# Hyperbolic Hierarchical Representation Learning for Generalized Category Discovery

Yu Duan, Feiping Nie, *Senior Member, IEEE*, Huimin Chen, Zhanxuan Hu, Rong Wang, and Xuelong Li, *Fellow, IEEE*

***Abstract*—This study addresses the problem of generalized category discovery (GCD), an advanced and challenging semi-supervised learning scenario that deals with unlabeled data from both known and novel categories. Although recent research has effectively engaged with this issue, these studies typically map features into Euclidean space, which fails to maintain the latent semantic hierarchy of the training samples effectively. This limitation restricts the exploration of more detailed and rich information and degrades the performance in discovering new categories. The emerging field of hyperbolic representation learning suggests that hyperbolic geometry could be advantageous for extracting semantic information to tackle this problem. Motivated by this, we proposed hyperbolic hierarchical representation learning for GCD (HypGCD). Specifically, HypGCD enhances representations in hyperbolic space, building upon the Euclidean space representation from two perspectives: instance-class level and instance-instance level. At the instance-class level, HypGCD endeavors to construct well-defined clusters, with each sample forming a robust hierarchical cluster structure. Concurrently, at the instance-instance level, HypGCD anticipates that a subset of samples will display a tree-like structure in local space, which aligns more closely with real-world scenarios. Finally, HypGCD optimizes the Euclidean and hyperbolic space collectively to obtain refined features. Additionally, we show that HypGCD is exceptionally effective, achieving state-of-the-art (SOTA) results on several datasets. The code is available at `https://github.com/DuannYu/HypGCD`**

***Index Terms*—Generalized category discovery (GCD), hierarchical representation learning, hyperbolic space, open-world semi-supervised learning.**

## I. Introduction

Machine learning and deep learning have significantly surpassed human performance in cognitive models for tasks such as image classification, according to extensive labeled data. Nonetheless, labeling massive amounts of data poses significant challenges, only a small portion of them are labeled in practical scenarios. As a result, researchers have shifted their focus to tasks that involve incomplete labeling. These include semi-supervised learning [1], [2], self-supervised learning [3], noisy label learning [4], [5], and partial label learning [6], [7], few-shot [8], [9] and zero-shot learning [10] and among others.

Recent research has introduced the concept of novel category discovery (NCD) [11], [12], [13], a concept inspired by the human propensity to build upon existing knowledge when learning new information. NCD primarily aims to discover new categories by leveraging the knowledge derived from a set of labeled ones. However, the implementation of NCD assumes that all unlabeled data solely consists of novel categories, which is not feasible in real-world applications. To address these constraints, Vaze et al. [14] proposed generalized category discovery (GCD), which considers unlabeled data from both new and previously labeled categories. In addition, NCD can be regarded as a sub-task of out-of-distribution (OOD), since the datasets are all novel classes in distribution [15], [16].

This article primarily focuses on GCD. Broadly speaking, GCD can be categorized into two main types: *one-stage methods based on parametric classifiers* and *two-stage methods based on nonparametric classifiers*. The former methods generally involve constructing a classifier on the backbone and optimizing it jointly using both labeled and unlabeled data. Conversely, the latter methods initially learn cluster-favorable embeddings through the application of self-supervised learning, followed by the use of a nonparametric classifier, such as semi-supervised k-means, to establish the final cluster assignments [14], [17].

Regardless of the method type, it is evident that representation learning plays a vital role in GCD [18], [19]. Previous studies utilized both supervised and unsupervised contrastive learning techniques to acquire cluster-friendly representations, depending on whether the samples were labeled or unlabeled [17], [20]. However, contrastive learning primarily focuses on drawing positive pairs closer while distancing them from

Received 27 December 2023; revised 6 May 2024, 23 October 2024, 8 January 2025, and 2 June 2025; accepted 20 July 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101902; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2021JM-071; and in part by the National Natural Science Foundation of China under Grant 62176203, Grant 62176212, Grant 61936014, and Grant 61772427. *(Corresponding author: Zhanxuan Hu.)*

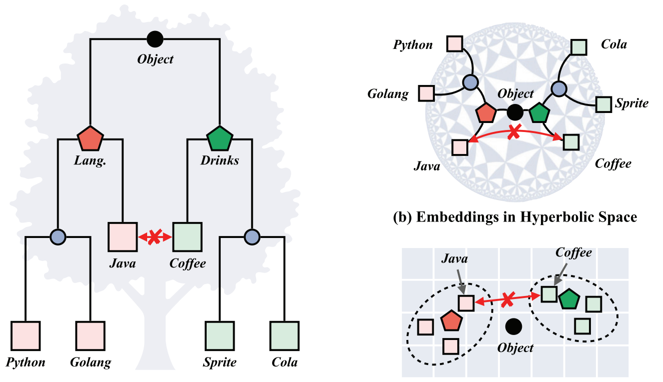
Yu Duan is with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: duanyuee@gmail.com).

Feiping Nie, Huimin Chen, and Rong Wang are with the School of Computer Science, the School of Artificial Intelligence, Optics and Electronics (iOPEN), and the Key Laboratory of Intelligent Interaction and Applications, Ministry of Industry and Information Technology, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China (e-mail: feipingnie@gmail.com; chenhuimin@mail.nwpu.edu.cn; wangrong07@tsinghua.org.cn).

Zhanxuan Hu is with the School of Information Science and Technology, Yunnan Normal University, Kunming 650500, China (e-mail: zhanxuanhu@gmail.com).

Xuelong Li is with the Institute of Artificial Intelligence (TeleAI), China Telecom, Beijing 100033, China (e-mail: xuelongli@ieee.org).

Digital Object Identifier 10.1109/TNNLS.2025.3597074



(a) Hierarchical Structure of Samples

(b) Embeddings in Hyperbolic Space

(c) Embeddings in Euclidean Space

Fig. 1. Motivation of our proposed method. The *squares* denote data samples and the *pentagons* denote the predefined class centers. Even though *java* and *coffee* are neighboring leaf nodes, their concepts are completely different. (a) Directly mapped to Euclidean space, the hierarchical structure will be lost. (b) *Tree-like* structure is mapped into hyperbolic space, and it can effectively reflect the hierarchical information between samples. (c) *Java* and *coffee* are very close in this space.

others [21], often neglecting the inherent structural information within clusters or classes. Therefore, it is intuitive that samples belonging to the same category should be close to one another. These categories should be conceptualized as sub-classes and predefined classes, as depicted in Fig. 1. Specifically, Fig. 1(a) demonstrates that *Python*, *Golang*, and *Java* are part of the predefined class of programming languages (*Lang.*). However, unlike *Java*, the first two are interpreted languages, whereas *Java* is a compiled language. Similarly, within the *Drinks* category, we discern that *Sprite* and *Cola* are carbonated beverages, while *Coffee* typically lacks carbonic acid. Moreover, both programming languages and drinks fall within the broader category of *Object*. Therefore, an effective representation should not only illustrate clear intra-class cluster structures for predefined classes but also display low intraclass variance and high interclass variance across sub-classes, thereby forming a desirable *tree-like* structure. This hierarchical relationship aids in establishing more nuanced semantic connections between samples, thereby augmenting the network's ability to learn superior representations and discover novel classes. Building on this concept, Pu et al. [17] introduced DCCL, a method that employs a nonparametric clustering approach to dynamically cluster data, maximizing the similarity between samples and their respective cluster centers.

To obtain a better representations, the aforementioned methods usually project the embeddings onto a $l_2$-normalized hypersphere, yielding distances with cosine similarity. However, previous studies [22], [23] have demonstrated that Euclidean spaces are unsuitable for *tree-like* data. As illustrated in Fig. 1(a), even the two neighboring leaf nodes such as *Java* and *Coffee*, exhibit certain similarities, and their relationship is quite distant in reality (*Java*'s icon is a coffee cup). Fig. 1(c) shows that directly mapping both *Java* and *Coffee* to Euclidean space would likely result in close distances between them, thereby producing a suboptimal representation. However, as shown in Fig. 1(b), mapping the samples to hyperbolic space can effectively mitigate this uncertainty. Because the negative curvature characteristic of hyperbolic space causes distances within the space to grow exponentially, improving the preservation of the tree structure without distortion.

Motivated by the benefits of hyperbolic space, we present hyperbolic hierarchical learning for GCD (HypGCD). HypGCD is adept at learning representations in hyperbolic space, enabling more fine-grained differentiation among sample categories, encompassing sub-classes and predefined classes. This fine-grained capability enhances our ability to discover new categories. Broadly, HypGCD fulfills folds: instance-class level and instance-instance level. At the instance-class level, we underscore the differences among various classes. In this context, we utilize pseudo-labels to establish cluster centers in hyperbolic space for samples from the same class, maximizing the similarity between samples and their corresponding centers. At the instance-instance level, our goal is to discern local hierarchical structures within unsupervised samples. More specifically, given a sample triplet consisting of a positive sample pair and a negative one, we form local *tree-like* structures by leveraging hyperbolic distances among all samples. This ensures that positive sample pairs are proximate to each other, while concurrently distanced from the negative one. Empirically, the HypGCD method effectively harnesses the hierarchical semantic information procured in hyperbolic space, improving the performance. The main contributions of HypGCD are summarized as follows.

1) To the best of our knowledge, HypGCD is the first attempt to integrate hierarchical information from hyperbolic space into the GCD, bridging the gap between hyperbolic space and Euclidean space.
2) By leveraging the hyperbolic space's suitability for capturing hierarchical structures, HypGCD improves the model's capability to identify and explore new categories, preserving both local and global hierarchical structures without any distortion.
3) We introduce a novel GCD framework, termed as HypGCD, which seamlessly integrates the advantages of both one-stage and two-stage GCD methods, without necessitating additional modules.
4) Extensive experiments demonstrate significant improvements over state-of-the-art (SOTA) GCD algorithms in both generic and fine-grained tasks.

## II. Related Works

### A. Semi-Supervised Clustering

The aim of semi-supervised clustering (SSC) is to partition samples into distinct groups by utilizing a limited amount of labeled data and massive unlabeled samples. Consistency-based method, as the most prevalent type of SSC, has been extensively applied across a variety of domains. Broadly speaking, these methods primarily strive to achieve consistent outputs from the model under different augmentations. For example, FixMatch [24], one of the most widely used methods, introduces a consistency regularization between strong and weak augmentations. ShrinkMatch [25] reduces the class space to enhance prediction certainty and subsequently employs diverse augmentations with consistency. In addition, several methods aim to enhance effectiveness from other perspectives.

For example, FreeMatch [26] dynamically adjusts class-specific confidence thresholds according to varying learning difficulties. CoMatch [27] and SimMatch [28] provide evidence that self-supervised representation brings advantages to SSC tasks.

Another perspective is on the connection relationship of samples, which use label propagation [29], alternating optimization [30], and other methods to make samples of the same type as close as possible.

While the aforementioned methods have made notable contributions, they all operate under the assumption that labeled data are available for every predefined category. Nevertheless, meeting this assumption proves challenging in numerous real-world scenarios. Consequently, this article addresses a more practical scenario of open-set SSC.

