MetaGCD: Learning to Continually Learn in Generalized Category Discovery

Yanan Wu¹,²; Zhixiang Chi³; Yang Wang⁴; Songhe Feng¹,² †
¹Key Laboratory of Big Data & Artificial Intelligence in Transportation, Ministry of Education, Beijing Jiaotong University, Beijing, 100044, China
²School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China
³Department of Electrical and Computer Engineering, University of Toronto, Toronto, M5G1V7, Canada
⁴Department of Computer Science and Software Engineering, Concordia University, Montreal, H3G2J1, Canada
{ynwu0510,shfeng}@bjtu.edu.cn, zhixiang.chi@mail.utoronto.ca, yang.wang@concordia.ca

Abstract

In this paper, we consider a real-world scenario where a model that is trained on pre-defined classes continually encounters unlabeled data that contains both known and novel classes. The goal is to continually discover novel classes while maintaining the performance in known classes. We name the setting Continual Generalized Category Discovery (C-GCD). Existing methods for novel class discovery cannot directly handle the C-GCD setting due to some unrealistic assumptions, such as the unlabeled data only containing novel classes. Furthermore, they fail to discover novel classes in a continual fashion. In this work, we lift all these assumptions and propose an approach, called MetaGCD, to learn how to incrementally discover with less forgetting. Our proposed method uses a meta-learning framework and leverages the offline labeled data to simulate the testing incremental learning process. A meta-objective is defined to revolve around two conflicting learning objectives to achieve novel class discovery without forgetting. Furthermore, a soft neighborhood-based contrastive network is proposed to discriminate uncorrelated images while attracting correlated images. We build strong baselines and conduct extensive experiments on three widely used benchmarks to demonstrate the superiority of our method.

1. Introduction

Object categories in real-world environments are dynamically evolving and expanding over time. However, conventional deep learning-based visual recognition methods normally focus on closed-world scenarios with pre-defined categories [19, 37]. Such systems are brittle when deployed to an ever-changing realistic open-world setting, where object instances may come from new categories. In contrast, recognizing the known categories and utilizing them to discern the unknowns are intrinsic to human perception.

Recently, discovering the novel classes among unlabeled data has been an active area of research [15, 11, 40, 35, 45]. However, most prior works make several assumptions that are unrealistic in practice. For example, the works in [15, 11, 40] assume the co-existence of both labeled data (with known classes) and unlabeled data (contains potential unknown classes to be discovered) at the training phase and
the models are learned from scratch. This leads to repetitive large-scale training every time when new classes are expected to be discovered. The works in [15, 11, 35, 45] assume the newly encountered unlabeled data only belongs to the novel classes. This is unrealistic in practice. To meet such conditions, a rigorous filtering method is needed to precisely filter out known class data to avoid degenerate solutions. Due to these limitations, none of these works can be used to build recognition systems that can deal with evolving object categories sequentially over a long time horizon.

In this paper, we consider a more flexible setting for real-world applications. Let us consider the application of home robots. The robots are equipped with an offline trained object recognition model on pre-defined categories during manufacturing. After deployment, the robots are expected to operate in diverse environments. While operating, they continually receive data that belongs to known and possibly unknown classes. Ideally, we would like the robots to continually discover and learn novel classes from such data. We dub such a setting as Continuous Generalized Class Discovery (C-GCD). As shown in Fig. 1, C-GCD has two phases: 1) an offline training phase that allows the model to be trained on large-scale labeled data with pre-defined classes; 2) when the model is deployed, it continually encounters unlabeled data that comes from both known and novel classes on a longer horizon. At each incremental session, the data from the previous sessions is inaccessible. The model needs to precisely classify the known classes and discover novel ones to expand its knowledge base. Obviously, the main challenge of C-GCD is to discover the novel classes among unlabeled images that contain both known and unknown categories while maintaining the performance on old classes. However, learning novel knowledge normally leads to notorious catastrophic forgetting [7], which further exacerbates the model performance.

