24.6±0.2%
MCL [14] w/ S.S.	81.4±0.4%	64.8±0.4%	73.1±0.1%	18.2±0.3%	18.0±0.1%	18.2±0.2%
DTC [12] w/ S.S.	58.7±0.6%	78.6±0.2%	68.7±0.3%	47.6±0.2%	49.1±0.2%	47.9±0.2%
AutoNovel [17] w/ I.L.	90.6±0.2%	88.8±0.2%	89.7±0.1%	71.2±0.1%	56.8±0.3%	68.3±0.1%
Ours	**97.1±0.4%**	**93.8±0.5%**	**95.4±0.4%**	**76.9±0.3%**	**64.0±0.6%**	**74.3±0.4%**

TABLE 4
Comparative performance for the novel category discovery on Omniglot and ImageNet unlabelled set.

Method	OmniGlot	ImageNet
k-means [61]	77.2%	71.9%
KCL [15]	82.4%	73.8%
MCL [14]	83.3%	74.4%
DTC [12]	89.0%	78.3%
AutoNovel [17]	89.1%	82.5%
WTA-NCD [18]	-	86.7%
Ours	**93.4%**	**88.8%**

TABLE 5
Ablation studies of the proposed method. "InstDis" and "CatDis" stand for instance discrimination and category discrimination in stage I, respectively. "PST" refers to the prototypical self-training in stage II. All methods use the same hyperparameters and are evaluated with clustering accuracy (ACC).

Method	CIFAR10	CIFAR100	OmniGlot
Ours w/o InstDis	91.0%	64.7%	89.5%
Ours w/o CatDis	92.9%	72.8%	91.2%
Ours w/o PST	93.6%	75.6%	92.0%
Ours	**96.0%**	**78.9%**	**93.4%**

TABLE 6
The performance comparison with different self-supervised learning methods for the instance discrimination in NCD.

Method		CIFAR10	CIFAR100
k-means [61]	SimCLR [50]	85.2%	49.7%
	MoCo [48]	77.4%	51.2%
	MoCo v2 [49]	81.3%	54.1%
	DINO [19]	**89.3%**	**69.6%**
Ours	SimCLR [50]	92.1%	61.8%
	MoCo [48]	90.9%	62.9%
	MoCo v2 [49]	93.8%	65.3%
	DINO [19]	**96.0%**	**78.9%**

Alternative self-supervised methods. As discussed in Section 3.1.1, DINO [19] is adopted in our method to enhance the instance discrimination. However, other self-supervised learning methods can be inserted into our model. Since DINO is a non-contrastive method, we choose other state-of-the-art contrastive methods as alternatives, including SimCLR [50], MoCo [48], and MoCo v2 [49]. In Table 6, we first compare the learned features of different self-supervised methods using k-means [61] as the clustering method. DINO significantly outperforms other alternatives as it is an effective nearest neighbour classifier without any fine-tuning or linear classifier [19]. This is a desirable property for k-means clustering. Then, when combined with our method, DINO is further improved by 6.7% and 9.3% on CIFAR10 and CIFAR100, respectively. In addition, SimCLR and MoCo are also significantly boosted and achieve comparable performance, which validates the effectiveness of our method and its compatibility with other self-supervised learning methods.

Different domains and augmentations. We consider two main categories of domains [67], natural (CIFAR10) and symbolic (Omniglot), for experiments. To systematically study the impact of data augmentation, we follow SimCLR [50] to divide augmentations into two types. One type of augmentation involves *geometric/spatial* transformation of data, such as cropping, flipping, and the proposed restricted rotation. The other type of augmentation involves *appearance* transformation, such as color distortion (including color jittering and dropping, solarization) and Gaussian blur. We report the performance of different compositions of data augmentations for both natural and symbolic data in Table 7. For symbolic data, the proposed restricted rotation and appearance transformation contribute a lot to self-supervised learning. In contrast, geometric transformations lead to significant performance degradation, since they would negatively affect the structure of symbolic data. On the other hand, natural data benefits from the composition of appearance and geometric transformation, as also noted in [50]. Unlike symbolic data, natural data do not need strict structure preservation, hence we do not apply the restricted rotation to them. In Table 8, we show the set of augmentations in our implementation for natural and symbolic datasets, respectively.

