Towards On-the-Fly Novel Category Discovery in Dynamic Long-Tailed Distributions

Hoin Jung, Xiaoqian Wang
Elmore Family School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907
{jung414, joywang}@purdue.edu

Abstract

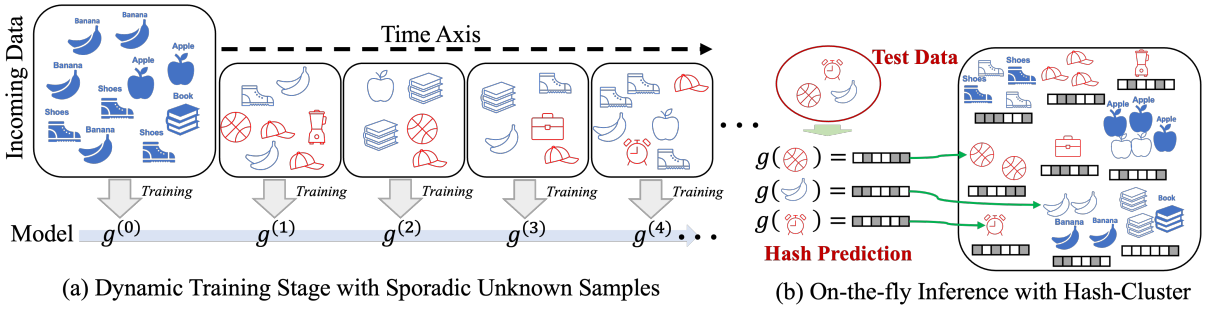
As the diversity of real-world object categories increases, the need for sophisticated classification methods also grows. However, Novel Category Discovery (NCD), which aims to predict unseen categories, often falls short in scenarios where new categories are constantly updated and data distributions are potentially biased. In addition, existing dynamic NCD approaches assume that incremental stages introduce a fixed number of new classes and often overlook distributional biases in real-world classes. To address these limitations, we propose a novel framework, ***N****ovel* ***C****ategory* ***D****iscovery for* ***D****ynamic* ***L****ong-****T****ailed distribution (****NCD-DLT****), which deals with the more realistic and challenging scenario where imbalanced, unlabeled data are introduced incrementally and sporadically over time. Unlike conventional methods requiring k-means clustering on all test samples, our approach identifies novel categories on-the-fly, predicting categories for individual data points as they arrive. We propose an advanced hash-based clustering technique, leveraging a double-hashing strategy to mitigate collisions and incorporating a greedy hash regularization loss for sparse representations to enhance clustering capabilities. Furthermore, we implement distillation losses during training to preserve the model's discriminative power across stages without forgetting prior knowledge. Finally, we introduce a novel graph merging algorithm based on the Hash Hamming Graph, revealing the dataset's clustering structure. It serves as a mechanism for pseudo-labeling in training and acts as a post-processing tool, reallocating less confident samples to more appropriate clusters. Our comprehensive approach addresses the limitations of existing NCD methods in the dynamic scenario of novel category discovery in long-tailed distributions, demonstrating improved accuracy for both uniform and long-tailed scenarios.*

1. Introduction

Object categories in the real world are constantly evolving and dynamically changing over time [35]. Additionally, the number of instances in each category often exhibits significant bias, typically forming long-tailed distributions [1, 32]. In such environments, traditional classification models fall short because they can only classify previously seen classes. For example, in applications like vehicle classification [17, 30] and E-commerce [24, 25], traditional classification methods require arduous human effort to label new data, necessitating training on unlabeled samples that potentially belong to unseen categories. Moreover, new types of vehicles and products continuously emerge, increasing the number of unknown classes over time, where labels for new data are not provided. In addition, ensuring uniform sample distribution across categories is challenging, as the number of instances in each category may reflect people's preferences, leading to some categories being very rare. Therefore, our objective is to design a classification framework capable of correctly discovering and identifying previously unseen categories in dynamic and imbalanced environments, where data is provided incrementally and sporadically.

However, although Novel Class Discovery (NCD) frameworks [22, 27] aim to predict previously unseen categories, this remains challenging under such dynamic and imbalanced conditions. Moreover, the majority of existing NCD works on dynamic environment [29, 33] operate under the strong assumption that the number of new classes within newly arrived data expand incrementally, and known prior to the training. For example, in the CIFAR100 dataset [18], it is presupposed that 80 classes are labeled and that the remaining 20 classes are introduced incrementally in four stages, with 5 classes at each stage [29]. While this incremental assumption presents its own set of challenges, it may

The code is available on https://github.com/HoinJung/NCD-DLT.



(a) Dynamic Training Stage with Sporadic Unknown Samples (b) On-the-fly Inference with Hash-Cluster

Figure 1. Blue items represent known classes, and red items represent unknown classes where labels are never provided. Color-filled items in the initial stage denote labeled samples, while unfilled items in incremental stages denote unlabeled samples. In real-world classification scenarios (e.g., E-commerce), new data continuously arrive without labels, where they might belong to unknown classes as objects in the real world are constantly evolving. Additionally, the data distribution could potentially be biased. For example, (a) the known banana class has 8 samples throughout the training, while the unknown clock class has only one sample in 4-th incremental stage. (b) Test data continues to arrive and is clustered immediately, even if it belongs to an unknown class.

