Towards Robust Incremental Learning under Ambiguous Supervision

Rui Wang[1,2] , **Mingxuan Xia**[1,2] , **Haobo Wang**[1,2] ,
Lei Feng[4] , **Junbo Zhao**[3] **Gang Chen**[3] , **Chang Yao**[1,2*]
[1]School of Software Technology, Zhejiang University
[2] Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[3] College of Computer Science and Technology, Zhejiang University
[4] School of Computer Science and Engineering, Southeast University, China
{r.wang, xiamingxuan, wanghaobo}@zju.edu.cn, fenglei@seu.edu.cn, {j.zhao, cg, changy}@zju.edu.cn

Abstract

Traditional Incremental Learning (IL) targets to handle sequential fully-supervised learning problems where novel classes emerge from time to time. However, due to inherent annotation uncertainty and ambiguity, collecting high-quality annotated data in a dynamic learning system can be extremely expensive. To mitigate this problem, we propose a novel weakly-supervised learning paradigm called Incremental Partial Label Learning (IPLL), where the sequentially arrived data relate to a set of candidate labels rather than the ground truth. Technically, we develop the Prototype-Guided Disambiguation and Replay Algorithm (PGDR) which leverages the class prototypes as a proxy to mitigate two intertwined challenges in IPLL, i.e., label ambiguity and catastrophic forgetting. To handle the former, PGDR encapsulates a momentum-based pseudo-labeling algorithm along with prototype-guided initialization, resulting in a balanced perception of classes. To alleviate forgetting, we develop a memory replay technique that collects well-disambiguated samples while maintaining representativeness and diversity. By jointly distilling knowledge from curated memory data, our framework exhibits a great disambiguation ability for samples of new tasks and achieves less forgetting of knowledge. Extensive experiments demonstrate that PGDR achieves superior performance over the baselines in the IPLL task.

1 Introduction

Modern deep models are mostly developed in curated and static benchmark datasets, but data in the real world typically emerge dynamically. This motivates the study of incremental learning (IL) [Cichon and Gan, 2015; Yuan *et al.*, 2022; Kim *et al.*, 2024] that enables models to learn from sequentially arriving tasks. Despite the flexibility, it is well known that deep models struggle to retain their known concepts, i.e., *catastrophic forgetting*, making it hard to gradually accumulate knowledge. To alleviate this problem, a

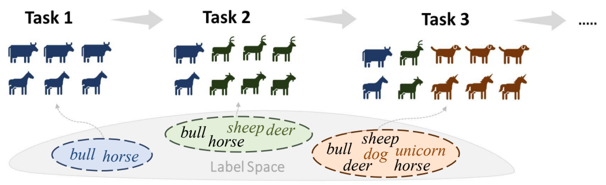


Figure 1: In the first task, all samples are from new classes, while subsequent tasks consist of samples containing both new and old classes. Each sample is assigned a set of candidate labels, ensuring the inclusion of the true label.

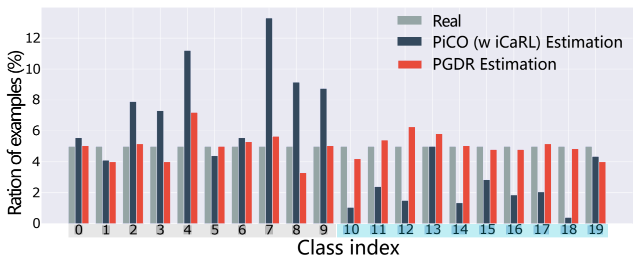
plethora of IL methods have been studied, including replay-based methods [Rebuffi *et al.*, 2017; Zheng *et al.*, 2024], regularization-based methods [Zenke *et al.*, 2017; Szatkowski *et al.*, 2023], architecture-based methods [Yoon *et al.*, 2018; Marouf *et al.*, 2024] and so on [Marczak *et al.*, 2024].