4.4.2 Category Discrimination

As online prototype learning plays a crucial role in the category discrimination of stage I, we analyze the hyperparameters of prototypes and compare pseudo labels provided by prototype assignment and k-means.

Hyperparamters of prototypes. In the online prototype learning module, a representative feature of novel categories is maintained as the prototype. First, we study how the embedding dimensions of prototypes affect the performance on CIFAR10. As shown in Figure 5, the proposed method

TABLE 7
The performance comparison with different types of data augmentations in the self-supervised feature learning for both natural and symbolic datasets. Green and blue colors denote the best two results.

Dataset	Appearance	Geometric	ACC (%)
OmniGlot (symbolic)	✓		90.2
	✓	R.R.	93.4
		✓	66.3
	✓	✓	88.4
	✓	✓ + R.R.	91.3
CIFAR10 (natural)	✓		79.5
	✓	R.R.	81.7
		✓	65.9
	✓	✓	96.0
	✓	✓ + R.R.	95.4

TABLE 8
Different sets of augmentations (transformations) in the self-supervised feature learning for the natural and symbolic domains.

Augmentation for S.S	Natural	Symbolic
Brightness, contrast, hue, and saturation adjustment	✓	✓
Random cropping	✓	×
Random left-right flipping	✓	×
Restricted rotation	×	✓
Color jittering and dropping	✓	✓
Gaussian blurring, solarization	✓	✓

is stable with different prototype dimensions, and the performances plateau when dimensions are greater than 128. Next, we show different similarity metrics for the prototype when assigning online pseudo labels. Compared to the Euclidean distance and dot product, our cosine similarity metric achieves the best accuracy on both natural and symbolic datasets, as shown in Table 9.

Prototype assignment and k-means. As two potential ways to generate pseudo labels in stage I, we compare the prototype assignment and k-means in Table 10. With the proposed OPL, the results of k-means become more accurate. Actually, k-means is a simple but effective clustering method to evaluate the distinctiveness of our feature extractor in stage I. Table 10 also shows that the "online (prototype assignment)" and "offline (k-mean, ours full)" have comparable accuracy, and the latter slightly outperforms the former at the end of training. This can be attributed to that offline pseudo labels could see the whole dataset, whereas online pseudo labels are generated batch by batch. Therefore, we use offline pseudo labels provided by "k-means, ours full" as the bridge between stages I and II. In addition, we note that only online (prototype-based) pseudo labels are used to train the discriminative feature extractor in stage I. This is motivated by two factors: (1) online pseudo labels are in a more efficient batch-by-batch way, while k-means need to see the whole dataset to calculate the distance between every two data instances and conduct an iterative EM-like process, which is relatively time-consuming; and (2) the separation constraint of online prototypes, *i.e.*, pairwise angular separation, can also contribute to learning discriminative representations.

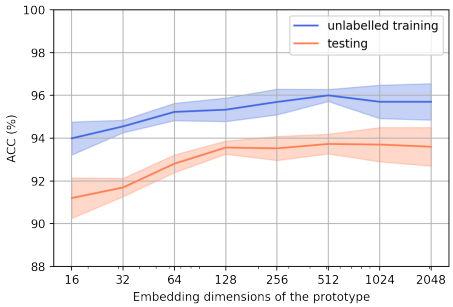


Fig. 5. Illustration of the performance evolution for different embedding dimensions of the prototype. Experiments are performed on CIFAR10.

TABLE 9
The performance comparison with different similarity metrics for the prototype when assigning online pseudo labels.

Similarity	CIFAR10	OmniGlot
Euclidean	95.4%	91.8%
Dot Product	94.7%	92.0%
Cosine (ours)	**96.0%**	**93.4%**

TABLE 10
The performance comparison with online and offline pseudo-labels after stage I on CIFAR10.