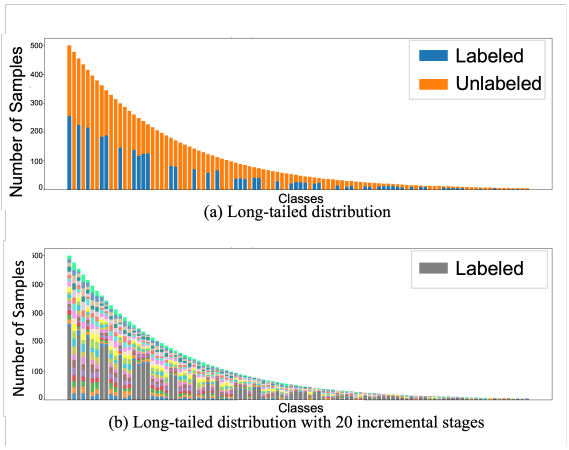
not always reflect open-world scenarios where the class of arriving data in the future is not determined.

Furthermore, these studies often assume a uniform distribution within the training dataset, with the same number of instances in each class. Despite this, some research has tackled NCD within a long-tailed distribution [1,32], where the class distribution is extremely biased. However, the existing literature often overlooks dynamic environments within long-tailed distributions, primarily focusing on static settings where all labeled and unlabeled data are readily available simultaneously.

To this end, we propose a scenario that poses a greater challenge reflecting more realistic circumstances. In our scenario, the dataset is characterized by a long-tailed distribution, and unlabeled data are introduced in a sporadic and incremental manner over multiple periods, as illustrated in Figure 1 (a). In such circumstances, samples from the minority class may be introduced sporadically, leading to a situation where the model could easily forget them. (Figure 2.) Despite the significance of this scenario, no prior study has specifically addressed this problem, and we are the first to tackle this challenging setting.

Additionally, we address a common limitation in existing NCD frameworks, which typically rely on k-means clustering with all test samples available and require prior knowledge of k. This is impractical for real-world applications, especially in long-tailed and dynamic scenarios where novel classes appear sporadically. Determining k through binary search, as done in [1, 27, 29], is computationally expensive and often inaccurate in estimating the total number of classes. Even when k is given, k-means clustering requires all test samples to be processed at once, preventing inductive inference on individual samples and adding significant computational cost.

In contrast, our approach recognizes novel categories in real time as they are encountered (Figure 1 (b)) without



(a) Long-tailed distribution

(b) Long-tailed distribution with 20 incremental stages

Figure 2. The dynamic setting within a long-tail distribution poses an increased challenge. Specifically, (a) showcases the typical long-tailed distribution encountered in a static NCD scenario. Conversely, (b) reveals the dynamics of an NCD scenario within a long-tailed distribution, only the labeled dataset (gray) is introduced in the initial stage. Subsequently, samples represented by various colors are provided at each stage, with each color denoting a different incremental stage, reflecting real-world scenarios where data availability and class representation can vary over time.

any prior knowledge. To achieve this, we incorporate a hash-based clustering approach inspired by [5], where the model generates a hash code indicative of a cluster. By ensuring the hash code can distinguish between old and new categories, our model effectively handles dynamic, evolving data. This allows for on-the-fly inference without prior knowledge, overcoming the limitations of existing methods.

However, the efficacy of hash-based clustering is inherently limited, as it relies solely on contrastive loss during the training phase. To overcome this limitation, we introduce an advanced hash-based clustering technique that incorporates a *double-hashing* strategy to mitigate hash collisions—instances where distinct classes are erroneously as-

	Not requiring prior knowledge (# of novel classes)	Dynamic scenario (Incremental learning)	Sporadicalness in incremental stages	Long-tailed distribution	On-the-fly inference
GCD [27]	-	-	-	-	-
OCD [5]	✓	-	-	-	✓
GM [33]	-	✓	-	-	-
MetaGCD [29]	-	✓	-	-	-
BaCon [1]	-	-	-	✓	-
Ours (NCD-DLT)	✓	✓	✓	✓	✓

Table 1. Requirements for each NCD framework. NCD-DLT operates in the most challenging scenario compared to other methods. It handles a constantly changing training environment, sometimes with few samples per class under a biased distribution, without prior knowledge of the number of classes, and provides real-time inference.

signed the same hash code. For a double-hashing, we employ two separate hash functions where their combined output constitutes the final hash code. Additionally, we introduce a *greedy hash regularization loss* as an objective function to foster the development of sparse representations. During the training phase, we incorporate both static and dynamic distillation losses to preserve the model's capacity to classify old classes from preceding stages effectively.

Lastly, we propose an innovative hash-graph merging algorithm. Although a hash code denotes a distinctive cluster, it solely relies on binary hash output without an explicit clustering algorithm. Therefore, we construct a Hash Hamming Graph (HHG) using hash outputs, where hash codes serve as nodes. The size of a node denotes the number of samples in the cluster, and edges are established between nodes separated by a Hamming distance of one. This graph effectively illustrates the overarching clustering structure of the dataset. Through the identification and merging of less confident, smaller nodes into their nearest, larger counterparts, the hash-graph merging algorithm fulfills a dual role: it acts as a pseudo-label tool during training for supervised contrastive learning [15] with unlabeled data, and as a post-processing mechanism to reassign less confident samples to their nearest, larger nodes following inference. The details of HHG are introduced in Section 4.3.