There are some initial attempts on C-GCD [21, 45]. However, they only consider C-GCD at the deployment stage mentioned above. The offline training stage is not fully exploited in these works. Concretely, the labeled data during offline training is only used for pre-training model representations. Therefore, the model at the offline stage is unaware of its subsequent learning duty (discover novel classes and retain the performance of known classes) [8] and is prone to overfit to the labeled set [40]. Such learning objective misalignment leads to cumbersome heuristic strategies to facilitate the new learning task while keeping the previous knowledge. For example, to learn the novel classes, [21] requires a self-labeling method, which may cause error propagation. A routing strategy is also required to determine the known and novel classifier heads. [45] relies on a thresholding method to filter novel instances. However, the overall robustness of the method can be sensitive to the threshold. To alleviate the forgetting issues, [21, 45] propose to distill the knowledge from the pre-trained base models. Data replay is also utilized to either directly select representative labeled examples [45] or generate pseudo-latent representations from them [21]. Consequently, the base models and the replay buffers have to be stored locally which may cause storage problems, especially in a resource-constraint environment.

In this work, we propose a fully learning-based solution, named MetaGCD, to minimize the hand-engineered heuristics in prior works. Concretely, at the offline training phase, instead of pre-training a model representation, we directly train an initialization that is learned to discover novel categories with less forgetting when deployed. It is realized by meta-learning-based bi-level optimization [12] to couple the offline training and downstream learning objectives. During the offline training, we simulate the testing scenario and construct pseudo incremental novel class discovery sessions using the labeled data. At each incremental session, we discover novel classes by updating the model using an unsupervised contrastive loss. The meta-objective is then defined by validating the updated model on all classes encountered on a labeled pseudo test set. Therefore, the meta-objective of the offline training is aligned with the evaluation protocol at deployment. It enforces the model to learn to balance two conflicting objectives, namely discovering new objects and not-forgetting old objects. The meta-objective also reinforces the unsupervised updated model to be supervised by the true labels to ensure valid novel class discovery.

MetaGCD uses unsupervised contrastive learning to explore the relationship among instances for novel class discovery. Therefore, it is less prone to label overfitting [40]. However, we observe that the negative pairs in contrastive learning normally dominate the loss function. So we further propose soft neighborhood contrastive learning to mine more positiveness. Concretely, for each image instance, we select the nearest candidate neighbors within the batch to treat them as soft positive samples to contribute to the discriminative feature learning. Overall, our contributions are summarised as follows:

- We consider a realistic setting C-GCD for applications in real-world scenarios. It allows the model trained on pre-defined classes to continually explore novel classes through incoming unlabeled data while simultaneously keeping the performance of known classes.
- We propose a meta-learning approach where the learning objective is well aligned with the evaluation protocol during testing. It directly optimizes the model to achieve novel object discovery without forgetting.
- A soft neighborhood contrastive learning method is also proposed to mine more soft positive pairs to elevate the discovery capability.
- We establish strong baselines and show that our method achieves superior performance with less hand-
engineered design through extensive experiments.

2. Related Work

Discovering novel classes. Novel Class Discovery (NCD) aims to discover the novel classes from unlabeled data by utilizing the prior knowledge from the labeled data [15, 11, 17]. However, NCD assumes that the unlabeled data only belongs to the novel classes, which is unrealistic. Alternatively, a generalized version of NCD (GCD) [40] relaxes such constrain. Although GCD allows the unlabeled data to contain both known and novel classes, they are both required to be present in the training phase. It leads to repetitive large-scale training when different groups of unlabeled data are continually presented to the recognition system. Recently, a class incremental variant of NCD (class-iNCD) is proposed to learn the tasks of labeled known and unlabeled novel classes sequentially [35]. When learning the novel classes, the data of old classes is inaccessible. In the end, the model is evaluated on all encountered classes. Nevertheless, only a few incremental sessions containing unlabeled novel classes are allowed in class-iNCD. This limitation hinders its applicability under the realistic setting with continually evolved object categories. Our proposed C-GCD alleviates the above limitations in real-world scenarios. Our approach can learn from labeled pre-defined classes during offline training, and then continuously encounter unlabeled data with both known and novel classes after deployment. Our model will learn to discover novel classes without forgetting old classes. C-GCD is also related to the classic class-incremental setting [26, 41, 31, 29]. But C-GCD is more challenging as the newly evolved classes are unlabeled and an automatic class discovery mechanism is required [21].