### B. Generalized Category Discovery

As mentioned above, GCD, as an open-world semi-supervised learning, is an extension of SSC and NCD. GCD allows the overlap between labeled and unlabeled categories. Based on the approaches used to acquire the final data clusters/labels, GCD can be broadly classified into two groups: *one-stage methods based on parametric classifier* and *two-stage methods based on nonparametric classifier*. Next, we will discuss them in detail.

*One-stage methods based on parametric classifier* aim to train a parametric classifier using both labeled and unlabeled data to directly assign samples to clusters. For example, PIM [31] maximizes mutual information from an information theory perspective to discover novel categories. SimGCD [32] is designed with parallel feature heads and classification heads integrated after the backbone. It employs a feature head to learn representations and uses a classification head to make predictions on the data.

*Two-stage methods based on nonparametric classifier* involve learning a good representation initially, followed by nonparametric classifier methods (such as semi-supervised k-means) to obtain the final clusters/labels. For instance, GCD [14] first utilizes supervised and self-supervised contrastive learning to fine-tune a pretrained model and employs semi-supervised k-means is applied on all the features to obtain the final cluster alignments. In contrast to GCD, DCCL [17] adopts an alternating approach that entails updating concepts and leveraging contrastive learning to acquire enhanced representations. Finally, it also employs semi-supervised k-means to obtain the final clustering results. It is important to note that while these types of methods can yield competitive results, they pose challenges in practical scenarios and large-scale datasets due to the high computational complexity of nonparametric postprocessing methods.

Recent works have expanded GCD in several directions. MetaGCD [33] introduced continual learning for GCD, where models face sequential unlabeled data containing both known and new categories. This requires continuous discovery of new categories while maintaining recognition of known ones. ImbaGCD [34] addressed data imbalance in GCD scenarios. It focuses on cases where known categories significantly outnumber unknown ones in the unlabeled data. IGCD [35] proposed an incremental learning approach. The model evolves through time steps, processing new labeled and unlabeled data while discarding old information. It must both classify known categories and discover new ones at each step. Finally, AGCD [36] explored active learning for GCD. The method strategically selects a small subset of unlabeled samples for labeling to enhance overall GCD performance.

### C. Hyperbolic Embedding

Hyperbolic embeddings have gained considerable attention and find wide application in NLP [37], [38], due to their effectiveness in capturing semantic information and hierarchical structures in text. Previous works have extended conventional linear layers to hyperbolic counterparts, redefined mathematical operations and recurrent neural networks, directly learning embeddings in hyperbolic space [39], [40]. Drawing from the success in NLP, many researchers have employed these effective tools in computer vision, leading to enhanced performance in few-shot learning [41] and representation learning [42], [43]. Unlike constructing complex hyperbolic networks, above methods proposed a hybrid structure that maps only the last layers into hyperbolic space, while all other operations are performed in the Euclidean space. For example, Yan et al. [44] proposed an unsupervised hyperbolic metric learning framework by employing hierarchical clustering. Ermolov et al. [23] employed pairwise cross-entropy loss with hyperbolic distances in conjunction with vision transformers. HyCoCLIP [45] uses the combined semantic information of images and texts to optimize the semantic alignment between images, image boxes, texts, and text boxes through hierarchical contrast learning. In the anomaly detection task, HypAD [46] uses the hyperbolic distance metric to optimize the model, breaking through the limitations of traditional Euclidean space on data expression capabilities.

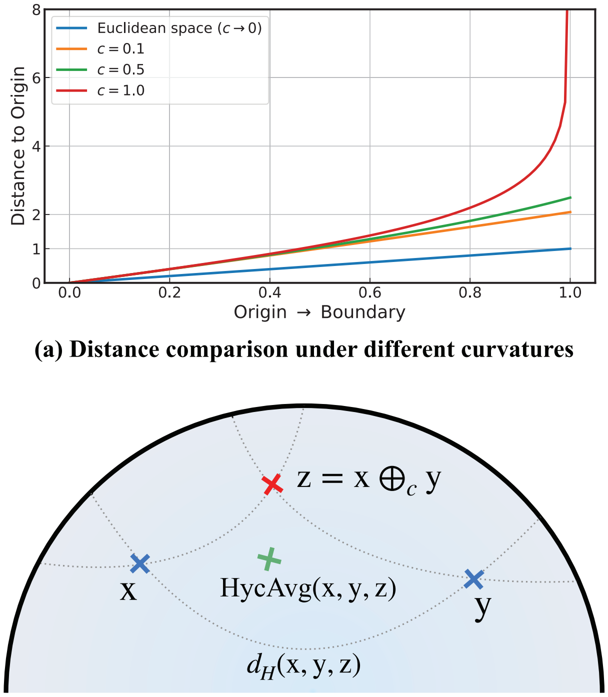
Building upon the aforementioned observations, we develop our HypGCD using this hybrid structure. However, in contract to the above mentioned methods, we establish connections between the Euclidean and hyperbolic spaces by leveraging pseudo labels and optimize them jointly. Furthermore, we explore the local relationships among samples in a hierarchical structure to acquire more detailed representations.

## III. Hyperbolic Hierarchical Representation Learning for GCD

### A. Preliminary: Poincaré Ball Model

In order to enhance comprehension of our proposed method, it is necessary to first introduce several definitions and operations in this section. Fig. 2 illustrates the characteristics and basic operations on Poincaré ball.

Formally, the $n$-dimensional hyperbolic space, denoted as $\mathbb{H}^n$, is mathematically defined as a homogeneous space characterized by a constant negative curvature. It is well known that hyperbolic space cannot be isometrically embedded in Euclidean space [47], [48]. However, there are several well-established models of hyperbolic geometry. Consistent with prior research [43], [49], we employ the Poincaré ball model in our study.



**(a) Distance comparison under different curvatures**

**(b) Visualization of basic operation on Poincaré ball**

Fig. 2. (a) Comparison of Euclidean distance and hyperbolic distance. (b) Visualization of the 2-D poincaré ball. Point $\mathbf{z}$ denotes the result of the *Mobius addition* of points $\mathbf{x}$ and $\mathbf{y}$. HycAvg stands for average operation in hyperbolic space. Gray lines represent geodesics, demonstrating the shortest length connecting two points.

The $n$-dimensional Poincaré ball model $(\mathbb{D}_c^n, g^{\mathbb{D}})$ is defined by the manifold $\mathbb{D}_c^n = \{\mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\| < 1\}$ and Riemannian metric $g^{\mathbb{D}} = \lambda_c^2 g^E$, where $c$ is a curvature hyperparameter, $\lambda_c = (2/1 - c\|\mathbf{x}\|^2)$ is the *conformal factor*, and $g^E = \mathbf{I}_n$ is Euclidean matrix tensor. The process of mapping embeddings from Euclidean space to the Poincaré ball is referred to as the *exponential map*. The common form of the *exponential map* $\exp_0(\mathbf{x})$ is as follows:

$$\exp_0(\mathbf{x}) = \tanh\left(\sqrt{c}\|\mathbf{x}\|\right)\frac{\mathbf{x}}{\sqrt{c}\|\mathbf{x}\|}. \tag{1}$$

As hyperbolic spaces do not possess the properties of vector spaces, a gyrovector formalism [50] is introduced to perform operations such as addition. For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{D}_c^n$, their addition operation in gyrovector spaces, called *Mobius addition*, is defined as

$$\mathbf{x} \oplus_c \mathbf{y} := \frac{\left(1 + 2c\langle\mathbf{x}, \mathbf{y}\rangle + c\|\mathbf{y}\|^2\right)\mathbf{x} + \left(1 - c\|\mathbf{x}\|^2\right)\mathbf{y}}{1 + 2c\langle\mathbf{x}, \mathbf{y}\rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \tag{2}$$

Based on the above definition, the geodesic distance between the vectors $\mathbf{x}$ and $\mathbf{y}$ in the Poincaré ball is defined as follows:

$$d_H(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \arctan\left(\sqrt{c}\,\|-\mathbf{x} \oplus_c \mathbf{y}\|\right). \tag{3}$$

It should be noted that when $c \to 0$, the $d_H(\mathbf{x}, \mathbf{y})$ becomes the Euclidean distance and we have

$$\lim_{c \to 0} d_H(\mathbf{x}, \mathbf{y}) = 2\,\|\mathbf{x} - \mathbf{y}\|. \tag{4}$$

Additionally, we need to introduce the average operation in hyperbolic space. As we all know, this operation in the Euclidean setting is represented as $(\mathbf{x}_1, \ldots, \mathbf{x}_N) \to (1/N)\sum_i \mathbf{x}_i$. The extension of this operation in hyperbolic space is referred to as the *Einstein midpoint*, which can be expressed in its simplest form in *Klein* coordinates

$$\texttt{HycAvg}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \frac{\sum_{i=1}^N \gamma_i \mathbf{x}_i}{\sum_{i=1}^N \gamma_i} \tag{5}$$

where $\gamma_i = 1/(1 - c\|\mathbf{x}\|^2)^{1/2}$ are the *Lorentz factors*, and $\mathbf{x}_i$ is sample in *Klein* coordinates. Here, to map the point from Poincaré ball to the *Klein* coordinates, we denote $\mathbf{x}_{\mathbb{P}}$ and $\mathbf{x}_{\mathbb{K}}$ as the coordinates of the same point in the Poincaré and *Klein* models correspondingly, and the following transition formulas hold:

$$\mathbf{x}_{\mathbb{K}} = \frac{2\mathbf{x}_{\mathbb{P}}}{1 + c\|\mathbf{x}_{\mathbb{P}}\|^2}, \quad \text{and } \mathbf{x}_{\mathbb{P}} = \frac{\mathbf{x}_{\mathbb{K}}}{1 + \sqrt{1 - c\|\mathbf{x}_{\mathbb{K}}\|^2}}. \tag{6}$$

### B. Problem Formulation

Given the dataset $\mathcal{D}$ defined as $\mathcal{D} = \mathcal{D}_L \cup \mathcal{D}_U$, where $\mathcal{D}_U = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}_U\}$ represents the unlabeled dataset and $\mathcal{D}_L = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}_L\}$ represents the labeled dataset with only a subset of categories labeled (i.e., $\mathcal{Y}_L \subset \mathcal{Y}_U$). The goal of GCD is to categorize the unlabeled images in $\mathcal{D}_U$ based on the prior supervised information in $\mathcal{D}_L$. Following similar approaches in existing studies [31], [51], we assume the number of labeled categories $|\mathcal{C}_l|$ and unlabeled categories $|\mathcal{C}_u|$ are known. The total number of categories is $|\mathcal{C}| = |\mathcal{C}_u|$.

### C. Proposed Method

*1) Overview:* In order to effectively uncover the latent hierarchical structure of representations, we have developed a novel GCD framework called HypGCD. As illustrated in Fig. 3, HypGCD integrates the information from separate Euclidean space into hyperbolic space, facilitating hierarchical representation learning of the data.