The performance of our proposed comprehensive framework, Novel Category Discovery for Dynamic Long-Tailed Distribution (NCD-DLT), demonstrates its superiority in challenging scenarios involving a number of unknown categories. This achievement underscores our contribution to enhancing the adaptability and effectiveness of NCD systems in real-world settings, particularly in dynamic and long-tailed distributions with real-time inference. We summarize the contributions of this paper as follows:

- NCD-DLT addresses the challenge of NCD within a dynamic and imbalanced data scenario, which has not been thoroughly explored in previous work.
- NCD-DLT eliminates the need for prior knowledge of k and avoids the use of binary search for k, enabling real-time application.
- NCD-DLT introduces a novel hash-based framework, enhanced through the integration of knowledge distillation and a new graph-merging algorithm.

2. Related Work

2.1. Novel Category Discovery (NCD)

NCD seeks to identify new classes within unlabeled data by leveraging the class information from a labeled dataset [8, 11]. In the generalized NCD framework [2, 27, 28], the unlabeled dataset encompasses both known and novel classes, serving as a fundamental component of the NCD society. [22] proposed a scenario in which only 10% of the samples in known classes are labeled. However, prior studies often presuppose a uniform class distribution across the dataset, relying solely on the vanilla benchmark datasets for analysis. [32] and [1] introduce a scenario characterized by a long-tailed distribution, highlighting class imbalance as a more realistic setting in open-world scenarios.

The complexity of these scenarios is further underscored in a continual learning context, where labeled and unlabeled datasets are provided sequentially rather than simultaneously. This includes a one-step continual training on unlabeled samples after initial training on labeled datasets [14, 16, 23]. Additionally, studies such as [29, 33, 35] assume a sequential scenario where unlabeled samples are incrementally provided in multiple stages. However, existing incremental approaches typically assume a predetermined, fixed number of new classes at each stage, requiring prior knowledge of the total number of novel classes.

In this paper, we address two primary challenges: the highly skewed class distribution known as the long-tailed distribution, and the incremental scenario where samples from the unlabeled pool are selected sporadically at each stage. We assume that we do not have any previous information about how many new classes there are, even not requiring to figure out the total number of classes by binary search as was done in the literature [1, 2, 8, 11, 22, 23, 27, 28, 32, 33].

2.2. Hash Function

Deep learning applications enhanced with hashing functions [6,7,21,26,34] have proven effective in image retrieval tasks. These approaches transform deep features into compact binary codes, preserving semantic similarities. Such a transformation drastically reduces storage requirements and accelerates the retrieval process, benefits that are particularly valuable for managing large-scale datasets prevalent in deep learning scenarios. Notably, [5] bridged the gap between deep hashing and NCD, promoting the use of binary hash codes to represent clusters. This method facilitates model recognition of distinct categories without memorizing the specific characteristics of each category in storage, thereby enabling on-the-fly inference of unlabeled queries independent of the test sample size. This technique represents a departure from traditional NCD methodologies, which typically rely on performing k-means clustering on all test samples simultaneously, a process that can lead to one sample's prediction being influenced by others in the query set. Our framework adopts [5] as a baseline, identifies its limitations, and implements enhancements to improve the discovery of novel classes in Section 4.1.

In summary, Table 1 demonstrates the effectiveness of the proposed NCD-DLT approach in dynamic scenarios characterized by random class occurrences sporadically within long-tailed data distributions. Notably, this approach does not necessitate prior knowledge of the number of novel classes, nor does it require a time-intensive calibration through binary search to find the number of clusters for k-means clustering, thereby enabling on-the-fly inference.

3. Problem Setting

In this paper, we delineate a comprehensive and realistic problem setting within the NCD framework. Specifically, we have a labeled set $\mathcal{D}_L = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_L} \in \mathcal{X} \times \mathcal{Y}_L$ and unlabeled sets $\mathcal{D}_U = \{\mathcal{D}_U^t\}_{t=1}^{T} \in \mathcal{X} \times \mathcal{Y}_U$ where $\mathcal{D}_U^t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_U^t}$ with $\mathcal{Y}_L \subset \mathcal{Y}_U$. The overall class distribution of the entire dataset adheres to a long-tailed setting, such that the number of samples in each class N_i decreases according to $N_i \leftarrow N_i \times \left(\frac{1}{\rho}\right)^{i/C}$, where i is the class index, C is the total number of classes, and ρ is an imbalance ratio. We assume that labels in $\mathcal{D}_U^t$ are unavailable, and the class distribution at each stage t is randomly selected and remains unknown. For the experiments, we choose two different imbalance ratio values $\rho = 20$ and $\rho = 100$ besides the uniform distribution scenario.

3.1. Dataset

We utilize three widely recognized datasets for both standard NCD and their long-tailed variants: CIFAR10 [18], CIFAR100 [18], and TinyImageNet [20], containing 10, 100, and 200 classes, respectively. Each dataset is divided into three parts: a labeled-known set (i.e., labeled data from known classes), an unlabeled-known set (i.e., unlabeled data from known classes), and an unlabeled-unknown set (i.e., unlabeled data from novel classes). Half of the classes are designated as unknown. Within the known classes, 50% are unlabeled, resulting in only 25% of the entire dataset comprising the labeled-known set, which is utilized for initial training. The remaining 75% of the entire dataset, a union of the unlabeled-known and unlabeled-unknown sets, is mixed and randomly introduced during the incremental training stages. Therefore, with the total number of stages set to 20, only 3.75% of the training dataset is provided at each stage.

4. Proposed method

NCD-DLT combines three complementary strategies, hash-code learning, knowledge distillation, and graph merging to address the challenge of NCD in dynamic, long-tailed scenarios. While hash-code learning serves as the core method for efficiently discovering novel categories, knowledge distillation and graph merging play crucial roles in preventing the forgetting of previously learned classes and enhancing performance. These three components work together to provide a robust framework that not only adapts to new data incrementally but also maintains high accuracy across both old and newly introduced categories.