Traditional IL methods [Rebuffi *et al.*, 2017; Zhao *et al.*, 2020] are built on the assumption that data is accurately annotated. However, real-world data often has label ambiguity, making precise annotations labor-intensive, especially in sequential learning. For instance, the Alaskan Malamute can be visually similar to the Siberian Husky, hindering non-experts from accurately identifying the true breeds. Recently, this ambiguous supervision problem has attracted great attention from the community [Lyu *et al.*, 2022; Lv *et al.*, 2023; Yan and Guo, 2023; Jia *et al.*, 2024].

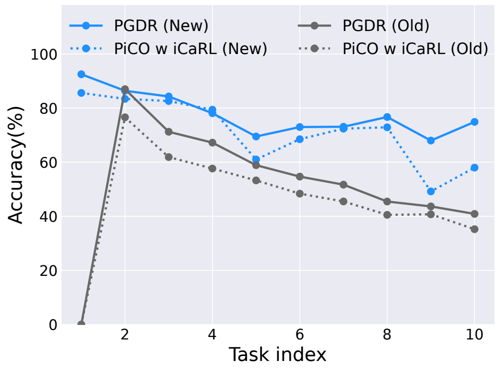
To reduce annotation costs, we study a novel weakly supervised learning framework dubbed *incremental partial label learning* (IPLL), where (i)-data is given sequentially as a stream; (ii)-each task contains a vast number of new class samples while potentially carries old class data, and (iii)-each sample is associated with a candidate label set instead of the ground truth; see Figure 1. Notably, since the annotator may confuse previous experience with the true label of the sample, our IPLL setup allows the candidate label set of the sample to include both new and old classes, while ensuring the inclusion of the true one. Arguably, the IPLL problem is deemed more practical in real-world scenarios due to its relatively lower cost to annotations.

The key to successful learning from ambiguous supervision is label disambiguation, i.e., identifying the true labels from the candidate sets. To achieve this, existing par-

[*]Corresponding author.



(a) Class prior estimation.



(b) Per-Task accuracy.

Figure 2: (a) Class distribution based on both predicted labels and ground-truth labels. The real/estimated distribution of old classes (0∼9) and new classes (10∼19) in the early stage of the second task of the CIFAR100 (10 tasks). Combining the PLL baseline (PiCO) and the IL baseline (iCaRL) leads to **classification bias**. (b) Comparison of our PGDR and PiCO with iCaRL in IPLL. It displays the average accuracy of new and old classes for each task in CIFAR100 where PiCO with iCaRL demonstrates inferior and unstable performance compared to our method.

tial label learning (PLL) algorithms [Wang *et al.*, 2022c; Yan and Guo, 2023; Li *et al.*, 2023; Jia *et al.*, 2024; Liu *et al.*, 2024] mostly rely on the self-training algorithm that assigns pseudo-labels using the model predictions. However, such a strategy can be problematic in our IPLL framework. On the one hand, the dynamically changing environment leads to the forgetting of past knowledge, which amplifies knowledge confusion that **hinders the disambiguation procedure**. On the other hand, ambiguous supervision significantly undermines the ability of DNNs to acquire meaningful knowledge from samples, **exacerbating catastrophic forgetting**. As exemplified in our empirical studies, directly combining the PLL method with the IL method also leads to inferior results. Specifically, when combining the most popular PLL method PiCO [Wang *et al.*, 2022c] with one classic IL method iCaRL [Rebuffi *et al.*, 2017], the model overly emphasizes old classes in the early stages of each task (see Figure 2(a)), thus exacerbating the confusion of old class knowledge (see Figure 2(b)). In the more challenging Tiny-ImageNet, the negative impacts of classification bias become more pronounced (see Figure 4(b)). This gives rise to the fundamental issue in IPLL—*how to balance the model's perception of new and old knowledge*.