Meta-learning. Existing meta-learning methods can be categorized into: 1) model-based [36, 2, 4]; 2) Optimization-based [34, 12, 5]; and 3) metric-based [38]. Typical meta-learning methods utilize bi-level optimization to train a model that is applicable for downstream adaptations. Our work is built upon MAML [12], which trains a model initialization through episodes of tasks for fast adaptation via gradient updates. Such learning paradigm has been widely applied in different vision tasks, such as test-time adaptation [39, 28, 32, 30], continual learning [43, 20, 42] and domain shift [44, 46, 13]. In our case, the adaptation is achieved in an unsupervised manner, and the bi-level optimization is utilized to combine two conflicting learning objectives: discovering the novel classes without forgetting the old classes.

Contrastive learning. Contrastive learning has been popular in self-supervised visual representation learning [1, 6, 18, 33, 27]. It explores the relationships among data instances by constructing positive and negative pairs. Therefore, the overfitting on the label space is reduced to prove the generalization of downstream tasks. Zhong et al. [47] apply contrastive learning to discover novel classes by exploring the data neighborhood and choosing pseudo-positive pairs. However, those pseudo-positive pairs contribute equally regardless of their closeness compared to the reference sample. In this work, we introduce the soft positivity concept to allow adaptive contribution.

3. The Proposed Method

Problem definition. The goal of C-GCD is to have the offline trained model continually discover and learn novel object classes from unlabeled data containing both known and novel classes. We define a sequence of $T$ learning sessions $\{S^0, S^1, \cdots, S^T\}$. Let $x^t \in X^t$ and $y^t \in Y^t$ denote the input and label space at session $t$. We represent each session as: $S^0 = \{(x^0_i, y^0_i)\}_{i=1}^{n}$ and $S^t = \{(x^t_i)\}_{i=1}^{n}$. Note, only the first session (i.e., $t = 0$) contains large-scale labeled samples. As for $t > 0$, $S^t$ only contains unlabeled data. At the $t^{th}$ session, only $S^t$ is accessible, and the incoming data belongs to both learned known class from previous sessions and novel classes. Therefore, we can denote $Y^t = Y^{t-1} \cup Y^t_n$, where $Y^t_n$ represents the novel classes to be discovered at session $t$. After learning on $S^t$, the model is evaluated on all test images accumulated until session $t$ to test the performance on $Y^{t-1}$ (ideally the model should not forget old classes) and the discovery capability on $Y^t_n$. Compared with previous works [15, 11, 35, 45], C-GCD is much more challenging due to several factors. First, the unlabeled data contains both known and unknown classes, i.e., $Y^t = Y^{t-1} \cup Y^t_n$. Second, labeled data is absent at $t > 0$, i.e., $S^0 \cup S^t = \emptyset$ where $t > 0$. Finally, since C-GCD operates on a long horizon, i.e., $t \gg 1$, the catastrophic forgetting issue is more severe.

Method overview. Fig. 2 shows an overview of MetaGCD. Following [40], we use a model without parametric classification heads since it is more suitable for dealing with novel classes. Novel class discovery is performed by directly clustering the feature spaces and class labels are assigned through the classic $k$-means algorithm. Concretely, we learn a model initialization using the labeled data during offline training. During each continual learning session, we update the model using a soft neighborhood contrastive learning (see Fig. 2 (a)) on unlabeled data. To fully exploit the labeled data in offline learning, we further develop a bi-level optimization based on meta-learning to simulate the online learning scenario, so that the model is ready to adapt to new incoming unlabeled data and discover novel objects after the offline training (see Fig. 2 (b)). In the following, we describe these two parts of our method in detail.