4.1. Hash-code Learning

A hash code demonstrates its efficacy in image retrieval tasks [7, 31] due to its capability for efficient similarity search. [5] leveraged hashing by adopting a neural network-based hash function as a cluster head, assigning the representation from an encoder to a hash code, where each hash code signifies a unique cluster. However, hash-based clustering inherently faces the issue of hash collisions, where two distinct classes may be assigned the same hash code if their images have similar features and the training data is insufficient to distinguish them. To mitigate this problem, we introduce two advanced approaches: double hashing [9] and greedy hash regularization [26].

Double hashing enhances the diversity of the hash code by employing a secondary hash function that is independent of the first, thus increasing the diversity of hash mappings. This approach integrates two different perspectives of the data, leading to a more generalized representation by reducing the bias that a single hash function might introduce. Specifically, let $\boldsymbol{z}_1$ be a representation extracted from the encoder $\mathcal{E}$, such that $\boldsymbol{z}_1 = \mathcal{E}(\boldsymbol{x})$. Then,

$$
\begin{aligned}
\boldsymbol{z}_2 &= \boldsymbol{W}_2(\boldsymbol{z}_1) \\
\boldsymbol{h}_k &= \tanh(\boldsymbol{W}_k^h(z_k)),\ \boldsymbol{v}_k = \mathrm{abs}(\boldsymbol{W}_k^v(z_k)) \quad k = 1, 2 \\
\boldsymbol{f} &= \frac{\boldsymbol{h}_1 + \boldsymbol{h}_2}{2} \odot \frac{\boldsymbol{v}_1 + \boldsymbol{v}_2}{2},
\end{aligned} \tag{1}
$$

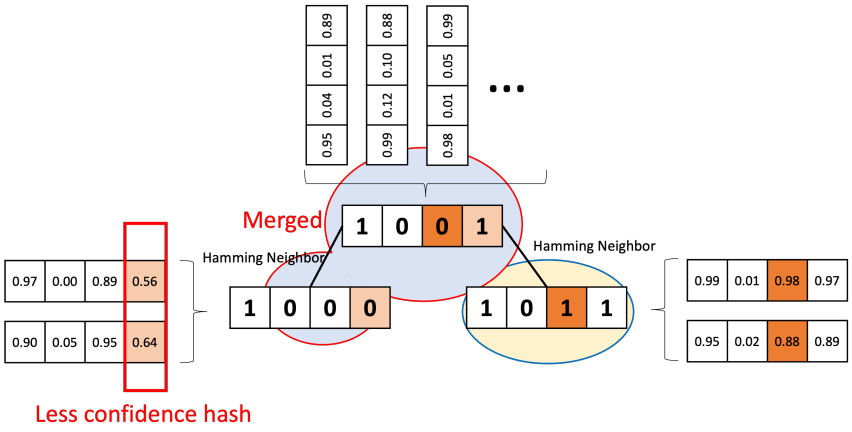


Figure 3. The Hash Hamming Graph represents the structural relationship of classes in the dataset. Each node corresponds to a cluster, the node size reflects the number of instances within the cluster, and an edge connects two nodes if the Hamming distance between them is 1. Additionally, a cluster with a less confident hash code implies that it is not distinctive enough to represent a unique class. Therefore, less confident hashes are merged into larger, more confident nodes, while highly confident hash codes are regarded as distinct classes.

where $\text{abs}(x) = \frac{x}{\tanh(x)}$ is a differentiable absolute function, and $\boldsymbol{W}_2$, $\boldsymbol{W}_k^h$, and $\boldsymbol{W}_k^v$ are linear transformations.

On the other hand, a greedy hash regularization loss encourages the learned features $\boldsymbol{f}_i$ from the hash function to be similar to the binary hash features $\boldsymbol{b}_i = \text{sign}(\boldsymbol{f}_i) \in \{-1, +1\}^L$, defined by

$$\mathcal{L}_i^{cos} = \frac{1}{N-1} \sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} \| \cos(\boldsymbol{f}_i, \boldsymbol{f}_k) - \cos(\boldsymbol{b}_i, \boldsymbol{b}_k) \|_2^2 \tag{2}$$

where N is the batch size, $\cos(\cdot, \cdot)$ denotes the cosine similarity between the features, and L is the hash code length adopted by [5]. Eq.(2) implies that if two images are assigned the same hash, their extracted features should also be similar. We can simplify the overall model function as $g = \mathcal{H}(\mathcal{E}(\boldsymbol{x}))$, where $\mathcal{H}$ represents the double-hash function described in Eq.(1).

4.2. Knowledge Distillation for not Forgetting

Let g_0 be the model in the initial stage, and g_t be the incremental model at time step t with a sample $\boldsymbol{x}_i$. In the initial stage, a known-labeled dataset is trained using supervised contrastive learning [15],

$$\mathcal{L}_i^l = -\frac{1}{|P_i|} \sum_{p \in P_i} \log \frac{\exp(g_0(\boldsymbol{x}_i) \cdot g_0(\boldsymbol{x}_p))}{\sum_{j=1}^{N} \mathbb{1}_{j \neq i} \exp(g_0(\boldsymbol{x}_i) \cdot g_0(\boldsymbol{x}_j))} \tag{3}$$

where N is the batch size, and $|P_i|$ represents the number of positive samples in the set P_i, which contains samples having the same label as the current instance i.