To address this issue, we propose the **P**rototype-**G**uided **D**isambiguation and **R**eplay Algorithm (PGDR), which leverages class prototypes carrying rich category information as proxies to balance the model's perception of classes. Concretely, to alleviate label confusion, we propose a *prototype-guided label disambiguation* strategy, which firstly performs distance-based old/new sample separation and then, momentumly updates on pseudo-labels. Secondly, to mitigate catastrophic forgetting, we construct an episodic memory by distance-based sampling for representative samples and region-aware sampling for diverse samples, which are replayed by a knowledge distillation loss. These two modules mutually benefit each other to achieve a balanced perception on all classes—the disambiguation module ensures accurate training on new classes while the memory replay module strengthens the old classes. We conduct comprehensive experiments on benchmark datasets to show that PGDR establishes state-of-the-art performance. In IPLL, our method outperforms the best baseline by **6.05%** and **11.60%** on the CIFAR100 and Tiny-ImageNet.

2 Related Work

Incremental Learning. The data appear in a sequence and the model continuously learns novel knowledge while maintaining the discrimination ability for previous knowledge. There is a classical incremental learning variant—blurry incremental learning [Bang *et al.*, 2021; Moon *et al.*, 2023], where different stages exhibit distinct data distributions. Due to the intersection of the label space, it faces heightened ambiguity. Nonetheless, IL and its variants are confronted with the challenge of knowledge forgetting [Kim *et al.*, 2024; Li *et al.*, 2024]. To address this issue, existing methods can be categorized into three major strategies. For architecture-based methods [Yoon *et al.*, 2020; Marouf *et al.*, 2024], different model parameters are allocated for each task. The regularization-based methods [Bian *et al.*, 2024] introduce regularization terms into the objective function. The replay-based methods [Bhat *et al.*, 2023; Yoo *et al.*, 2024] retain a subset of historical samples and incorporate them into subsequent tasks. Moreover, while some works [Lange and Tuytelaars, 2021; Asadi *et al.*, 2023] employ prototypes, they mainly focus on using them to alleviate knowledge forgetting and do not investigate their effectiveness in other aspects, distinguishing our approach from theirs.

The IL variants discussed above all assume accurately labeled data. More and more researchers are increasingly interested in incremental learning in limited or unsupervised settings, such as unsupervised continual learning [Cha *et al.*, 2024], semi-supervised continual learning [Fan *et al.*, 2024], and noisy labeled continual learning [Bang *et al.*, 2022]. However, most of the work overlooks label ambiguity. Although [Yu *et al.*, 2024] attempts to address this issue, the subsequent samples are unlabeled, making it similar to a semi-supervised learning problem.

Partial-Label Learning. The fundamental challenge in PLL is label disambiguation, requiring the model to select the true label from the set of candidate labels [Lyu *et al.*, 2021;

Wang *et al.*, 2022a; Bao *et al.*, 2024]. Some PLL methods enhance candidate label disambiguation through adversarial learning [Zhang *et al.*, 2020; Zhang *et al.*, 2023], and the majority of work [Wang *et al.*, 2022c; Tian *et al.*, 2024] is geared towards devising appropriate objectives for PLL. The strategy [Lv *et al.*, 2023] based on averaging assigns equal weights to the labels in the candidate labels, and then obtains predictions by averaging the output. The disambiguation strategy based on identification [Yu and Zhang, 2017; Tian *et al.*, 2024] treats the true labels of samples as latent variables and iteratively optimizes the objective function of these latent variables. Based on self-training disambiguation strategies [He *et al.*, 2022; Yan and Guo, 2023; Dong *et al.*, 2023; Liu *et al.*, 2024], pseudo-labels are used as the model's supervisory information for training, e.g., PRODEN [Lv *et al.*, 2020] re-normalizes the classifier's output and PiCO [Wang *et al.*, 2022c] introduces contrastive learning. Despite the promise, existing PLL work mostly assumes a static data distribution, which is typically not available in practice. Therefore, we introduce incremental learning to investigate IPLL that is more suitable for real-world scenarios.