3.1. Contrastive learning based clustering network

Considering the characteristics of labeled and unlabeled data, we employ different contrastive learning strategies. To
Figure 2: **Overview of the proposed MetaGCD.** (a) Our soft neighborhood contrastive learning network aims to discriminate uncorrelated instances while absorbing correlated instances to learn discriminative representations. (b) Our meta-learning optimization strategy utilizes the offline labeled data to simulate the testing incremental learning process by sampling sequential learning tasks. By learning from these sampled sequential tasks, our model learns a good initialization, so that it can effectively adapt to discover new novel classes without forgetting old classes.

3.1.1 Representation learning on labeled data

To learn a robust and semantically meaningful representation on labeled data, we utilize both self-supervised [14] and supervised [22] contrastive losses. When discovering the latent classes in continually encountered unlabeled data, we propose to mine soft positive neighbors for each data instance to elevate the discriminative feature learning.

The supervised contrastive counterpart is expressed as:

$$
\mathcal{L}_{i}^{scl} = - \frac{1}{|\mathcal{N}(i)|} \sum_{q \in \mathcal{N}(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_q / \tau)}{\sum_{n \in \mathcal{N}(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_n / \tau)}
$$

where $\mathcal{N}(i)$ denotes the indices of instances having the same label as $x_i$ within the batch. Finally, these two losses are weighted by $\lambda$ to train on the labeled data:

$$
\mathcal{L}_{labeled} = (1 - \lambda) \sum_{i \in B} \mathcal{L}_{i}^{scl} + \lambda \sum_{i \in B} \mathcal{L}_{i}^{ucl}
$$

3.1.2 Soft neighborhood contrastive learning on unlabeled data

When learning on unlabeled data, only the unsupervised contrastive loss Eq. 1 can be used. However, the samples within the same class could be mistakenly treated as negatives due to the missing labels. In addition, the number of negative pairs significantly surpasses positive pairs. Such imbalanced loss contribution could be sub-optimal. Aligning the positive and negative pairs with the true classes...
emerges as a desired solution. [47] has attempted to address the limitations by mining more positive pairs in the neighbor-
hood of each data sample. However, the pseudo-positive pairs are treated equally, regardless of how close they are to that data sample. To address this issue, we propose to en-
code soft positive correlation among instance neighbors to achieve adaptive contribution, as shown in Fig. 2 (a).

Specifically, for each \( x_i \), we first use the nearest neighbor operator on the projected features to select candidate neighbors. We denote them as \( NN(z_i) \), where we then pass them and \( z_i \) to an attention module to predict a set of positiveness values \( w_i \) \( \in (0,1) \) to weight the contribution of \( NN(z_i) \) to the loss. Accordingly, the soft neighborhood contrastive loss is defined as:

\[
L_{soft}^{s} = -\frac{1}{|NN(z_i)|} \sum_{k \in NN(z_i)} \log \frac{w_{ik} \cdot \exp(z_i \cdot z_k / \tau)}{\sum_n \exp(z_i \cdot z_n / \tau)}
\]

Candidate neighborhood. For each batch of data, we first compute their features \( F \) from the projection head at once. For each reference view \( x_i \), we retrieve nearest neighbors by comparing the cosine similarity to a threshold \( \epsilon \):

\[
NN(z_i) = \{ F \}, \text{ for } F \in F, \text{ if } \cos(z_i, F) \geq \epsilon
\]

where \( z_i \) and \( F \in F \) are normalized before computation.

Positiveness generation. Fig. 2 (a) shows an intuitive example of the candidate neighbors. The first two neighbors belong to the same category as the reference ‘tiger’ sample, while the 3rd and 4th neighbors are partially related (i.e., they belong to the ‘lion’ and ‘leopard’ categories, but not ‘tiger’). The remaining instances are not related to ‘tiger’. Therefore, the first four instances tend to be selected and they should contribute adaptively to the loss. We propose to learn an attention module to measure the soft correlations between the selected neighbors and the reference instance (instead of the binary in [47]). Given two inputs \( z_i \) and \( NN(z_i) \), we can calculate the positiveness score as:

\[
w_i = \text{Softmax}[f_1(z_i) \times f_2(NN(z_i)k)]
\]

where \( f_1(\cdot) \) and \( f_2(\cdot) \) are the new projection layers and \( \times \) denotes the cross attention operator. Eq. 6 is then normalized so that \( w_i \) has a max value as 1. Note that \( f_1(\cdot) \) and \( f_2(\cdot) \) can also be the non-parametric identity mappings, which are empirically found to be more effective. This observation may be attributed to the self-supervised learning paradigm, where the objective is to train the encoder effectively. Simplifying the attention module leads to less over-
fitting and improves learning attentive features.