Following the initial stage, random unlabeled samples are incrementally provided. In each stage, the unlabeled samples undergo training with unsupervised contrastive learning [10],

$$L_i^u = -\log \frac{\exp(g_t(\boldsymbol{x}_i^u) \cdot g_t(\boldsymbol{x}_i'^u))}{\sum_n \mathbb{1}_{[n \neq i]} \exp(g_t(\boldsymbol{x}_i^u) \cdot g_t(\boldsymbol{x}_n^u))}. \tag{4}$$

where $\boldsymbol{x}_i'^u$ denotes a different view of augmentation of $\boldsymbol{x}_i^u$.

In the incremental stages, which could number many stages (e.g., $T \geq 20$), if the model g_t ($t = 1, 2, \cdots, T$) is trained solely with the current subset $\mathcal{D}_t$ by Eq.(4), it might overfit to the current training data only. To prevent the model from forgetting knowledge acquired from the initial stage and previous incremental stages, we employ two types of distillation loss: *static distillation loss* and *dynamic distillation loss*.

To balance the static and dynamic distillation losses, we load an equal number of labeled (B^l) and unlabeled (B^u) samples in the batch during the dynamic stages, i.e., $\boldsymbol{x}_j^l \in B^l$ and $\boldsymbol{x}_i^u \in B^u$. To preserve initial stage's knowledge, the static distillation loss [13] is defined as:

$$\mathcal{L}_j^{sd} = 1 - \cos(g_0(\boldsymbol{x}_j^l), g_t(\boldsymbol{x}_j^l)), \forall t \geq 1. \tag{5}$$

Eq.(5) aims to maintain the similarity between the outputs from the initial model at $t = 0$ and the current model at time $t \geq 1$ for the re-loaded labeled samples. Additionally, to preserve the information from the previous stage $(t-1)$, we use the dynamic distillation loss,

$$\mathcal{L}_i^{dd} = 1 - \cos(g_{t-1}(\boldsymbol{x}_i^u), g_t(\boldsymbol{x}_i^u)), \forall t \geq 2. \tag{6}$$

Eq.(6) focuses on preserving the similarity between the outputs from the previous model (g_{t-1}) and the current model (g_t) during unsupervised contrastive learning for unlabeled samples.

4.3. Hash Hamming Graph and Graph Merging

The strategies of double hashing and greedy hash regularization are beneficial for avoiding hash collisions. However, these strategies may result in an excessive number of

hash-clusters, implying that instances from the same cluster could be assigned more than two different hashes. We hypothesize that if a class is separated into several hash codes, the hash outputs might not be sufficiently distinctive to represent an independent cluster confidently. Building on this concept, we propose the creation of a Hash Hamming Graph (HHG). This graph structure represents relationships between clusters, where each node corresponds to a cluster, the node size reflects the number of instances within the cluster, and an edge connects two nodes if the Hamming distance between them is 1, as illustrated in Figure 3. This structure enables us to uncover the underlying cluster structure of the dataset.

By utilizing the HHG, we identify less confident nodes characterized by their size being below a specific threshold and their code probability indices being close to the decision boundary of each code (i.e., 0.5). Even if a node is small, we consider it a distinct cluster if its confidence level is high. Our goal is to merge less confident nodes with appropriate, larger neighboring nodes. For each identified small and less confident node, we explore potential nodes for merging candidates by switching the binary code of the less confident indices (Algorithm 2 in Appendix A). By evaluating the soft-Hamming distance between the current node and each candidate, as well as considering their sizes, we determine the best candidates for merging. If no suitable candidate is found, we merge the node with the largest neighboring node (Algorithm 3 in Appendix A). The comprehensive algorithm is presented in Algorithm 1 in Appendix A and Figure 3. The visualized example is described in Figure 4.

The merged graph can be leveraged in two distinct manners: through pseudo-labeling during the training process and as a post-processing step following inference stage. Firstly, the formation of merged clusters suggests that instances within the merged node are highly probable to belong to the same class. This insight allows us to utilize pseudo-labels in conjunction with supervised contrastive learning, following the same formulation as Eq.(3), denoted $\mathcal{L}_i^{u,p}$. This approach aims to encourage the model to cluster features from the same group more closely together while distancing them from those of other groups. We clarify the objective function for each stage:

$$\mathcal{L}_i^{total} = \begin{cases} \mathcal{L}_i^{l} + \mathcal{L}_i^{reg} + \mathcal{L}_i^{cos} & t = 0 \\ \mathcal{L}_i^{u} + \mathcal{L}_i^{u,p} + \mathcal{L}_i^{reg} + \mathcal{L}_i^{cos} + \lambda(\mathcal{L}_j^{sd}) & t = 1 \\ \mathcal{L}_i^{u} + \mathcal{L}_i^{u,p} + \mathcal{L}_i^{reg} + \mathcal{L}_i^{cos} + \lambda(\mathcal{L}_j^{sd} + \mathcal{L}_i^{dd}) & t > 1, \end{cases} \tag{7}$$

where λ is a hyperpameter, and $\mathcal{L}_i^{reg}$ is a L1 regularizer suggested in [5] to encourage the sparsity of the hash output, $\mathcal{L}_i^{reg} = -|\boldsymbol{h}_i|$. After training, test samples might be associated with less confident hashes. Applying the merging algorithm to the test set as a form of post-processing can enhance accuracy by ensuring similar features are grouped together, potentially correcting instances where the initial hash assignment might not reflect true class similarity. This step significantly improves the model's performance by refining its predictions based on more nuanced, post-processed cluster assignments. The hyperparameters for HHG and their impact are introduced in Appendix B.3.