During the training stage, each sample in a mini-batch $\mathcal{B}$ is associated with two augmented views. Given an input image $\mathbf{x}_i \in \mathcal{B}$, we extract its representations from the backbone as $\mathbf{f}_i = f(\mathbf{x}_i) \in \mathcal{F}$. We define the label predictions and features as $\mathbf{p}_i = h(\mathbf{f}_i) \in \mathcal{P}$ and $\mathbf{e}_i = g(\mathbf{f}_i) \in \mathcal{E}$, respectively, where $h$ and $g$ represent the classifier and projector heads. Here, $\mathcal{F}$, $\mathcal{E}$, and $\mathcal{P}$ correspond to the Euclidean embedding, feature, and label spaces, respectively. Finally, we map the features $\mathbf{e}_i$ to the hyperbolic space to obtain $\mathbf{z}_i$, and combine the labels information from $\mathcal{P}$ to capture the latent hierarchical structure among samples. The overall objective loss function of HypGCD is

$$\mathcal{L} = \mathcal{L}_{Euc} + \mathcal{L}_{hyp}. \tag{7}$$

Here, $\mathcal{L}_{euc}$ and $\mathcal{L}_{hyp}$ represent the basic losses from Euclidean and hyperbolic spaces, respectively. Next, we will discuss strategies for obtaining improved representations in both spaces.

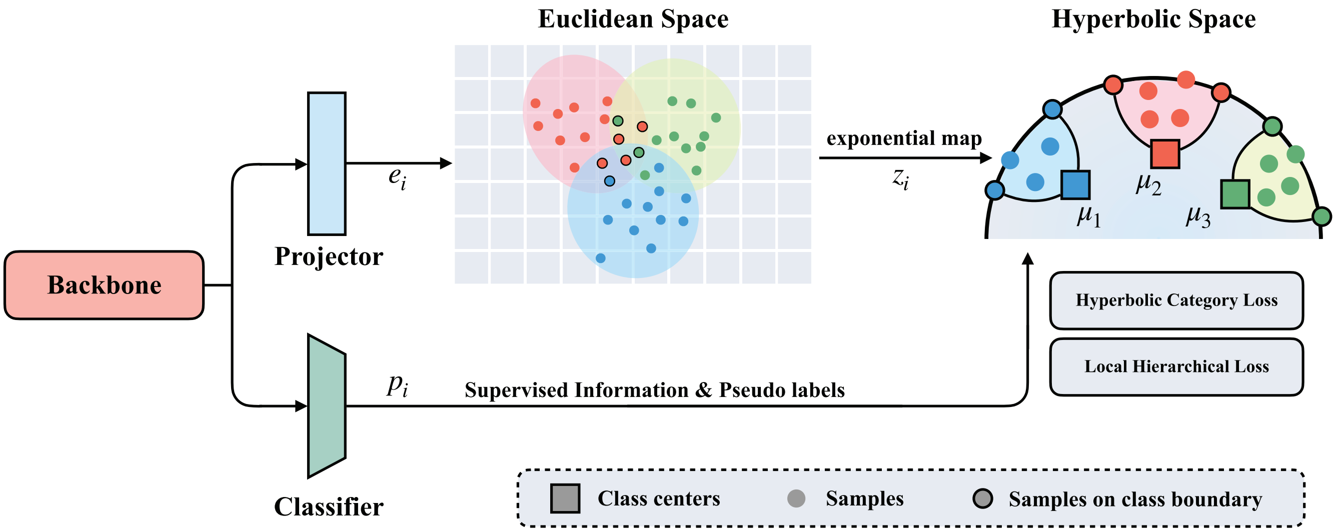


Fig. 3. Overview of the proposed method. It can be observed that in Euclidean space, the distribution of features at class boundaries shows intersections and errors. However, by mapping the features to hyperbolic space and learning a *tree-like* structure, the features belonging to different categories can be effectively pushed apart from one another.

*2) Euclidean Space Learning:* Inspired by [32], representation learning in Euclidean space can be divided into feature and label space learning. On one hand, contrastive learning has gained significant attention and widespread application in recent years, demonstrating promising performance [52], [53]. Accordingly, we employ self-supervised and supervised contrastive learning methods for feature space learning, depending on whether the data are labeled or not. On the other hand, for label space learning, we adopt a straightforward approach of employing cross-entropy for labeled data and leveraging self-distillation on all data to enhance feature stability. Next, we will provide a concise introduction to the process of learning representations in Euclidean space.

*1) Feature Space Learning:* Formally, given two random augmentations of input $\mathbf{x}$ and $\mathbf{x}'$, we jointly optimize a supervised contrastive loss on labeled samples

$$\mathcal{L}_{fea}^{s} = -\frac{1}{|\mathcal{B}_l|}\sum_{i\in\mathcal{B}^l}\frac{1}{|\mathcal{S}_i|}\sum_{q\in\mathcal{S}_i}\log\frac{\exp\left(\mathbf{e}_i^T\mathbf{e}_q'/\tau_s\right)}{\sum_{i\in\mathcal{B},i\neq j}\exp\left(\mathbf{e}_i^T\mathbf{e}_j'/\tau_s\right)} \tag{8}$$

and self-supervised contrastive loss on all samples, that is

$$\mathcal{L}_{fea}^{u} = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\log\frac{\exp\left(\mathbf{e}_i^T\mathbf{e}_i'/\tau_u\right)}{\sum_{i\in\mathcal{B},i\neq j}\exp\left(\mathbf{e}_i^T\mathbf{e}_j'/\tau_u\right)} \tag{9}$$

where $\tau_s$ and $\tau_u$ are temperature values, $\mathcal{S}_i$ denotes a sample set that shares the sample labels with $\mathbf{x}_i$, and $\mathcal{B}_l$ is a subset of $\mathcal{B}$ consisting of all labeled data in mini-batch.

*2) Label Space Learning:* Depending the samples have labels or not, we use cross entropy loss on labeled samples, that is

$$\mathcal{L}_{cls}^{s} = \frac{1}{2}\left(l\left(\mathbf{y}_i,\mathbf{p}_i\right)+l\left(\mathbf{y}_i,\mathbf{p}_i'\right)\right) \tag{10}$$

where $l(\mathbf{a},\mathbf{b}) = -\langle\mathbf{a},\log\mathbf{b}\rangle$, $\mathbf{y}_i$ is one-hot label of $\mathbf{x}_i$. Similarly, we use cross-entropy loss between predictions and pseudo-labels on unlabeled samples

$$\mathcal{L}_{cls}^{u} = l\left(\mathbf{p_i},\mathbf{p}_i'\right) - \epsilon H\left(\bar{\mathbf{p}}\right) \tag{11}$$

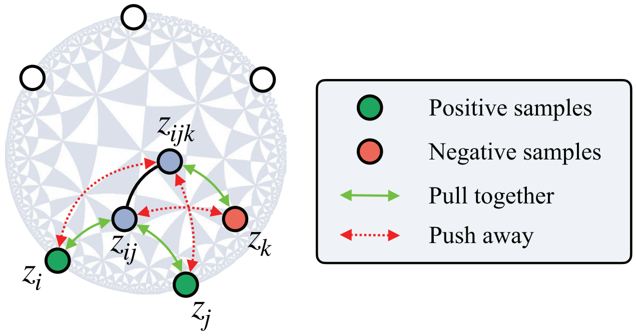


Fig. 4. Ideal local hierarchical structure of (21).

where $\epsilon$ is hyperparameter and $H(\cdot)$ is entropy regularization. To this end, the overall loss in Euclidean space is

$$\mathcal{L}_{Euc} = \lambda\left(\mathcal{L}_{fea}^{s}+\mathcal{L}_{cls}^{s}\right)+(1-\lambda)\left(\mathcal{L}_{fea}^{u}+\mathcal{L}_{cls}^{u}\right) \tag{12}$$

where $\lambda$ is a trade-off parameter.

*3) Hyperbolic Space Learning:* In this section, we implicitly combine the Euclidean feature and label space together, and map them to the Poincaré ball to learn more meaningful hierarchical representations. In sum, our representation learning can be divided into two folds, instance-class level and instance-instance level. For the former, our purpose is to encourage samples belonging to the same categories to be closer to each other. To achieve this goal, we introduce a hyperbolic category loss (HCL) that effectively pulls together samples with same labels in the hyperbolic space. For the latter, as shown in Fig. 4, even a group of samples also have local hierarchical structures. Thus, we propose local hierarchical loss. Next, we will provide a detailed discussion of the two components mentioned above.

*1) HCL:* As above-mentioned that the Euclidean feature space captures the distribution of all samples, while the label space indicates the cluster assignment for each sample.

Building upon the aforementioned generated representations, we integrate them together and propose our HCL. Specifically, we employ *exponential mapping* to obtain the hyperbolic representation $\mathbf{z}_i$ from $\mathbf{e}_i$ according to (1). Motivated by [54], [55], we then utilize the classification predictions from the label space to push samples of the same category closer while pulling samples from different categories apart. Formally, the hyperbolic category loss can be expressed as

$$\mathcal{L}_{hcl} = -\log \frac{\exp\left(-d_H\left(\mathbf{z}_i,\boldsymbol{\mu}_j\right)/\tau\right)}{\sum_{j=1}^{C}\exp\left(-d_H\left(\mathbf{z}_i,\boldsymbol{\mu}_j\right)/\tau\right)}. \tag{13}$$

Here, $\boldsymbol{\mu}_j$ represents the center of the $j$th class, and $\tau$ denotes the temperature value. It is important to note that our HCL is entirely based on the hyperbolic space, rather than the Euclidean space. Contrary to previous works [54], we employ (5) to compute all cluster centers as follows:

$$\boldsymbol{\mu}_j = \texttt{HycAvg}\left(\{\mathbf{z}_i\} \middle| \mathbf{z}_i \in \mathcal{C}_j\right) \tag{14}$$

where $\mathcal{C}_j$ is $j$th category. The advantages of HCL are twofold.

1) By integrating information from both labels and the feature space, we are able to jointly optimize features and enhance consistency in feature distributions across different spaces.
2) The properties of hyperbolic space allow for larger boundary margins between classes. In contrast to Euclidean space, even samples located on the class boundary that belongs to different classes, they still exhibit a large geodesic distance between each sample.

*2) Local Hierarchical Loss:* As mentioned earlier, samples exhibit not only predefined categories but also local hierarchical structures. As illustrated in Fig. 1, even if the samples belong to the category *Lang.*, they can be further subdivided into programming languages and compiled languages. To extract meaningful information and discover new categories, it is necessary to construct a local hierarchical structure of the samples.

Formally, given an anchor sample $\mathbf{x}_i$ and its positive pair $\mathbf{x}_j$ and negative pair $\mathbf{x}_k$, their representations in hyperbolic space can be obtained as $\mathbf{z}_i$, $\mathbf{z}_j$, and $\mathbf{z}_k$ using (1). Our objective is making $\mathbf{z}_i$ and $\mathbf{z}_j$ to be close to their common parent node while being far away from the negative one. The ideal local hierarchical structure is illustrated in Fig. 4.