3.2. Learning to incrementally discover categories

The main limitation of the prior works is that the la-
beled set \( S^0 \) is not fully exploited [7]. Instead of perform-
ing only representation learning on \( S^0 \) [21, 45], we borrow the meta-learning paradigm (in particular MAML [12]) to learn how to continually discover new classes. In few-shot learning, during meta-training, MAML constructs few-shot tasks to mimic the meta-testing scenario to achieve learning to quickly adapt. In our C-GCD case, the online continual class discovery tasks can be viewed as the “meta-testing” stage. Therefore, we propose to simulate the continual setting using \( S^0 \) during offline training, as shown in Fig. 2 (b).

We aim to produce a model initialization that is trained by aligning the training and evaluation objectives so that it is endowed with the capability to effectively discover novel classes with less forgetting during evaluation.

Sequential task sampling. To mimic the evaluation pro-
cess, we sample sequential learning tasks from \( S^0 \) [7]. Specifically, we first randomly separate the \( S^0 \) into pseudo labeled and pseudo unlabeled classes without overlapping. Next, we sample a sequence of \( T \) + 1 sessions, \( D = \{(D_{tr}^j, D_{te}^j)\}_{j=0}^T \), where \( D_{tr}^j \) and \( D_{te}^j \) are the training and test set for the \( j \)-th session. For the training splits \( D_{tr}^j \), we follow the evaluation protocol to only allow the first session to contain labeled data, (i.e., \( D_{tr}^0 = \{x_{tr}^0, y_{tr}^0\} \)) and the rest with unlabeled data (i.e., \( D_{tr}^j = \{x_{tr}^j\}, \forall j \geq 0 \)). We also set the first session to contain a larger number of samples, i.e., \( |D_{tr}^0| \gg |D_{tr}^j| \) to simulate the C-GCD setting where the model is first trained during an offline training stage with a large amount of data. For the test splits \( D_{te}^j \), all of them contain labels that will be used during the optimization (i.e., \( D_{te}^j = \{x_{te}^j, y_{te}^j\}, \forall j \)). Note that \( D_{te}^0 \) only contains the test data belonging to the current session \( j \).

Meta-training. For each sampled sequence \( D \), let we model continually explore the incoming unlabeled data in

```
Algorithm 1 The optimization procedure of MetaGCD

Require: \( \alpha, \beta, \gamma \): learning rates

Require: \( S^0 \): training set of labeled classes
1: randomly initialize parameters \( \theta \)
2: while not converged do
3: \( D = \{(D_{tr}^j, D_{te}^j)\}_{j=0}^T \)
4: \( \mathcal{P} = \emptyset \)
5: \( \mathcal{P} = \emptyset \)
6: \( \theta^E,P \leftarrow \theta^E,P - \gamma \nabla_{\theta^E,P} L_{soft}(x_{tr}^j, y_{tr}^j; \theta^E) \)
7: update parameters using pseudo labeled classes
8: \( \mathcal{P} = \mathcal{P} \cup D_{te}^0 \)
9: \( \{ x_j, \forall x \in \mathcal{P} \} \)
10: \( \theta^E,P \leftarrow \theta^E,P - \alpha \nabla_{\theta^E,P} L_{soft}(x_{tr}^j, \theta^E) \)
11: \( \theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{(x_i,y_i) \in \mathcal{P}} L_{cl}(x_i, y_i; \theta^E, \theta^P) \)
12: \( \mathcal{P} = \mathcal{P} \cup D_{te}^j \)
13: \( \theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{(x_i,y_i) \in \mathcal{P}} L_{cl}(x_i, y_i; \theta^E, \theta^P) \)
14: update meta-params \( \theta \) to new session
15: end for
16: end while
```
an unsupervised manner. To reduce the forgetting issue due to learning new knowledge, we utilize the bi-level optimization [12, 8, 7] to directly formulate incrementally discovering without forgetting as the meta-objective. The meta-learning procedure is illustrated in Alg. 1 and Fig. 2 (b). Concretely, we decouple the network as \( \theta = \{ \theta^E, \theta^P \} \), where \( \theta^E \) and \( \theta^P \) are the encoder and projection layers. At each incremental session, we aim to evaluate all the classes that have encountered so far. Hence, at the beginning of each sequence, we define an empty cumulative pseudo test set \( \mathcal{P} \) to store the test samples from previous sessions. After that, we first train \( \theta \) on the pseudo labeled classes \((j = 0)\) using the unsupervised and supervised contrastive loss (Eq. 3). At each \( j^{th} \) session \((j > 0)\), we update \( \theta \) on unlabeled samples \( \mathcal{D}^t_{tr} = \{ \mathbf{x}^t_{tr} \} \) via a few gradient steps:

\[
\hat{\theta}^E, P = \theta^E, P - \alpha \nabla_{\theta^E, P} \mathcal{L}_{soft}(\mathbf{x}^t_{tr}; \theta)
\]

(7)

where \( \mathcal{L}_{soft}(\cdot) \) is the proposed soft neighborhood contrastive loss (Eq. 4). It aims to discriminate uncorrelated samples while absorbing correlated ones. By thoroughly exploring the unlabeled data, it maintains comprehensive old knowledge while efficiently discovering novel classes.

Eq. 7 mimics how the model discovers novel classes on the incoming unlabeled data at test-time. Ideally, we like the adapted \( \hat{\theta}^E, P \) to perform well on all encountered classes. The test data from previous sessions and the current session separately reflect the catastrophic forgetting robustness and novel class discovery capability. Thus, we append \( \mathcal{D}^t_{te} \) to \( \mathcal{P} \). Accordingly, the meta-objective is defined as follows for the outer loop of the meta-level optimization:

\[
\min_{\hat{\theta}^E, \hat{\theta}^P} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{P}} \mathcal{L}_{sel}(\mathbf{x}, \mathbf{y}; \hat{\theta}^E, \hat{\theta}^P)
\]

(8)

where \( \mathcal{L}_{sel}(\cdot) \) is the supervised contrastive loss in Eq. 1. Note that the optimization is performed on \( \hat{\theta} \), although \( \mathcal{L}_{sel}(\cdot) \) is a function of \( \hat{\theta}^E \) and \( \hat{\theta}^P \). The meta-objective in Eq. 8 is then optimized using gradient descent, as shown in Line 13 of Alg. 1. We empty \( \mathcal{P} \) when all \( T + 1 \) sessions are iterated. After meta-training, we obtain an initialization of the model \( \theta \) which has been specifically trained to discover and learn novel objects from a sequence of unlabeled data.

Meta-testing. It is worth mentioning that the procedure in Alg. 1 aligns with the evaluation protocol. After discovering novel classes at each incremental session, the model is evaluated on all encountered classes. Our meta-objective optimizes the model towards what it is supposed to do at evaluation to maximize the performance. In addition, despite some uncertainties that may occur for unsupervised learning, the model is constrained by a fully supervised meta-objective. Thus, when training converges, the meta-model \( \hat{\theta} \) is ready to discover novel classes while maintaining the old knowledge by only running Line 10 of Alg. 1.

### 4. Experiments

#### 4.1. Dataset and setup

**Dataset.** We construct the C-GCD benchmark using three widely used datasets as in NCD [40, 35, 47], i.e., CIFAR10 [23], CIFAR100 [23] and Tiny-ImageNet [25]. Each dataset is split into two subsets, 1) large-scale labeled samples accounting for 80% of the known classes data constitute a labeled set for offline training; and 2) the remaining data containing known and novel classes are used as an unlabeled set for continual object discovery. In Tab. 1, we summarize the dataset splits used in our training.