3 Background

3.1 Problem Definition

The goal of IPLL is to sequentially learn a unified model from ambiguous supervised datasets and classify unseen test samples of all classes that have been learned so far. Formally, assume we are given a stream of datasets $\{\mathcal{D}_t\}_{t=1}^{T}$, where each subset $\mathcal{D}_t = \{(\boldsymbol{x}_i^t, \mathcal{S}_i^t)\}_{i=1}^{N_t}$ contains N_t samples. Here, $\boldsymbol{x}_i^t \in \mathcal{X}$ represents a sample in the input space $\mathbb{R}^d$. Different from the supervised setup where the ground truth y_i is known, we follow the setup of previous PLL studies [Feng *et al.*, 2020; Lv *et al.*, 2020; Wang *et al.*, 2022c] and allow the annotator to assign a rough candidate label set $\mathcal{S}_i^t \subset \mathcal{Y}_t$ containing the true label, i.e., $y_i^t \in \mathcal{S}_i^t$. For the label space, we consider a *blurry* incremental learning [Bang *et al.*, 2021] that can be ubiquitous in real-world applications[1], whose data distribution demonstrates: (i)-a majority of samples with new labels emerge incrementally $\mathcal{Y}_t = \mathcal{Y}_{t-1} \cup \mathcal{Y}_t^{\text{new}}$; (ii)-each subset $\mathcal{D}_t$ potentially contains data samples from all labels in $\mathcal{Y}_t$. At step t, the goal of IPLL is to train a model f from the new dataset $\mathcal{D}_t$ without interfering with previous data, where f consists of the feature extractor backbone ϕ with a fully-connected layer upon it. During training, since the ground-truth label is not accessible, we assign each sample $\boldsymbol{x}_i$ a pseudo-label vector $\boldsymbol{p}_i$ and update the model by cross-entropy loss:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{j=1}^{|\mathcal{Y}_t|} p_{ij}\log(f_j(\boldsymbol{x}_i)) \tag{1}$$

In the remainder of this work, we omit the task index t when the context is clear.

[1]The classic IL setup simply assumes the label spaces have no intersection, which is less practical and the label ambiguity issue is typically not significant.

3.2 Prototype Generation

Recall the crucial challenge of IPLL is to balance the perception of new and old classes. We observe that while individual samples exhibit strong ambiguity, prototypes derived from sample aggregation can serve as stable and non-parameterized proxies to guide the model in *identifying new patterns* and *memorizing old tasks*. Formally, at the end of the t-th training task, we generate prototypes for class $c \in \mathcal{Y}_t$ by feature averaging $\boldsymbol{\mu}_c = \frac{1}{|\boldsymbol{P}_c|}\sum \boldsymbol{P}_c$, where $\boldsymbol{P}_c = \{\phi(\boldsymbol{x}_i) | c = \arg\max_{j\in\mathcal{S}_i^t} f_j(\boldsymbol{x}_i)\}$ represents the feature set of samples whose classifier prediction is class c. In later rounds, holding the belief that the model always produces accurate predictions on old classes, we momentum update the prototypes during the training procedure:

$$\boldsymbol{\mu}_c = \gamma\boldsymbol{\mu}_c + (1-\gamma)\frac{1}{|\boldsymbol{P}_c|}\sum \boldsymbol{P}_c, \tag{2}$$

where $\gamma > 0$ is a hyperparameter. In what follows, we elaborate on how prototypes help improve both disambiguation and memorization ability.

4 Proposed Method

In this section, we describe our novel **P**rototype-**G**uided **D**isambiguation and **R**eplay algorithm (PGDR) in detail. Overall, it consists of two components: (i)-a *prototype-guided label disambiguation* module that first performs old/new sample separation according to their distance to the prototypes, allowing the subsequent task-aware pseudo-labeling; (ii)-a *memory replay* module that constructs a sample pool containing both diverse and representative samples. The overall training scheme of PGDR is outlined in Figure 3.