Nevertheless, constructing a triplet $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$ solely based on pairwise distances is unreasonable [56]. To address this issue, we employ $K$-reciprocal nearest neighbors to generate a set of triplets. Specifically, let us denote the set of triplets as $\mathcal{T}$, which is sampled as

$$\mathcal{T} = \left\{\left(\mathbf{z}_i,\mathbf{z}_j,\mathbf{z}_k\right) \,\middle|\, \left(\mathbf{z}_j \in \mathcal{R}'_K\left(\mathbf{z}_i\right)\right) \cap \left(\mathbf{z}_k \notin \mathcal{R}'_K\left(\mathbf{z}_i\right)\right)\right\} \tag{15}$$

where $\mathcal{R}'_K$ considers the both $K$-reciprocal nearest and semi-supervised information. Denote $\mathcal{R}_K(\mathbf{z}) = \{\mathbf{z}' \big| (\mathbf{z}' \in \mathcal{N}_K(\mathbf{z})) \cap (\mathbf{z} \in \mathcal{N}_K(\mathbf{z}'))\}$ as the $K$-reciprocal nearest neighbors of $\mathbf{z}$ and $\mathcal{N}_K(\mathbf{z})$ is the $K$-nearest neighbors of $\mathbf{z}$, we could use following manner to obtain $\mathcal{R}'_K$:

$$\mathcal{R}'_K\left(\mathbf{z}\right) = \left\{\mathcal{R}_K\left(\mathbf{z}\right) \cup \mathbf{z}'\right\} \quad \forall \mathbf{z}' \in \mathcal{S} \tag{16}$$

where $\mathcal{S}$ denotes the sample set that shares the same label with $\mathbf{z}$. It should be note that all representations are in hyperbolic space, the distances between each other should obey the (3).

After obtaining triplets $\mathcal{T}$, in a mini-batch, the probability that samples $\mathbf{z}$ is nearest common father note of $\mathbf{z}_i$ and $\mathbf{z}_j$ is

$$\pi_{ij}\left(\mathbf{z}\right) = \exp\left(-\max\left\{d_H\left(\mathbf{z}_i,\mathbf{z}\right), d_H\left(\mathbf{z}_j,\mathbf{z}\right)\right\}\right). \tag{17}$$

Then, we introduce Gumbel-max trick to sample $\mathbf{z}_{ij}$ as common father note

$$\mathbf{z}_{ij} = \max_{\mathbf{z}}\left(\pi_{ij}\left(\mathbf{z}\right) + g_{ij}\right). \tag{18}$$

Here, $g_{ij}$ represents an i.i.d. sample drawn from the Gumbel-softmax distribution, preventing samples from falling into local sub-optima. Additionally, it is necessary to sample the common father node of the entire triplet. Similarly, the probability that samples $\mathbf{z}$ is nearest common father note of triple is given by

$$\pi_{ijk}\left(\mathbf{z}\right) = \exp\left(-\max\left\{d_H\left(\mathbf{z}_i,\mathbf{z}\right), d_H\left(\mathbf{z}_j,\mathbf{z}\right), d_H\left(\mathbf{z}_k,\mathbf{z}\right)\right\}\right) \tag{19}$$

and we sample $\mathbf{z}_{ijk}$[1] as common father note of entire triplet as follows:

$$\mathbf{z}_{ijk} = \max_{\mathbf{z}}\left(\pi_{ijk}\left(\mathbf{z}\right) + g_{ij}\right). \tag{20}$$

Inspired by Dasgupta cost [57], we utilize triplet loss in hyperbolic space to conduct local hierarchical structure

$$\begin{aligned}\mathcal{L}_{lhl} = &\left[d_H\left(\mathbf{z}_i,\mathbf{z}_{ij}\right) - d_H\left(\mathbf{z}_i,\mathbf{z}_{ijk}\right) + \delta\right]_+ \\ &+ \left[d_H\left(\mathbf{z}_j,\mathbf{z}_{ij}\right) - d_H\left(\mathbf{z}_j,\mathbf{z}_{ijk}\right) + \delta\right]_+ \\ &+ \left[d_H\left(\mathbf{z}_k,\mathbf{z}_{ijk}\right) - d_H\left(\mathbf{z}_k,\mathbf{z}_{ij}\right) + \delta\right]_+\end{aligned} \tag{21}$$

where $\delta$ is margin hyperparameters. To this end, the overall loss in hyperbolic space can be summarized as

$$\mathcal{L}_{hyp} = \alpha\mathcal{L}_{hcl} + \beta\mathcal{L}_{lhl} \tag{22}$$

where $\alpha$ and $\beta$ is trade-off hyperparameter and will be discussed in Section IV-E. Additionally, Algorithm 1 presents the PyTorch-like pseudo-code for our HypGCD.

## IV. Experiments

### A. Experimental Setup

*1) Datasets:* We demonstrate the effectiveness of our HypGCD on six widely used datasets: CIFAR-10, CIFAR-100 [58], and ImageNet-1K, which are generic image classification datasets, as well as three challenging fine-grained image classification datasets, including CUB [59], Stanford Cars [60], and FGVC-Aircraft [61]. Each dataset is divided into labeled and unlabeled subsets. Following the [14], we select 80% of the categories as labeled categories $\mathcal{Y}_l$ in CIFAR-100, and half of the categories as labeled categories for the other datasets. We construct the labeled set $\mathcal{D}_l$ by selecting half of the samples from these labeled class subsets. And the remaining samples constitute the unlabeled dataset $\mathcal{D}_u$. Detailed statistics and dataset separation are summarized in Table I.

[1] $\mathbf{z}_{ij}$ and $\mathbf{z}_{ijk}$ represent samples, and the subscripts $i, j, k$ emphasize their relationship with $\mathbf{z}_i, \mathbf{z}_j$, and $\mathbf{z}_k$.

**Algorithm 1** Pseudo Code for HypGCD

```
# f, g, h: backbone, projector and classifier
# x, Y_l: input images, known labels
# tau, lamb, alpha, beta: Hyperparameters

for epoch in range(max_epoch):
  # training step
  for x, uq_idxs in loader:
    # random augmentations
    x1 = aug(x); x2 = aug(x)

    e1 = g(f(x1)); e2 = g(f(x2)) # projector
    p = h(f(x1)); q = h(f(x2)) # classifier

    # 1. Euclidean space learning
    # 1.1 Feature space learning
    fea_s = supcon(e1, e2, Y_l)# supervised
    contrastive
    fea_u = un-supcon(e1, e2)# unsupervised
    contrastive

    fea_loss = lamb*fea_s + (1)-lamb)*fea_u

    # 1.2 Label space learning
    sup1 = CrossEntropy(p, Y_l)# supervised loss
    sup2 = CrossEntropy(q, Y_l)

    label_s = 0.5*(sup1 + sup2)
    label_u = CrossEntropy(p/tau, q)# un-supervised
    loss

    label_loss = lamb*label_s + (1)-lamb)*label_u

    # 1.3 Euclidean space loss
    euc_loss = fea_loss + label_loss

    # 2. Hyperbolic space learning
    z1, z2 = expmap(e1), expmap(e2) # exponential
    mapping

    # 2.1 Hyperbolic category loss
    c1, c2 = argmax(z1), argmax(z2) # pseudo labels
    mu1 = HycAvg(z1, c1) # compute centers from Eq.
    (14)
    mu2 = HycAvg(z2, c2)

    hcl1 = L_hcl(z1, mu1) # compute hcl loss
    hcl2 = L_hcl(z2, mu2)

    hcl_loss = 0.5*(hcl1 + hcl2)

    # 2.2 Local hierarchical loss
    lhl_loss = L_lhl(z1, z2)# From Eq. (13) to Eq.
    (20)

    # 3. Overall loss
    loss = euc_loss + alpha*hcl_loss +
    beta*lhl_loss

    # 4. Training model
    loss.backward()
```

*2) Comparison Methods:* We evaluate our method by comparing it against multiple baselines and SOTA methods. As discussed in Section II, the methods can be categorized into two groups. We leverage RS+ [62], UNO+ [11], ORCA [63], GCA [64], AMEND [65], and PIM [31] as one-stage methods, and also use four two-stage methods including k-means, GCD [14], DCCL [17] and GPC [66].

*3) Evaluation Protocol:* We follow the evaluation protocol presented in GCD [14], and compute accuracy using the Hungarian algorithm to compare the ground-truth labels with the model's cluster assignments. The algorithm is defined as follows:

TABLE I
STATISTICS AND SEPARATION OF DATASETS FOR GCD

| **Types** | **Dataset** | $\lvert\mathcal{Y}_l\rvert$ | $\lvert\mathcal{D}_l\rvert$ | $\lvert\mathcal{Y}_u\rvert$ | $\lvert\mathcal{D}_u\rvert$ |
|---|---|---|---|---|---|
| *Generic* | CIFAR-10 | 5 | 12,500 | 10 | 37,500 |
| | CIFAR-100 | 80 | 20,000 | 100 | 30,000 |
| | ImageNet-1K | 500 | 321,000 | 1000 | 960,000 |
| *Fine-grained* | CUB | 100 | 1,500 | 200 | 4,500 |
| | Stanford Cars | 98 | 2,000 | 196 | 6,100 |
| | FGVC-Aircraft | 50 | 1,700 | 100 | 5,000 |

$$ACC = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\hat{\mathbf{y}} = \mathtt{map}\left(\mathbf{y}\right)\right) \tag{23}$$

where $N$ represents the data scale, and $\mathtt{map}(\cdot)$ refers to the optimal Hungarian algorithm. Subsequently, we estimate the accuracy for all classes (All), known classes (Old), and novel classes (New).

*4) Implementation Details:* Consistent with previous studies [14], [17], [32], we utilize the ViT-B-16 model pretrained by DINO [67] as the backbone. The feature representation for each input image is obtained by utilizing the 768-D output of the [CLS] token. For all datasets, we only fine-tune the last transformer block in backbone. During the training stage, the model is presented with two views of each input image, each having random augmentations. Mini-batches are constructed by evaluating the output of the projector and selecting the nearest neighbors. Each mini-batch is composed of 125 images, consisting of 25 samples and their corresponding four nearest neighbors. The network is trained for 200 epochs on each dataset using a cosine decay schedule and an initial learning rate of 0.1.

To ensure a fair comparison, we set the trade-off factor $\lambda$ to 0.35. For Euclidean space learning, we assign the temperature parameters $\tau_u$ and $\tau_s$ as 1.0 and 0.07, respectively. In the case of hyperbolic space learning, we experimentally maintain a consistent set of hyperparameters: $\tau$ in $\mathcal{L}_{hcl}$ is set to be 0.1, the curvature is set to $c = 0.1$, the number of nearest neighbors in (15) is $K = 20$, and the margin in (21) $\delta$ is set to be 0.1. We also provide additional discussion regarding other hyperparameters, such as $\alpha$ and $\beta$, $K$, and $\delta$. All experiments are conducted using an NVIDIA GeForce RTX 3090 GPU.

## B. Qualitative Analysis

In this section, we use two visualized experiments to demonstrate the effectiveness of HypGCD intuitively.

*1) Visualization of Embeddings in Poincaré Ball:* In order to demonstrate that HypGCD effectively represents a latent semantic hierarchy of samples, we visualize the learned embedding vectors by projecting them onto a 2-D Poincaré ball. For visualization purposes, we employ UMAP [68] with the hyperbolic distance metric as a dimensional reduction technique.