Different pseudo-labels at the end of stage I	ACC
Offline (k-means, ours full)	**93.6%**
Offline (k-means, ours w/o OPL)	91.1%
Online (prototype assignment)	90.5%
Offline (k-means)	65.5%

4.4.3 *Prototypical Self-training*

Based on the discriminative feature extractor in stage I, we collect all novel images and run the global clustering via the off-the-shelf methods (*e.g.*, k-means) to generate offline pseudo labels at the start of stage II (PST). Note that the clustering method is conducted only once at the beginning of PST. During the iteration of PST, offline pseudo labels are provided by the explicit classifier for novel categories.

The number of iterations. We report the performance with different numbers of iterations in Table 11. At first, the performance significantly increases with the prototypical self-training, then it plateaus after two iterations. Hence we use two iterations of prototypical self-training in our implementation, as described in Section 4.1.2.

Confusion matrix. We compare the confusion matrices w/ and w/o PST in Figure 6. The accuracy for novel categories is improved with PST. Similar to the t-SNE visualization in Figure 4, there is confusion between the dog and horse, mainly due to their similar shape and appearance. Therefore, separating similar or fine-grained categories in NCD is of great interest for future study.

TABLE 11
The performance evolution for different times of iterations with the PST.

# Iters	0	1	2	3	4
CIFAR10	93.6%	95.6%	96.0%	**96.1**%	96.0%
OmniGlot	92.0%	92.5%	**93.4**%	93.3%	93.3%

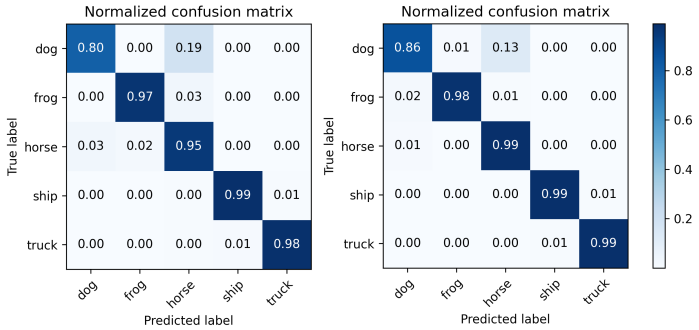


Fig. 6. Confusion matrix on 5 novel categories of CIFAR10. Left: our method stage I; Right: our method with PST (stage II).

5 Conclusion

In this paper, we propose a novel method to improve the category discrimination for semantic partition in the NCD task. Our approach consists of two stages: (1) prototypical representation learning and (2) prototypical self-training. Specifically, we leverage the prototype to conduct both instance and category discrimination at stage I, thereby obtaining a robust feature extractor to serve all base and novel images. Then, we train a parametric classifier by self-training with prototypical rectified pseudo labels at stage II. Extensive experiments on widely-used benchmark datasets show that the proposed method achieves state-of-the-art performance, and demonstrates the effectiveness and robustness of all modules. In the future, we plan to explore NCD by including or improving the open-set recognition and incremental learning, and develop a framework for discovering more novel concepts in the real world.

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 1, 6

[3] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference (BMVC)*, 2015. 1

[4] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 212–220. 1

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255. 1, 2, 6

[6] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2013, pp. 397–403. 1

[7] E. Orhan, V. Gupta, and B. M. Lake, "Self-supervised learning through the eyes of a child," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9960–9971, 2020. 1

[8] N. Krishnaswamy and S. Ghaffari, "Exploiting embodied simulation to detect novel object classes through interaction," *arXiv preprint arXiv:2204.08107*, 2022. 1

[9] P. C. Bomba and E. R. Siqueland, "The nature and structure of infant form categories," *Journal of Experimental Child Psychology*, vol. 35, no. 2, pp. 294–328, 1983. 1

[10] P. C. Quinn, P. D. Eimas, and S. L. Rosenkrantz, "Evidence for representations of perceptually similar natural categories by 3-month-old and 4-month-old infants," *Perception*, vol. 22, no. 4, pp. 463–475, 1993. 1