**Session-wise data split.** All labeled samples \( S^0 \) are used for offline training in our setting. During the online incremental learning stage, the unlabeled samples are dynamically added (i.e., sessions \( t \geq 1 \)). Specifically, CIFAR10 is divided into 3 incremental sessions. In the \( t^{th} (t > 0) \) session, 3000 unlabeled images from 1 novel class and 2000 unlabeled images from \( 7 + (t - 1) \times 1 \) known classes are added. CIFAR100 is divided into 4 sessions, in which 1500 unlabeled images from 5 novel classes and 2000 unlabeled images from \( 80 + (t - 1) \times 5 \) known classes are added in the \( t^{th} \) session. The Tiny-ImageNet consists of 5 incremental sessions, each containing 3000 unlabeled images from 10 novel classes and 3000 unlabeled images from \( 150 + (t - 1) \times 10 \) known classes.

**Sequential task sampling.** During offline training, we use \( S^0 \) to sample sequential tasks. We first split \( S^0 \) into non-overlapping pseudo labeled and novel classes (4/3 for CIFAR10, 60/20 for CIFAR100 and 100/50 for Tiny-ImageNet). For each task, the pseudo labeled set is first used to warm up the model, followed by \( T \) incremental sessions of unlabeled samples containing the pseudo labeled and novel classes. Both the session number and the number of novel classes in each offline incremental session are consistent with the online incremental learning scenario.

**Evaluation metrics.** After learning the model on unlabeled samples at every online incremental stage, we follow [40] to measure the clustering accuracy between the ground truth labels \( y_i \) and the model’s predictions \( \hat{y}_i \) as:

\[
ACC = \max_{p \in \mathcal{P}(\mathbf{y})} \frac{1}{N} \sum_{i} (\mathbb{1}(y_i = p(\hat{y}_i)))
\]

(9)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Labeled Set</th>
<th>Unlabeled Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#class #image</td>
<td>#class #image</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>7 28000</td>
<td>10 22000</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>80 32000</td>
<td>100 18000</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>150 60000</td>
<td>200 40000</td>
</tr>
</tbody>
</table>

Table 1: Datasets used in our experiments. We show the number of classes in the labeled and unlabeled sets, as well as the number of samples.
where \( N \) is the total number of test samples and \( \mathcal{P}(Y) \) is the set of all permutations of the class labels \( Y \) encountered so far. The optimal permutation can be obtained via the Hungarian algorithm [24]. Our main metric is \( ACC \) on ‘All’ classes, indicating the accuracy across all accumulated test sets so far. To decouple the evaluation on forgetting and discovery, we further report accuracy for both the ‘Old’ classes subset (samples in the test set belonging to previous known classes) and ‘New’ classes subset (samples in the test set belonging to novel classes).

### Implementation details.

Following [40], we employ a vision transformer (ViT-B-16) [10] pretrained on ImageNet [9] with DINO [3] as the feature extractor throughout the paper. We use the Adam optimizer and the learning rates in Alg. 1 are set as \( \gamma = 0.1, \alpha = 0.001 \) and \( \beta = 0.0001 \). We use a batch size of 256 and \( \lambda = 0.35 \) to balance the losses in Eq. 3. Unless otherwise stated, we select the threshold \( \epsilon \) to be 0.85 in Eq. 5. At the meta-training stage, we first perform training on pseudo labeled set for 50 epochs, followed by 10 inner and 1 outer gradient updates for incremental sessions. At the meta-test stage, we directly perform 20 gradient updates to adapt using unlabeled samples. Furthermore, we follow standard practice in self-supervised learning to use the same projection head as in [3] and discard it at test-time.

### 4.2. Comparison with the state-of-the-art

Since this paper considers a new problem setting, there is no prior work that we can directly compare. Nevertheless, we choose SOTA methods on NCD and run their codes under our C-GCD setting, including RankStats [16], VanillaGCD [40], and recent continual NCD models FROST [35], GM [45]. Both RankStats and FROST train two classifiers on top of a shared feature representation. The first head is fed images from the labeled set and is trained with the cross-entropy loss, while the second head sees only images from unlabeled images of novel classes. In order to adapt RankStats and FROST to C-GCD, we train them with a single classification head for the total number of classes in the dataset. The sequential version of VanillaGCD is adopted and serves as the baseline for our model. We leverage the original training mechanism for GM.