At the beginning of training (Fig. 5 left), the samples exhibit a random distribution on the Poincaré ball, suggesting the presence of noise in distribution. The middle of Fig. 5

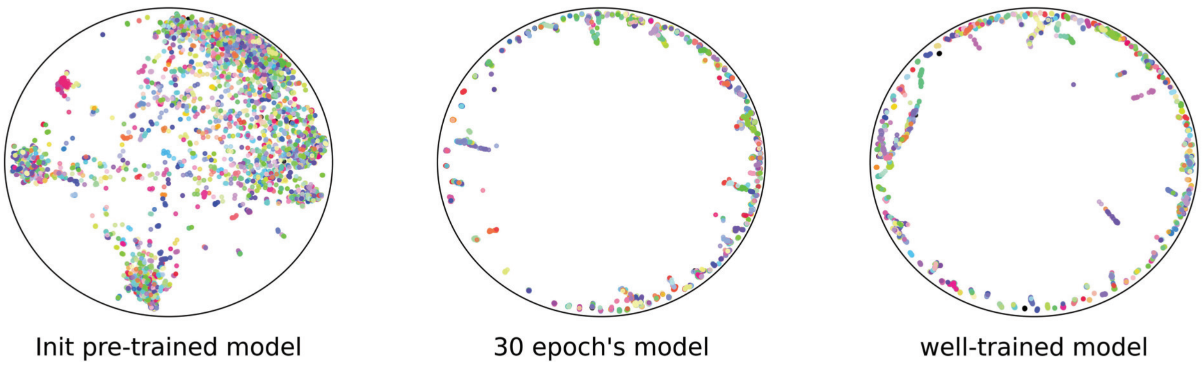


Fig. 5. Poincaré ball visualization of embeddings on Stanford cars. Different colors indicate different classes.

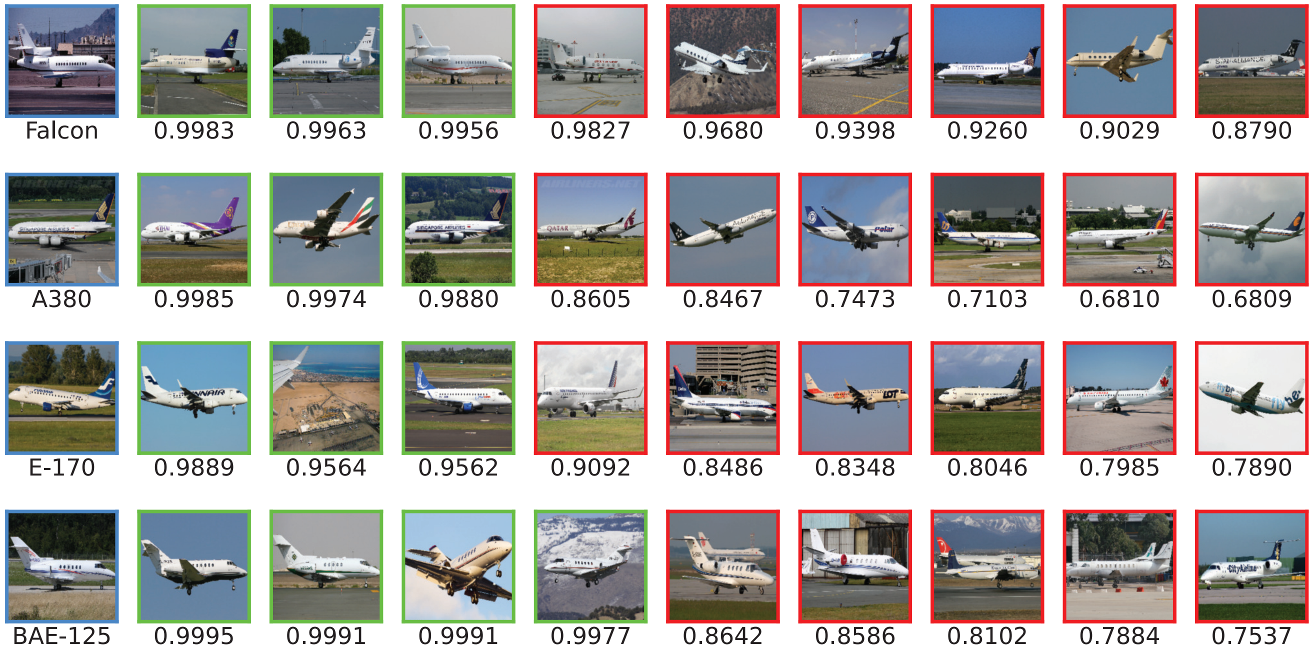


Fig. 6. Illustration of batch-wise nearest neighbors obtained from the well-trained model on FGVC-aircraft. Blue boundary represents the query image, while green and red boundaries indicate images that share the same and different labels corresponding the query, respectively. We simply utilize the cosine similarity to measure the relationship between two images.

presents the distribution of features at 30 epochs. It is apparent that the features progressively distribute toward the boundary of the Poincaré ball, indicating a structured pattern that enhances clustering and classification tasks. However, it is noticeable that the feature distribution is still not uniformity. One possible reason is that the model tends to predict features toward old classes, which degrades the quality of the feature distribution. A more detailed analysis of this issue will be conducted in subsequent sections. Conversely, following training, a well-trained model will result in the majority of samples converging near the boundary of the Poincaré ball. In contrast to Euclidean space, in the Poincaré boundary, even for two closely located points in Fig. 5, their geodesic distance is noticeably large. Furthermore, it is apparent that some samples are closer to the center of the Poincaré ball. Actually, these samples, along with those located on the boundary, form a local *tree-like* structures. Consequently, we can deduce that HypGCD efficiently captures intricate semantic hierarchical structures within the samples, improving representation learning and enhancing the model's performance to discover novel categories.

*2) Nearest-Neighbors Visualization:* Fig. 6 illustrates the batch-wise nearest neighbors obtained from the well-trained model on FGVC-aircraft. Evidently, samples belonging to the same class demonstrate high cosine similarities, predominantly surpassing 0.95. Conversely, for distinct classes, there is a substantial decline in cosine similarity. For example, the top three nearest neighbors of the A380 class exhibit a cosine similarity of 0.988, whereas the fourth nearest neighbor demonstrates a cosine similarity of only 0.8605. On the other hand, without prior knowledge, distinguishing the subtle differences between fine-grained aircraft poses a challenge for visual inspection. Our proposed model effectively discriminates the feature distribution, thereby improving its capability to identify new categories.

TABLE II

COMPARISON RESULTS (%) WITH SOTA METHODS. THE BEST RESULTS ARE BOLD. †DENOTES TWO-STAGE METHOD

| | CIFAR-10 | | | CIFAR-100 | | | CUB | | | Stanford Cars | | | FGVC-Aircraft | | | ImageNet 1K | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Methods** | **All** | **Old** | **New** | **All** | **Old** | **New** | **All** | **Old** | **New** | **All** | **Old** | **New** | **All** | **Old** | **New** | **All** | **Old** | **New** |
| K-means | 83.6 | 85.7 | 82.5 | 52.0 | 52.2 | 50.8 | 34.3 | 38.9 | 32.1 | 12.8 | 10.6 | 13.8 | 16.0 | 14.4 | 16.8 | - | - | - |
| GCD† [14] | 91.5 | 97.9 | 88.2 | 73.0 | 76.2 | 66.5 | 51.3 | 56.6 | 48.7 | 39.0 | 57.6 | 29.9 | 45.0 | 41.1 | 46.9 | 52.5 | 72.5 | 42.2 |
| DCCL† [17] | 96.3 | 96.5 | 96.9 | 75.3 | 76.8 | 70.2 | 63.5 | 60.8 | **64.9** | 43.1 | 55.7 | 36.2 | - | - | - | - | - | - |
| GPC† [66] | 90.6 | 97.6 | 87.0 | 75.4 | **84.6** | 60.1 | 55.4 | 58.2 | 53.1 | 42.8 | 59.2 | 32.8 | 46.3 | 42.5 | 47.9 | 49.3 | 70.2 | 41.4 |
| RS+ [62] | 46.8 | 19.2 | 60.5 | 58.2 | 77.6 | 19.3 | 33.3 | 51.6 | 24.2 | 28.3 | 61.8 | 12.1 | 26.9 | 36.4 | 22.2 | - | - | - |
| UNO+ [11] | 68.6 | **98.3** | 53.8 | 69.5 | 80.6 | 47.2 | 35.1 | 49.0 | 28.1 | 35.5 | 70.5 | 18.6 | 40.3 | 56.4 | 32.2 | 43.1 | 65.4 | 38.5 |
| ORCA [63] | 81.8 | 86.2 | 79.6 | 69.0 | 77.4 | 52.0 | 35.3 | 45.6 | 30.2 | 23.5 | 50.1 | 10.7 | 22.0 | 31.8 | 17.1 | - | - | - |
| GCA [64] | 92.8 | 94.4 | 91.9 | 76.6 | 79.5 | 70.7 | 62.3 | 72.0 | 57.5 | 45.4 | 65.5 | 35.6 | 47.1 | 57.1 | 42.2 | - | - | - |
| AMEND [65] | **96.8** | 94.6 | **97.8** | **81.0** | 79.9 | **83.3** | 64.9 | 75.6 | 59.6 | 56.4 | 73.3 | 48.2 | 52.8 | 61.8 | 48.3 | 54.3 | 74.2 | 43.7 |
| PIM [31] | 94.7 | 97.4 | 93.3 | 78.3 | 84.2 | 66.5 | 62.7 | 75.7 | 56.2 | 43.1 | 66.9 | 31.6 | - | - | - | 53.9 | 73.5 | 43.0 |
| HypGCD | 96.6 | 95.5 | 97.2 | 79.3 | 80.1 | 77.8 | **65.8** | 75.1 | 61.1 | **57.6** | **74.9** | **49.2** | **54.8** | **63.2** | **50.6** | **55.0** | **75.7** | **44.6** |

### *C. Comparison With SOTA*

In this section, we compare the proposed HycGCD with ten SOTA GCD algorithms. These include six one-stage methods (RS+ [62], UNO+ [11], ORCA [63], PIM [31], GCA [64], and AMEND [65]), and four two-stage methods (GCD [14], DCCL [17], and GPC [66]). Additionally, k-means is employed as the reference baseline method. Table II provides a summary of the experimental results on five benchmark datasets, with the best results highlighted in bold.

According to the results presented in Table II, we can draw the following conclusions.