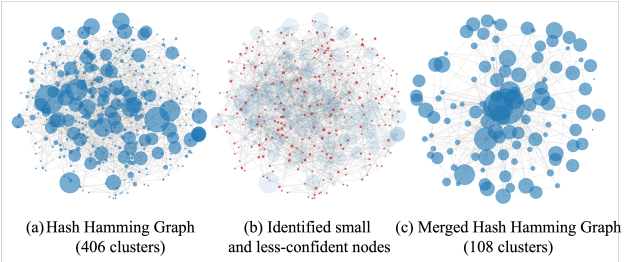


Figure 4. (a) The example of Hash Hamming Graph (HHG) on CIFAR100 dataset. (b) Identified small and less-confident nodes by Algorithm 1 (red nodes). (c) The merged graph can be used as a pseudo-labeling algorithm during the training, and post-processing during the inference stage.

5. Experimental Result

5.1. Implementation Details

For the baseline of our proposed method, we adopt OCD [5] with the same learning rate ($lr = 0.01$), where hash code length is determined as $L = 12$ by ablation studies. (See Appendix B.1). We use a pre-trained DINO [3] with ViT-B-16 [4] as a backbone network, consistent with GCD [27], OCD [5], MetaGCD [29], and BaCon [1]. In GM [33], MOCO [12] is employed as the backbone network. For all methods, we conduct the initial stage on the labeled dataset for 50 epochs, followed by 10 epochs for each incremental stage, totaling 20 stages. Except for variations in the incremental dataset and epochs, we adhere to the original implementations of each method.

In the inference stage, we operate under the assumption that not all test samples are immediately accessible, reflecting real-world conditions where a model may need to predict labels for only a subset of samples, sometimes even a single sample. NCD-DLT generates a hash code outcome, representing a cluster. Conversely, existing works in NCD often rely on k-means clustering at the inference stage, requiring prior knowledge of the number of clusters k, which contradicts the NCD setting where class labels of unlabeled samples are unknown. [27] proposed using binary search to find the optimal k, but this method does not accurately determine the number of clusters and is time-consuming. For fair comparison, we pre-define the number of clusters for each dataset by performing binary search on features extracted from the entire dataset using a pre-trained DINO. The k determined through binary search is applied to GCD, GM, MetaGCD, and BaCon. Additionally, even with a predefined number of clusters, applying k-means clustering di-

Original Dataset	CIFAR10			CIFAR100			TinyImageNet		
Model	All	Old	New	All	Old	New	All	Old	New
GCD	0.3043	0.3112	0.2974	0.3797	0.4284	0.3310	0.3872	0.4392	0.3352
OCD	0.8773	**0.9820**	0.7726	0.5105	0.7486	0.2724	0.4700	**0.7444**	0.1956
GM	0.2239	0.2774	0.1704	0.2672	0.4124	0.1220	0.1914	0.2848	0.0980
BaCon	0.7561	0.8642	0.6480	0.5274	0.6540	0.4008	0.4840	0.5256	**0.4424**
MetaGCD	0.8796	0.8232	0.9360	0.4156	0.4200	**0.4112**	0.3622	0.4628	0.2616
Ours (NCD-DLT)	**0.9733**	0.9792	**0.9674**	**0.5966**	**0.8218**	0.3714	**0.5286**	0.7208	0.3364
Long-tailed ($\rho = 20$)	CIFAR10			CIFAR100			TinyImageNet		
Model	All	Old	New	All	Old	New	All	Old	New
GCD	**0.9220**	0.9020	**0.9420**	0.2037	0.2202	0.1872	0.3107	0.3407	0.2824
OCD	0.8484	**0.9676**	0.7292	0.4528	0.6438	0.2618	0.3916	0.6048	0.1784
GM	0.2131	0.2762	0.1500	0.2566	0.3594	0.1538	0.1690	0.2272	0.1108
BaCon	0.7241	0.8290	0.6192	0.3950	0.4554	0.3346	0.4022	0.5160	0.2884
MetaGCD	0.7999	0.7706	0.8292	0.4792	0.5002	**0.4582**	0.3466	0.4040	0.2892
Ours (NCD-DLT)	0.8878	0.9322	0.8434	**0.5204**	**0.6910**	0.3498	**0.4690**	**0.6264**	**0.3116**
Long-tailed ($\rho = 100$)	CIFAR10			CIFAR100			TinyImageNet		
Model	All	Old	New	All	Old	New	All	Old	New
GCD	0.8946	0.9386	0.8878	0.2201	0.2352	0.2050	0.2954	0.3083	0.2845
OCD	0.8536	**0.9586**	0.7486	0.3981	0.5292	0.2670	0.3474	0.5328	0.1620
GM	0.1693	0.1824	0.1562	0.2013	0.2600	0.1426	0.1868	0.2444	0.1292
BaCon	0.5636	0.5788	0.5484	0.3900	0.4584	0.3216	0.3692	0.4412	**0.2972**
MetaGCD	**0.9158**	0.8842	**0.9474**	0.4258	0.4352	**0.4164**	0.3352	0.3944	0.2524
Ours (NCD-DLT)	0.9121	0.9424	0.8818	**0.4499**	**0.5368**	0.3630	**0.3878**	**0.5352**	0.2404

Table 2. Experimental results for original and long-tailed datasets with Hungarian algorithm. The best results are marked in **bold**, and the second best results are marked by underline.

rectly to a subset of test samples is impractical. For on-the-fly inference, we conduct k-means clustering on the training set to identify each cluster's centroid. Test sample predictions are then determined by the nearest centroid's cluster. Note that OCD and NCD-DLT don't require the prior knowledge for k and k-means clustering.