In Tab. 2, we report the \textit{All/Old/New} class accuracy per incremental session for all methods, and the relative improvement for the final session. As we can see, the proposed method consistently outperforms all other methods on all three datasets among the most incremental sessions. Specifically, our MetaGCD surpasses the most recent method GM by 5.78\%, 4.81\% and 7.56\% on CIFAR10, CIFAR100 and Tiny-ImageNet datasets for the final \textit{New} classes accuracy.
Table 3: Ablation study of various components of our MetaGCD on the CIFAR100 dataset. We report ‘All’, ‘Old’ and ‘New’ class accuracy for each incremental session, and the average of all sessions such as mean ‘All’ (mA), mean ‘Old’ (mO) and mean ‘New’ accuracy (mN). Here CN denotes candidate neighbors and SP denotes soft positiveness.

Figure 4: Hyper-parameter analysis on the CIFAR100 dataset regarding feature similarity threshold (left) and various numbers of novel classes (right). An appropriate threshold or the number of new classes helps to stabilize the training process and improve performance.

4.3. Ablation Study

We conduct ablation studies on the CIFAR100 dataset to evaluate each component in our proposed framework. **Importance of neighborhood.** In our contrastive learning framework, we compute the soft correlation to allow more positive pairs to contribute to the loss. As reported in the second row of Tab. 3, considering the neighborhood achieves a performance gain of 1.25% (i.e., 74.92% v.s. 73.67% on the mean accuracy of All classes). The performance gain may come from the abundant comparisons from positive samples, which facilitates the current instance to align with more highly-correlated samples. We also conduct experiments to assess the sensitivity of threshold $\epsilon$ when selecting the positive neighbor instances. Increasing $\epsilon$ allows more strict positive pairs, but some partially related samples might be ignored. On the other hand, reducing $\epsilon$ increases the likelihood of introducing true negative samples, which may negatively impact model performance. As empirically found in the left side of Fig. 4, a trade-off should be made, and a threshold of 0.85 achieves the best performance.

**Importance of soft positiveness.** We then analyze the importance of soft positiveness to the recognition performance in the third row of Tab. 3. When we compute a correlation weight for each selected neighbor, the clustering accuracy on Novel classes increases from 62.89% to 64.55%. It indicates that the binary labeling strategy is insufficient to measure the correlation at the feature space, thus causing the backbone network to produce less discriminative representations compared with soft labeling methods. In the lower part of Fig. 2 (a), we show the correlation weight of selected neighbors with an input instance. The high score corresponds to the same category while the low score corresponds to less-correlated categories, which shows that our attention module is effective in modeling correlations between the input instance and each candidate neighbor.

**Effectiveness of meta-learning.** Our meta-learning optimization strategy further improves All classes accuracy to 74.56% for the final incremental session in the last row...
of Tab. 3. It demonstrates the effectiveness of the proposed method where the meta-objective specifically forces the model to discover novel classes without forgetting old classes. Additionally, we analyze the impact of the number of novel classes that are sampled during meta-training. To investigate this, we train separate models by setting the number of novel classes in the range of \{1, 10\}. As illustrated on the right side of Fig. 4, a larger number of classes for sequence tasks is more optimal. When there are fewer classes, the model is at a higher risk of overfitting to certain classes rather than learning how to incrementally learn.

5. Conclusion

In this paper, we propose a more realistic setting for real-world applications, namely C-GCD. The ultimate goal of C-GCD is to discover novel classes while keeping the old knowledge without forgetting. We propose a meta-learning based optimization strategy to directly optimize the network to learn how to incrementally discover with less forgetting. In addition, we introduce a soft neighborhood contrastive learning to utilize the soft positiveness to adaptively support the current instances from their neighbors. Extensive experiments on three datasets demonstrate the superiority of our method over state-of-the-art methods.

6. Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (No. 2022JBZY019), the National Key Research and Development Project (No. 2018AAA0100300) and an NSERC Discovery grant.

References


[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual rep-