1) From the perspective of methods types, we observe that one-stage methods demonstrate superior performance on the old class, whereas two-stage methods exhibit a greater propensity for discovering new classes. One of the main reasons is the one-stage methods contain parametric classifier, where label information directly impose on it through cross-entropy loss. Consequently, it makes the classifier to favorably predict the old classes, without considering their accuracy. In contrast, two-stage methods commonly employ self-supervised and supervised learning in the first stage to acquire cluster-favorable features. Subsequently, in the second step, nonparametric classifiers (e.g., semi-supervised k-means, DBSCAN) are employed to derive clustering outcomes. Notably, these nonparametric classifiers solely concentrate on feature distributions and do not possess any prior assumptions or bias to new or old classes. Consequently, they can assign a greater number of samples to new classes. Further discussion on this phenomenon will be provided in Section IV-F.
2) HypGCD clearly outperforms other comparative methods, particularly on fine-grained datasets, where it achieves the best performance. On generic datasets, such as CIFAR-10/100, HypGCD still shows the competitive performance. We believe that that these three datasets encompass a substantial amount of labeled data and a considerable proportion of labeled categories. In this context, conventional self-supervised learning suffices for acquiring meaningful representations in Euclidean space. For the remaining three fine-grained datasets, both data scale and small amount of labeled data significantly amplify the challenge of predicting new classes. However, HypGCD still captures intricate hierarchical relationships between samples and categories in hyperbolic space without distortion. It strengthens representation learning and the ability for discovering new categories. Moreover, HypGCD implicitly integrates information from both the feature space and the label space in Euclidean space, thereby ensuring the consistency of sample distributions in both domains.
3) As for large-scale dataset ImageNet-1k, HypGCD still achieves competitive results. For the performance on“New” categories, it is still 0.9% and 1.6% higher than AMEND and PIM, respectively. In general, GCD, DCCL, AMEND, PIM and HypGCD all use the same backbone structures and training strategy. Therefore, they obtain similar high-dimensional embeddings at the early training stage, leading the similar performance on different datasets. However, AMEND and PIM, which are more similar to HypGCD, perform slightly worse than our proposed method. The reason is that due to the large number of samples in ImageNet-1k, AMEND will inevitably introduce noise in the neighbor considerations, pulls neighbors that do not belong to the same class together, resulting in a decrease in “New” categories. On the other hand, in the task of clustering with too many categories, the loss of PIM is a little bit weak, and the performance also depends on the initialization (semi-supervised k-means++) of the cluster center.

### *D. Ablation Analysis*

In this section, we conduct ablation analysis on CUB, Stanford Cars, CIFAR-100, and ImageNet-1K, aiming to showcase the effectiveness of HypGCD. Table III provides validation for the crucial components of HypGCD and showcases their performance. Throughout all ablation experiments, the baseline model solely employs $\mathcal{L}_{euc}$ as its objective function.

*1) Effectiveness of Hyperbolic Categories Loss:* Following the integration of $\mathcal{L}_{hcl}$ into the baseline, a notable enhancement in the model’s performance on the old class is observed, with respective increases of 3.4% and 3.7% on the CUB and cars datasets. The improvement on the new class on these

TABLE III
Ablation Study on the Different Components of HypGCD (%)

| | | | CUB | | | Stanford Cars | | | CIFAR-100 | | | ImageNet-1K | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{L}_{hcl}$ | $\mathcal{L}_{lhl}$ | All | Old | New | All | Old | New | All | Old | New | All | Old | New |
| 1 | Baseline | | 60.3 | 65.6 | 57.7 | 53.8 | 71.9 | 45.0 | 76.9 | 82.9 | 64.9 | 51.5 | 73.7 | 40.6 |
| 2 | ✓ | | 61.8+1.5 | 69.0+3.4 | 58.2+0.5 | 55.9+2.1 | 75.6+3.7 | 46.4+1.4 | 77.9+1.0 | 80.1-2.8 | 71+6.1 | 52.9+1.4 | 74+0.3 | 42.6+2.0 |
| 3 | | ✓ | 61.7+1.4 | 66.9+1.3 | 59.1+1.4 | 56.5+2.7 | 74.0+2.1 | 48.1+3.1 | 78.9+2.0 | 82.7-0.2 | 71.3+6.4 | 51.6+0.1 | 73.9+0.2 | 40.4-0.2 |
| 4 | ✓ | ✓ | 63.5+3.2 | 71.7+6.1 | 59.4+1.7 | 57.6+3.8 | 74.9+3.0 | 49.2+4.2 | 79.3+2.4 | 80.1-2.8 | 77.8+12.9 | 55+3.5 | 75.7+2.0 | 44.6+4.0 |

two datasets are limited, exhibiting increases of merely 0.5% and 1.4%, respectively. However, there are 6.1% and 2.0% improvements on generic datasets (CIFAR-100 and ImageNet-1K). As mentioned earlier, the label information applied to the parametric classifier is more effective in improving the performance on the old class.

*2) Effectiveness of Local Hierarchical Loss:* Observing the baseline and the third row in Table III, it becomes evident that the incorporation of $\mathcal{L}_{hcl}$ yields notable performance enhancements in both the old and new categories. Specifically, we observe increases of 1.4%, 3.1%, and 6.4% on the new class in CUB, Stanford cars, and CIFAR-100, respectively. It directly illustrates the effectiveness of constructing local hierarchical structures. In other words, the local hierarchical structure better reflects the subtle semantic relationships among samples in fine-grained datasets. Therefore, it could better leads the model to learn more meaningful representations and improves the ability to discover new classes. However, the effect of $\mathcal{L}_{hcl}$ on ImageNet-1K is not very pronounced. One possible reason is that the large scale makes the local hierarchical structure less apparent.

*3) Effectiveness of Hyperbolic Representation Learning:* The last row of Table III demonstrates the effectiveness of hyperbolic space representation learning. HypGCD consistently outperforms the baseline, exhibiting significant improvements, especially in the case of Stanford cars, where it achieves increases of 3.0% and 4.2% on the old and new classes, respectively. On one hand, HypGCD integrates information from both the feature and label spaces in Euclidean space, enabling the acquisition of features that encompass richer semantic information at both the instance–instance and instance-class levels. On the other hand, due to the intrinsic properties of hyperbolic space, it can faithfully depict the hierarchical relationships among samples. Directly expressing these hierarchical relationships in traditional Euclidean space is infeasible, which shows the superiority of our proposed method by mapping features into hyperbolic space. As for CIFAR-100, we can observe a slight decrease in performance on the old class, but a significant improvement on the New class. We could tolerate this slight performance degrades, especially aiming to discover novel categories.

### *E. Impact of Hyperparameters*

In this part, we explore the impact of hyperparameters in HypGCD, including trade-off hyperparameters $\alpha$ and $\beta$ in (22), number of nearest-neighbor $K$ in (15) and margin $\delta$ in (21).

*1) Impact of Trade-Off Hyperparameters:* In this test, we vary the values of $\alpha$ and $\beta$ within the ranges $\mathcal{S}_\alpha$ = {0.1, 1.0, 10.0} and $\mathcal{S}_\beta$ = {1.0, 5.0, 10.0}, respectively. As shown in Fig. 7, too large value of $\alpha$ could degrade the performance. One possible reason is that larger $\alpha$ will introduces noise from pseudo-labels. Since these pseudo-labels heavily rely on the classifier's output and introduce wrong predictions, reducing the quality of instance-class level representations. As for $\beta$, A larger $\beta$ allows the model to achieve better results, indicating that $\mathcal{L}_{lhl}$ can effectively extract local hierarchical information. However, on CUB, we could see that a larger $\beta$ leads to a decrease on old classes but an improve on new classes in performance. This phenomena aligns with our expectations, since $\mathcal{L}_{lhl}$ can extract more fine-grained features, enhancing the ability to discover new classes. In other words, under reasonable hyperparameter settings, our proposed strategy of hyperbolic space hierarchical representation is effective in the problem of discovering new classes.

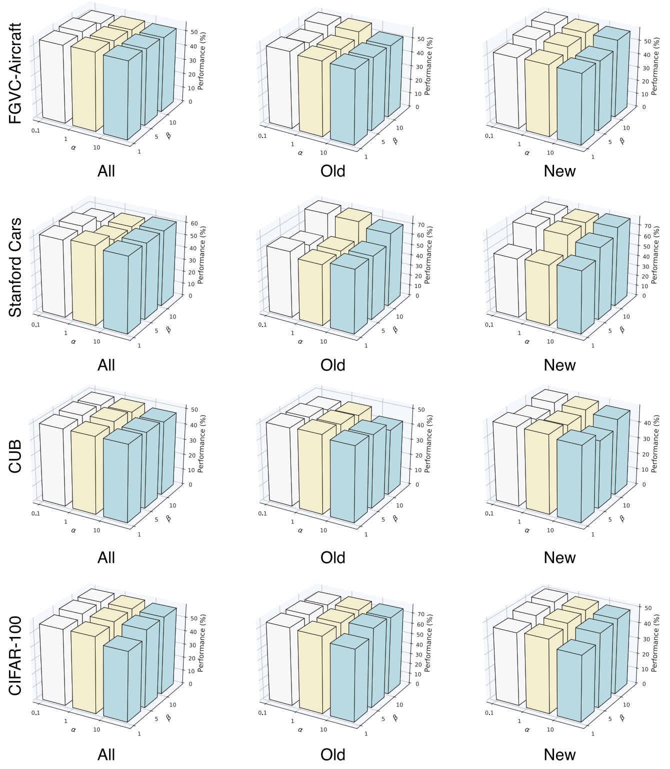


Fig. 7. Impact of trade-off hyperparameters on all, old, and new categories.

*2) Impact of nearest-neighbor $K$ in (15):* The selection of the common father node in the local hierarchical structure is directly determined by the number of nearest neighbors, denoted as $K$ in (15). As shown in Fig. 8, we define the range

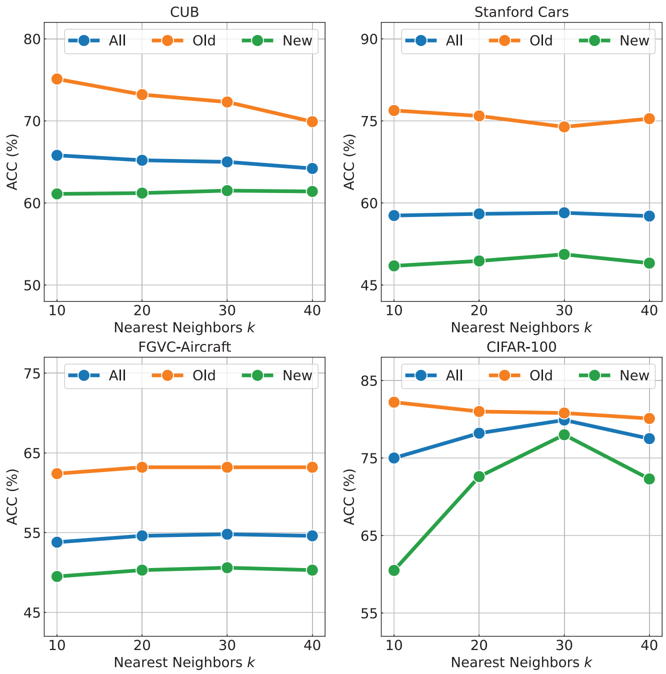


Fig. 8. Impact of nearest-neighbor $K$ on all categories.