5.2. Evalutaion Metric

Our evaluation employs Hungarian algorithm [19] as utilized by [8]. In the testing phase, the assignment of predicted labels to ground truth is inherently uncertain due to the absence of ground truth for novel classes during training. We organize all clusters by their size to optimize the match between predicted labels and ground truth through the best permutation. The Hungarian method involves segregating samples into 'New' and 'Old' subsets based on their ground truth labels, followed by separate accuracy calculations for each subset. This approach offers a detailed view of the model's performance on both 'New' and 'Old' subsets. The accuracy for ground truth label y_i and predicted label $\hat{y}_i$, calculated using the Hungarian algorithm, is defined as $ACC = \max_{p \in \mathcal{P}(\mathcal{Y})} \frac{1}{N} \sum_{i=1}^{N} (\mathbb{1}\{y_i = p(\hat{y}_i)\})$, where N represents the total number of test samples, and $\mathcal{P}(\mathcal{Y})$ denotes the set of all possible permutations of the class labels, with $\mathcal{Y} = \mathcal{Y}_L \cup \mathcal{Y}_U$ signifying the union of labeled and unlabeled class labels. As suggested in [29], the main metric is the accuracy on **'All'** classes, indicating the balanced accuracy between 'Old' and 'New' classes. The experimental results are introduced in Table 2. The detailed intrepretation of the evaluation metric is described in Appendix C.

5.3. Result Analysis

Table 2 presents the experimental results of various state-of-the-art methods on original and long-tailed versions of CIFAR10, CIFAR100, and TinyImageNet datasets, evaluated using the Hungarian algorithm. For the most essential metric, 'All' accuracy, NCD-DLT consistently outperforms the other methods across all datasets. Specifically, under a uniform distribution, NCD-DLT achieves the highest 'All' accuracy across all datasets, significantly surpassing the second-best method. In long-tailed scenarios, our method continues to demonstrate robust and balanced performance. It achieves the best 'All' accuracy for TinyImageNet and CIFAR100 at both imbalance levels, while for CIFAR-10, it secures second-best results, with only marginally lower ac-

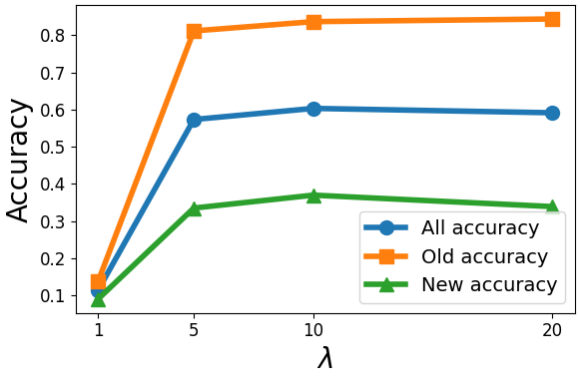


Figure 5. The impact of λ in Eq.(7).

curacies than the top-performing method. Although NCD-DLT may not always achieve the best performance in every case, it consistently ranks first or second across various datasets and imbalance ratios, whereas other methods often exhibit significantly worse results.

Furthermore, Figure 4 highlights the effectiveness of utilizing the Hash Hamming Graph combined with graph merging techniques. By identifying and merging smaller and less confident nodes, the refined Hash Hamming Graph contains 108 clusters, closely approximating the actual number of clusters, 100, in CIFAR100 dataset. This result is significantly more accurate than that achieved through binary search, which identified only 74 clusters.

5.4. Ablation Study

5.4.1 The Impact of λ

To assess the impact of λ in Eq.(7), we conduct an ablation study by adjusting λ's value within $1, 5, 10, 20$ with CIFAR100 dataset. As λ increases, the model more focuses on preserving previously acquired knowledge, which may impede its capacity to learn information from new data effectively. As depicted in Figure 5, the optimal λ value is determined to be 10. The detailed impact of λ is introduced in Appendix B.2.

5.4.2 The Impact of Each Component in NCD-DLT

To assess the impact of individual components within our comprehensive methodology, we conduct ablation studies on long-tailed CIFAR-100 dataset with $\rho = 100$, as presented in Table 3. NCD-DLT incorporates double-hashing and a greedy hash regularization loss to leverage the ability of hash-basesd clustering, and utilize the Hash Hamming Graph and graph-merging algorithm as pseudo-labeling during the training and post-processing. At first, we evaluate the influence of each element, double-hashing and greedy hash regularization loss. Similarly, we examine the effects of pseudo-labeling and post-processing through the

Double Hashing	Greedy Loss	All	Old	New
-	-	0.3999	0.5096	0.2902
✓	-	0.3981	0.4982	0.2980
-	✓	0.4103	**0.5268**	0.2938
✓	✓	**0.4201**	0.5142	**0.3260**
Pseudo-Labeling	Post-Processing	All	Old	New
-	-	0.4178	0.5228	0.3128
✓	-	0.4201	0.5142	0.3260
-	✓	0.4222	0.5000	0.3444
✓	✓	**0.4499**	**0.5368**	**0.3630**

Table 3. The impact of each component in NCD-DLT

Hash Hamming Graph and graph merging. This study validates the contribution of each component by comparing the 'New' accuracy which indicates successful discovery of novel categories. Our findings indicate that each component enhances the 'New' accuracy, with the combination of all elements yielding the best results, highlighted in bold.