[11] E. Colung and L. B. Smith, "The emergence of abstract ideas: Evidence from networks and babies," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1435, pp. 1205–1214, 2003. 1

[12] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 8401–8409. 1, 3, 6, 7, 8

[13] T. Serre, "Deep learning: the good, the bad, and the ugly," *Annual review of vision science*, vol. 5, no. 1, pp. 399–426, 2019. 1

[14] Y.-C. Hsu, Z. Lv, J. Schlosser, P. Odom, and Z. Kira, "Multi-class classification without multi-class labels," in *International Conference on Learning Representations (ICLR)*, 2019. 1, 3, 6, 7, 8

[15] Y.-C. Hsu, Z. Lv, and Z. Kira, "Learning to cluster in order to transfer across domains and tasks," in *International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 7, 8

[16] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, "Automatically discovering and learning new visual categories with ranking statistics," in *International Conference on Learning Representations (ICLR)*, 2020. 1, 3, 6, 7

[17] ——, "Autonovel: Automatically discovering and learning novel visual categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3, 6, 7, 8

[18] X. Jia, K. Han, Y. Zhu, and B. Green, "Joint representation learning and novel category discovery on single-and multi-modal data," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 6, 7, 8

[19] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," *arXiv preprint arXiv:2104.14294*, 2021. 2, 3, 4, 6, 8

[20] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Technical report*, 2009. 2, 6

[21] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015. 2, 5, 6

[22] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, pp. 3365–3373, 2014. 2

[23] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, pp. 1163–1171, 2016.

[24] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[25] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. 2

[26] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05)-Volume 01*, 2005, pp. 29–36. 2

[27] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 687–10 698. 2, 5

[28] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013, p. 896. 2, 5

[29] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," in *International Conference on Learning Representations (ICLR)*, 2020.

[30] L. Qi, J. Kuen, Z. Lin, J. Gu, F. Rao, D. Li, W. Guo, Z. Wen, and J. Jia, "Casp: Class-agnostic semi-supervised pretraining for detection and segmentation," *arXiv preprint arXiv:2112.04966*, 2021. 2

[31] G. J. McLachlan, "Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis," *Journal of the American Statistical Association*, vol. 70, no. 350, pp. 365–369, 1975. 2

[32] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965. 2

[33] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020. 2

[34] Q. Yang, X. Wei, B. Wang, X.-S. Hua, and L. Zhang, "Interactive self-training with mean teachers for semi-supervised object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5941–5950. 2

[35] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 22 243–22 255, 2020. 2

[36] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020. 5

[37] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021. 2

[38] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018. 2

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186. 2

[40] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880. 2

[41] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 649–666. 2

[42] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations (ICLR)*, 2018. 2

[43] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1422–1430. 2

[44] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 69–84. 2

[45] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3733–3742. 2

[46] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018. 2

[47] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 15 535–15 545, 2019. 2

[48] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representa-

tion learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738. 2, 8

[49] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020. 2, 8

[50] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 1597–1607. 2, 4, 8

[51] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *arXiv preprint arXiv:2005.10243*, 2020. 2

[52] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," *arXiv preprint arXiv:2102.06810*, 2021. 3

[53] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3

[54] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 4

[55] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3

[56] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning (ICML)*, 2016, pp. 478–487. 3

[57] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6707–6717. 4

[58] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149. 5

[59] P. Mettes, E. van der Pol, and C. Snoek, "Hyperspherical prototype networks," *Advances in neural information processing systems*, vol. 32, 2019. 5

[60] P. M. L. Tammes, "On the origin of number and arrangement of the places of exit on the surface of pollen-grains," *Recueil des travaux botaniques néerlandais*, vol. 27, no. 1, pp. 1–84, 1930. 5

[61] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297. 5, 7, 8

[62] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, 2001. 5

[63] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, pp. 3630–3638, 2016. 6

[64] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008. 7

[65] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," 2018. 6

[66] ——, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016. 6

[67] B. Wallace and B. Hariharan, "Extending and analyzing self-supervised learning across domains," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 717–734. 8