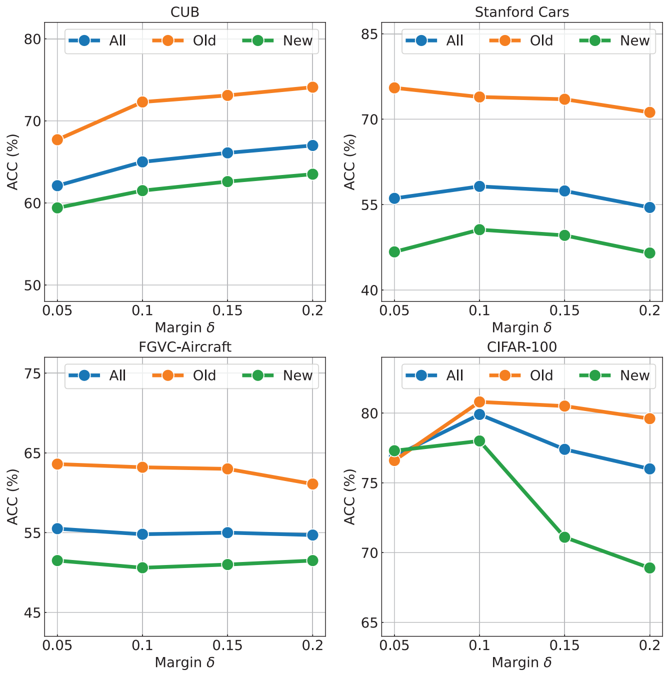


Fig. 9. Impact of margin $\delta$ on all categories.

of nearest neighbors as $\mathcal{S}_K = \{10, 20, 30, 40\}$ to observe its impact on performance. On one hand, a decrease in performance is observed when $K = 10$. One possible reason for this decrease is that a small number of nearest neighbors disregards the reciprocal relationships among samples, thereby distorting the construction of the local hierarchical structure. On the other hand, a larger number of K also leads to performance degradation. Since it includes too many redundant samples, resulting in the consideration of false-positive samples that do not belong to the same class.

*3) Impact of margin $\delta$ in (21):* The local hierarchical loss involves another margin parameter, denoted as $\delta$, which also requires consideration. As shown in Fig. 9, we define the range of margin as $\mathcal{S}_\delta = \{0.05, 0.10, 0.15, 0.20\}$, to observe its impact on performance. The value of it determines the *compactness* of the local structure tree. A larger value of $\delta$ indicates a more blurred local hierarchical structure, whereas a smaller value of $\delta$ results in a clearer one. Overall, it is observed that HypGCD achieves good results at $\delta = 0.10$ and $\delta = 0.15$. This observation is intuitive. When $\delta$ is too small, HypGCD excessively emphasizes the local hierarchical structure, diminishing the generalization representation and detrimentally impacting the method performance. Conversely, when $\delta$ is too large, the local structure constructed by HypGCD becomes blurred, decreasing the discriminate of fine-grained features and consequently hurting the model's performance. In summary, for all experiments, we set $\delta$ to 0.1 to achieve the optimal results.

### *F. In Depth Analysis*

*1) Analysis on Euclidean Space and Hyperbolic Space:* This section primarily focuses on analyzing the influence of the hyperbolic space itself on the performance. Specifically, while keeping other experimental settings unchanged, we set the curvature of the hyperbolic space to 0, implying that all computations are conducted in Euclidean space according to (4). Fig. 10 illustrates the performance comparison in two distinct spaces. It could be clearly observed that the performance based on hyperbolic space is consistently better than Euclidean space across all datasets. This phenomenon can be attributed to two primary factors as follows.

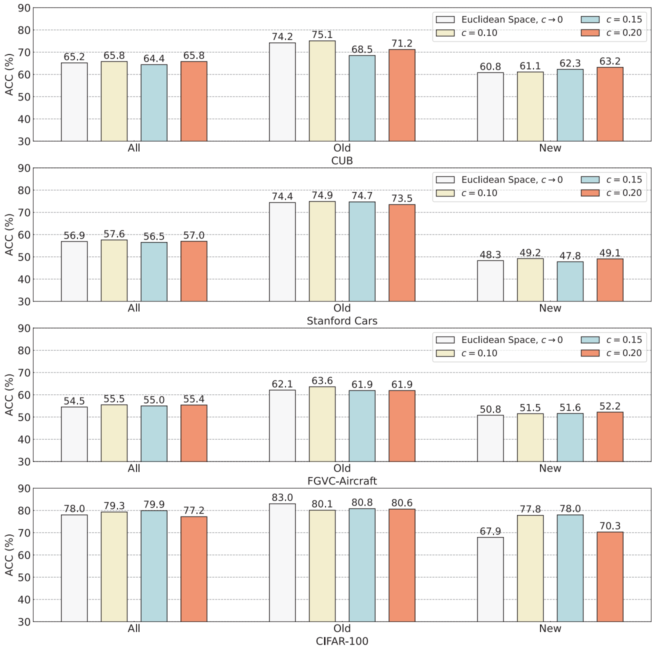


Fig. 10. Comparison result between Euclidean space and hyperbolic space.

1) At the instance-instance level, hyperbolic space can better captures hierarchical relationships among samples in triplets, learning more fine-grained features. In contrast,

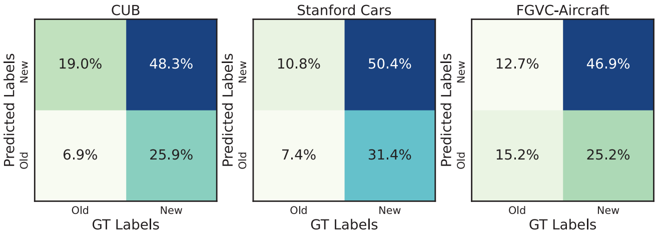


Fig. 11. Bias of prediction errors on three fine-grained datasets.

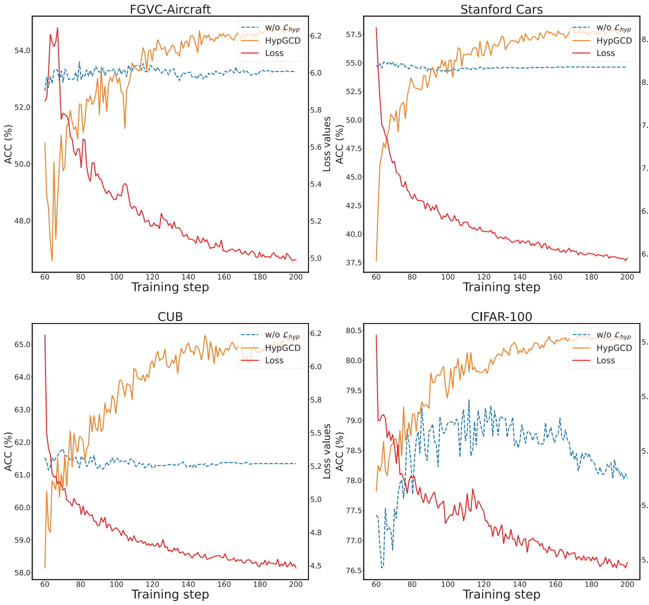


Fig. 12. Complete performance of HypGCD on four datasets.

Euclidean space introduces unnecessary semantic noise. This improvement is reasoned by the intrinsic negative curvature of hyperbolic space.

2) Based on the clear local tree structure, the computation of cluster centers in hyperbolic space achieves greater accuracy at the instance-class level. Due to the presence of negative curvature, it incorporates the hierarchical information of samples into the computation, instead of distanced-based averaging only.

To this end, hyperbolic space could discover the latent semantic hierarchy of training data, and deploy the hierarchy to provide richer and more fine-grained representations than Euclidean space.

*2) Analysis on Prediction Bias:* In this section, we explore deeper into the discussion of HypGCD's ability to discover new classes, specifically focusing on the bias in classifier predictions. To achieve these goal, we divide the prediction errors into four categories: Old–Old, Old–New, New–Old, and New–New. For example, Old–New indicates that the sample's true label is old, but it is incorrectly assigned to another new class. And New–New denotes that the samples from new categories are wrongly assigned to another new categories.

Fig. 11 presents a summary of the error type proportions across the CUB, Stanford cars, and FGVC-aircraft. We can clearly see that the classifier's primary errors stem from predicting new classes, constituting over 75% of the total errors in average. Since during the model's training process, there always absent the supervised information of new classes. Solely depending on self-supervised and self-distillation, learning limits the ability to discover new classes. Additionally, we observe that New–Old account for 25.9%, 31.4%, and 25.2% of the errors through the three datasets, respectively. This suggests that the classifier has a tendency to incorrectly predict unknown classes as known classes, consequently enhancing the performance on the old classes, no matter the sample truly belongs to a new or old class. We believe that during the training process, the label information shows a direct influence on the classifier via cross-entropy loss, leading the classifier more sensitive to the old classes. Conversely, there is insufficient *strong* supervision to guide the classifier's performance on new classes, restricting its capacity to discover new classes.

*3) Robustness of HypGCD:* Here, we have extended the GCD to noise labels scenarios. For simplicity, we refer to this task as Noisy GCD (NGCD) in this context. To ensure fairness in NGCD, we keep the settings for new class partitioning and hyperparameters consistent with HypGCD. The only difference is that we introduce some noise into the ground truth labels.

Table IV summarizes the performance of HypGCD on four datasets with different noise ratios. It can be observed that as the noise ratio increases, the performance significantly decreases, which aligns with expectations. Furthermore, we can see that on three fine-grained datasets, the performances greatly decrease for old classes compared to new one. We believe there are two reasons for this phenomena. First, noisy supervised information directly degrades the classifier and distorts accuracy. Second, even we will leverage noisy labels to calculate $\mathcal{L}_{hcl}$, $\mathcal{L}_{lhl}$ still constructs local *tree-like* structures. This indicates that our model exhibits robustness to noise. It is worth noting that the performance on CIFAR-100 is quite challenging. It demonstrates the strong noise resistance capability of HypGCD when dealing with generic image datasets. In sum, although HypGCD is designed for the GCD task, it can be extended to other tasks, which is one of our future works.

*4) Complete Performance of HypGCD:* Fig. 12 illustrates the performance of our model throughout the training process, along with the corresponding values of the objective functions. We can see that the model's performance stabilizes after 60 epochs when $\mathcal{L}_{hyp}$ is not incorporated. Notably, the introduction of $\mathcal{L}_{hyp}$ results in a short decreasing in the performance. Nonetheless, the overall performance of the HypGCD eventually achieving optimal performance. One possible reason is that $\mathcal{L}_{hyp}$ disrupts the stable feature distribution in the original Euclidean space. It considers the *tree-like* distribution of features in the hyperbolic space both on the instance–instance and instance-class levels, which could better capture the fine-grained semantic features. Finally, when the feature distribution stabilizes in hyperbolic space, HypGCD achieves the optimal results.

TABLE IV
RESULTS (%) ON FOUR DATASETS OF VARIOUS NOISY LABEL RATIOS

| | CUB | | | Stanford Cars | | | FGVC-Aircraft | | | CIFAR-100 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **# Noise** | **All** | **Old** | **New** | **All** | **Old** | **New** | **All** | **Old** | **New** | **All** | **Old** | **New** |
| 0% | 65.0 | 72.3 | 61.5 | 58.2 | 73.9 | 50.6 | 54.8 | 63.2 | 50.6 | 79.9 | 80.8 | 78.0 |
| 20% | 54.6 | 60.0 | 51.9 | 39.0 | 46.7 | 35.3 | 40.9 | 37.7 | 42.5 | 78.4 | 81.4 | 72.2 |
| 40% | 44.1 | 39.0 | 46.7 | 39.0 | 46.9 | 35.1 | 37.3 | 31.3 | 40.3 | 78.1 | 81.9 | 70.6 |
| 60% | 40.4 | 36.8 | 42.2 | 30.0 | 38.9 | 25.7 | 34.6 | 26.9 | 38.4 | 77.2 | 80.1 | 71.5 |

## V. CONCLUSION

This article introduces HypGCD, a novel representation learning method for GCD. HypGCD effectively explores the hierarchical *tree-like* relationships among samples in hyperbolic space at both the instance-class and instance–instance levels. Furthermore, we validate the effectiveness of HypGCD through extensive experiments, demonstrating its superior performance compared to SOTA methods on five widely used datasets. In the future, our aims are expanding HypGCD to address more challenging scenarios, including situations where the number of new categories are unknown and dealing with noisy labels in GCD, among others.