6. Conclusion

In this study, we introduced a novel approach to NCD aimed at addressing the challenges posed by dynamic and incrementally evolving datasets with long-tailed distributions. Our method, which integrates enhanced hash-based clustering with a double-hashing strategy and a unique hash-graph merging algorithm, has demonstrated its effectiveness in identifying novel categories in complex, real-world scenarios. By mitigating hash collisions and leveraging a graph-based structure for cluster analysis, our approach overcomes the limitations of existing methods, offering a significant improvement in the accuracy of novel category discovery.

Our findings underscore the potential of NCD-DLT to adapt to highly skewed class distributions and manage incremental data introductions without prior knowledge of the number of new classes. This research contributes to the ongoing challenge of adapting machine learning to better reflect and navigate the dynamic nature of real-world class distributions such as E-commerce. Future work will focus on adapting the proposed method to more empirical data rather than benchmark datasets to demonstrate its effectiveness and adaptability in real environments.

Acknowledgements

This work was partially supported by the EMBRIO Institute, contract #2120200, a National Science Foundation (NSF) Biology Integration Institute, Purdue's Elmore ECE Emerging Frontiers Center, and NSF IIS #1955890, IIS #2146091, IIS #2345235.

References

[1] Jianhong Bai, Zuozhu Liu, Hualiang Wang, Ruizhe Chen, Lianrui Mu, Xiaomeng Li, Joey Tianyi Zhou, YANG FENG, Jian Wu, and Haoji Hu. Towards distribution-agnostic generalized category discovery. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3, 6, 14

[2] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *International Conference on Learning Representations*, 2021. 3

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[5] Ruoyi Du, Dongliang Chang, Kongming Liang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. On-the-fly category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11691–11700, 2023. 2, 3, 4, 5, 6, 11

[6] Shiv Ram Dubey, Satish Kumar Singh, and Wei-Ta Chu. Vision transformer hashing for image retrieval. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 4

[7] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, and Chee Seng Chan. Deep polarized network for supervised learning of accurate binary hashing codes. In *IJCAI*, pages 825–831, 2020. 4

[8] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9284–9292, 2021. 3, 7, 13

[9] Leo J Guibas and Endre Szemeredi. The analysis of double hashing. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, pages 187–191, 1976. 4

[10] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 5

[11] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6767–6781, 2021. 3

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 6

[13] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019. 5

[14] KJ Joseph, Sujoy Paul, Gaurav Aggarwal, Soma Biswas, Piyush Rai, Kai Han, and Vineeth N Balasubramanian. Novel class discovery without forgetting. In *European Conference on Computer Vision*, pages 570–586. Springer, 2022. 3

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 3, 5

[16] Hyungmin Kim, Sungho Suh, Daehwan Kim, Daun Jeong, Hansang Cho, and Junmo Kim. Proxy anchor-based unsupervised learning for continuous generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16688–16697, 2023. 3

[17] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 4

[19] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 7, 13

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 4

[21] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072, 2016. 4

[22] Jiaming Liu, Yangqiming Wang, Tongze Zhang, Yulu Fan, Qinli Yang, and Junming Shao. Open-world semi-supervised novel class discovery. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 4002–4010. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track. 1, 3

[23] Subhankar Roy, Mingxuan Liu, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Class-incremental novel class discovery. In *European Conference on Computer Vision*, pages 317–333. Springer, 2022. 3

[24] Uma Sawant and Vijay Gabale. Product discovery from e-commerce listings via deep text parsing. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 98–107, 2018. 1

[25] Qiang Su and Lu Chen. A method for discovering clusters of e-commerce interest patterns using click-stream data. *electronic commerce research and applications*, 14(1):1–13, 2015. 1

[26] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. *Advances in neural information processing systems*, 31, 2018. 4

[27] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7492–7501, 2022. 1, 2, 3, 6, 13

[28] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16590–16600, 2023. 3

[29] Yanan Wu, Zhixiang Chi, Yang Wang, and Songhe Feng. Metagcd: Learning to continually learn in generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1655–1665, 2023. 1, 2, 3, 6, 7, 14

[30] Chang Xu, Yingguan Wang, Xinghe Bao, and Fengrong Li. Vehicle classification using an imbalanced dataset based on a single magnetic sensor. *Sensors*, 18(6):1690, 2018. 1

[31] Wenjing Yang, Liejun Wang, and Shuli Cheng. Deep parameter-free attention hashing for image retrieval. *Scientific Reports*, 12(1):7082, 2022. 4

[32] Chuyu Zhang, Ruijie Xu, and Xuming He. Novel class discovery for long-tailed recognition. *Transactions on Machine Learning Research*, 2023. 1, 2, 3

[33] Xinwei Zhang, Jianwen Jiang, Yutong Feng, Zhi-Fan Wu, Xibin Zhao, Hai Wan, Mingqian Tang, Rong Jin, and Yue Gao. Grow and merge: A unified framework for continuous categories discovery. *Advances in Neural Information Processing Systems*, 35:27455–27468, 2022. 1, 3, 6, 14

[34] Zheng Zhang, Qin Zou, Yuewei Lin, Long Chen, and Song Wang. Improved deep hashing with soft pairwise similarity for multi-label image retrieval. *IEEE Transactions on Multimedia*, 22(2):540–553, 2019. 4

[35] Bingchen Zhao and Oisin Mac Aodha. Incremental generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19137–19147, October 2023. 1, 3