## VI. CONTRIBUTION

Yu Duan conceived the main idea of the paper, designed the methodology, conducted experiments, and wrote the article. Feiping Nie provided critical guidance on the main idea and methodology, supervised the research process, and reviewed the article. Huimin Chen conducted the additional experiments required during the review process, analyzed the results, and reviewed the article. Zhanxuan Hu assisted in the development of the methodology, conducted some experiments, and contributed to reviewed the article. Rong Wang supervised the research process and provided guidance on the overall structure of the article. Xuelong Li provided critical feedback and contributed to reviewing and refining the article.

## REFERENCES

[1] L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi, "Revisiting weak-to-strong consistency in semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7236–7246.

[2] M.-K. Xie, J.-H. Xiao, H.-Z. Liu, G. Niu, M. Sugiyama, and S.-J. Huang, "Class-distribution-aware pseudo labeling for semi-supervised multi-label learning," 2023, *arXiv:2305.02795*.

[3] S. Lin et al., "Prototypical graph contrastive learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2747–2758, Feb. 2024.

[4] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8135–8153, Nov. 2023.

[5] X. Xia et al., "Part-dependent label noise: Towards instance-dependent label noise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7597–7610.

[6] T. Xu, Y. Xu, S. Yang, B. Li, and W. Zhang, "Learning accurate label-specific features from partially multilabeled data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 10436–10450, Aug. 2024.

[7] D.-D. Wu, D.-B. Wang, and M.-L. Zhang, "Revisiting consistency regularization for deep partial label learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 24212–24225.

[8] W. Ju et al., "Few-shot molecular property prediction via hierarchically structured learning on relation graphs," *Neural Netw.*, vol. 163, pp. 122–131, Jun. 2023.

[9] Y. Zhao et al., "Personalized federated few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2534–2544, Feb. 2024.

[10] W. Ju et al., "Zero-shot node classification with graph contrastive embedding network," *Trans. Mach. Learn. Res.*, Jan. 2023.

[11] E. Fini, E. Sangineto, S. Lathuiliere, Z. Zhong, M. Nabi, and E. Ricci, "A unified objective for novel class discovery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9284–9292.

[12] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8401–8409.

[13] S. Roy, M. Liu, Z. Zhong, N. Sebe, and E. Ricci, "Class-incremental novel class discovery," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 317–333.

[14] S. Vaze, K. Hant, A. Vedaldi, and A. Zisserman, "Generalized category discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7482–7491.

[15] Y. Li et al., "Out-of-distribution detection with an adaptive likelihood ratio on informative hierarchical VAE," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 7383–7396.

[16] H. Zheng et al., "Out-of-distribution detection learning with unreliable out-of-distribution sources," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023.

[17] N. Pu, Z. Zhong, and N. Sebe, "Dynamic conceptional contrastive learning for generalized category discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7579–7588.

[18] B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang, and M. Long, "Debiased self-training for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 32424–32437.

[19] W. An, F. Tian, Q. Zheng, W. Ding, Q. Wang, and P. Chen, "Generalized category discovery with decoupled prototypical network," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 11, pp. 12527–12535.

[20] W. Ju et al., "Unsupervised graph-level representation learning with hierarchical contrasts," *Neural Netw.*, vol. 158, pp. 359–368, Jan. 2023.

[21] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang, "Nearest neighbor matching for deep clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13693–13702.

[22] R. Sarkar, "Low distortion Delaunay embedding of trees in hyperbolic plane," in *Proc. Int. Symp. Graph Drawing*. Cham, Switzerland: Springer, 2011, pp. 355–366.

[23] A. Ermolov, L. Mirvakhabova, V. Khrulkov, N. Sebe, and I. Oseledets, "Hyperbolic vision transformers: Combining improvements in metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7409–7419.

[24] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 596–608.

[25] L. Yang, Z. Zhao, L. Qi, Y. Qiao, Y. Shi, and H. Zhao, "Shrinking class space for enhanced certainty in semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 16187–16196.

[26] Y. Wang et al., "FreeMatch: Self-adaptive thresholding for semi-supervised learning," 2022, *arXiv:2205.07246*.

[27] J. Li, C. Xiong, and S. C. H. Hoi, "CoMatch: Semi-supervised learning with contrastive graph regularization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9455–9464.

[28] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14471–14481.

[29] T. Zheng et al., "Transition propagation graph neural networks for temporal networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4567–4579, Apr. 2024.

[30] Y. Duan, Z. Lu, R. Wang, X. Li, and F. Nie, "Toward balance deep semisupervised clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 2, pp. 2816–2828, Feb. 2025.

[31] F. Chiaroni, J. Dolz, Z. I. Masud, A. Mitiche, and I. B. Ayed, "Parametric information maximization for generalized category discovery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1729–1739.

[32] X. Wen, B. Zhao, and X. Qi, "Parametric classification for generalized category discovery: A baseline study," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 16590–16600.

[33] Y. Wu, Z. Chi, Y. Wang, and S. Feng, "MetaGCD: Learning to continually learn in generalized category discovery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1655–1665.

[34] Z. Li, B. Dai, F. Simsek, C. Meinel, and H. Yang, "ImbaGCD: Imbalanced generalized category discovery," 2023, *arXiv:2401.05353*.

[35] B. Zhao and O. M. Aodha, "Incremental generalized category discovery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 19080–19090.

[36] S. Ma, F. Zhu, Z. Zhong, X.-Y. Zhang, and C.-L. Liu, "Active generalized category discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16890–16900.

[37] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[38] M. Nickel and D. Kiela, "Learning continuous hierarchies in the Lorentz model of hyperbolic geometry," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3779–3788.

[39] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.

[40] C. Gulcehre et al., "Hyperbolic attention networks," 2018, *arXiv:1805.09786*.

[41] Z. Gao, Y. Wu, Y. Jia, and M. Harandi, "Curvature generation in curved spaces for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8691–8700.

[42] X. Fu et al., "ACE-HGNN: Adaptive curvature exploration hyperbolic graph neural network," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2021, pp. 111–120.

[43] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6417–6427.

[44] J. Yan, L. Luo, C. Deng, and H. Huang, "Unsupervised hyperbolic metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12460–12469.

[45] A. Pal, M. van Spengler, G. Maria D'Amely di Melendugno, A. Flaborea, F. Galasso, and P. Mettes, "Compositional entailment learning for hyperbolic vision-language models," 2024, *arXiv:2410.06912*.

[46] H. Li, Z. Chen, Y. Xu, and J. Hu, "Hyperbolic anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 17511–17520.

[47] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, "Hyperbolic geometry of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 82, no. 3, Sep. 2010, Art. no. 036106.

[48] N. Linial, A. Magen, and M. E. Saks, "Low distortion Euclidean embeddings of trees," *Isr. J. Math.*, vol. 106, no. 1, pp. 339–348, Dec. 1998.

[49] S. Kim, B. Jeong, and S. Kwak, "HIER: Metric learning beyond class labels via hierarchical regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19903–19912.

[50] A. A. Ungar, *Analytic Hyperbolic Geometry: Mathematical Foundations and Applications*. Singapore: World Scientific, 2005.

[51] X. Wen, B. Zhao, and X. Qi, "Parametric classification for generalized category discovery: A baseline study," 2022, *arXiv:2211.11727*.

[52] X. Yang, X. Hu, S. Zhou, X. Liu, and E. Zhu, "Interpolation-based contrastive learning for few-label semi-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2054–2065, Feb. 2024.

[53] H. Wu, X. Li, and K.-T. Cheng, "Exploring feature representation learning for semi-supervised medical image segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16589–16601, Nov. 2024.

[54] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3238–3247.

[55] Z. Yang, M. Bastan, X. Zhu, D. Gray, and D. Samaras, "Hierarchical proxy-based loss for deep metric learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 449–458.

[56] Z. Zheng, X. Wang, N. Zheng, and Y. Yang, "Parameter-efficient person re-identification in the 3D space," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7534–7547, Jun. 2024.

[57] N. Monath, M. Zaheer, D. Silva, A. McCallum, and A. Ahmed, "Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 714–722.

[58] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[59] P. Welinder et al., "Caltech-UCSD birds 200," California Inst. Technol., Pasadena, CA, USA, Tech. Rep., 2010.

[60] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 554–561.

[61] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*.

[62] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, "AutoNovel: Automatically discovering and learning novel visual categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6767–6781, Oct. 2022.

[63] K. Cao, M. Brbić, and J. Leskovec, "Open-world semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[64] J. Otholt, C. Meinel, and H. Yang, "Guided cluster aggregation: A hierarchical approach to generalized category discovery," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2606–2615.

[65] A. Banerjee, L. S. Kallooriyakath, and S. Biswas, "AMEND: Adaptive margin and expanded neighborhood for efficient generalized category discovery," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2090–2099.

[66] B. Zhao, X. Wen, and K. Han, "Learning semi-supervised Gaussian mixture models for generalized category discovery," 2023, *arXiv:2305.06144*.

[67] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9650–9660.

[68] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

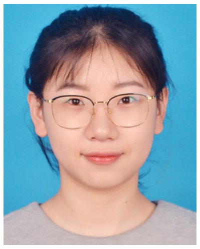
**Yu Duan** received the M.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2021 and 2024, respectively.

He is currently working as a Post-Doctoral with Xidian University, Xi'an. His research interests focus on clustering, semi-supervised learning, and representation learning.

**Feiping Nie** (Senior Member, IEEE) received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He is currently a Full Professor with Northwestern Polytechnical University, Xi'an, China. He has published more than 100 papers in the following journals and conferences: IEEE Transactions on Pattern Analysis and Machine Intelligence, *International Journal of Computer Vision*, IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Knowledge and Data Engineering, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited more than 20 000 times and the H-index is 84. His research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dr. Nie is currently serving as an Associate Editor or a PC Member for several prestigious journals and conferences in the related fields.



**Huimin Chen** received the B.S. degree in software engineering and the M.S. degree in computer technology from Northwestern Polytechnical University, Xi'an, China, in 2019 and 2022, respectively.

Her current research interests include machine learning and its applications, including clustering, and dimensionality reduction.

**Zhanxuan Hu** is currently an Associate Professor with the School of Information Science and Technology, Yunnan Normal University, Kunming, China. His research interests include clustering, image representation learning, and ReID.

**Rong Wang** received the B.S. degree in information engineering and the M.S. degree in signal and information processing from Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004 and 2007, respectively, and the joint Ph.D. degree in computer science from Xi'an Research Institute of Hi-Tech, and the Department of Automation, Tsinghua University, Beijing, China, in 2013.

He is currently an Associate Professor at the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an. His research interests focus on machine learning and its applications.

**Xuelong Li** (Fellow, IEEE) is the Chief Technology Officer (CTO) and Chief Scientist of the China Telecom.