4.1 Prototype-Guided Label Disambiguation

To handle the label ambiguity, the most seminal PLL algorithms [Lv *et al.*, 2020] adopt a self-training paradigm that elicits pseudo-labels from the model outputs. However, in the IPLL setup, such a strategy can be problematic since the classifier can be largely biased after the previous rounds of training. Thus, the disambiguation process would be disrupted since even new class samples can receive very confident predictions on old classes.

Old/New Data Separation. To address the bias, we introduce a prerequisite step that separates samples into distinct subsets for old and new classes. Specifically, we first construct a distance measure set of those samples containing at least one old class,

$$\mathcal{A} = \{e_i = \min_{j\in\mathcal{S}_i^t\cap\mathcal{Y}_{t-1}} ||\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_j||_2, \text{ if } \mathcal{S}_i^t \cap \mathcal{Y}_{t-1} \neq \varnothing\}. \tag{3}$$

Our intuition is that, the prototypes fully condense the knowledge of old classes, and thus, samples from old classes are located closer to their prototypes than new ones. To achieve fully automated separation, we draw inspiration from the noisy label learning literature [Li *et al.*, 2020] to fit a two-component Gaussian Mixture Model (GMM) on $\mathcal{A}$. Let $w_i = p(g|e_i)$ represent the probability of $\boldsymbol{x}_i$ belonging to the Gaussian component with smaller mean g, which can also be

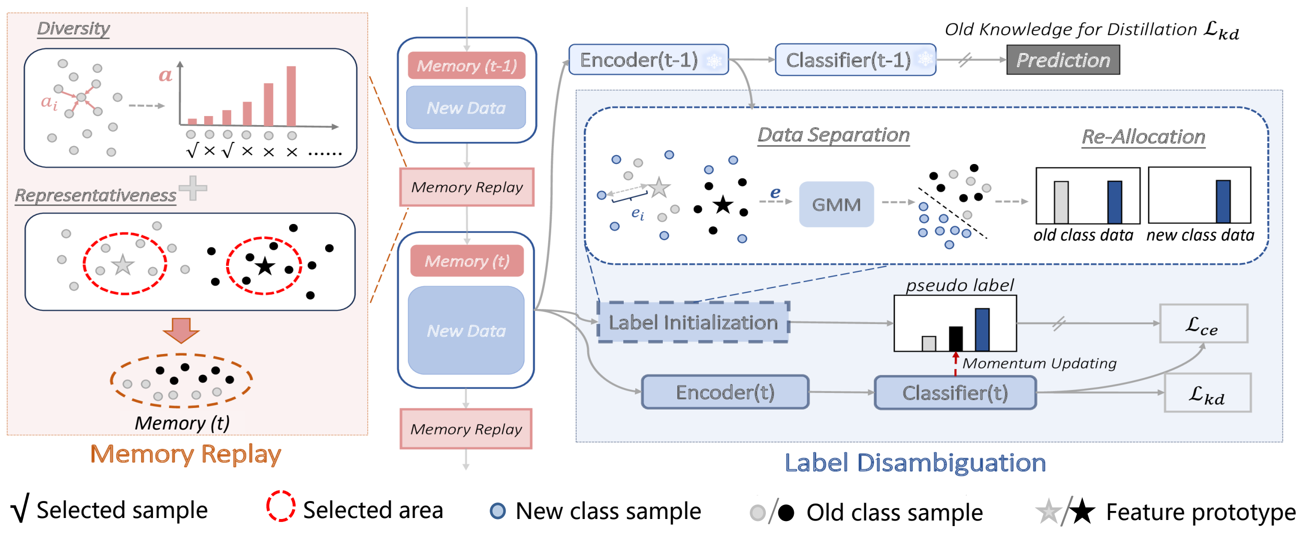


Figure 3: Overall framework of PGDR. The label disambiguation module divides new and old class samples based on prototypes and assigns different labels. PGDR then combines the momentum-updated pseudo-labels to achieve label disambiguation. After completing the task, the memory replay module is utilized to filter samples for subsequent training to mitigate forgetting.

deemed as its old task probability. Then, we collect those samples whose w_i exceeds the threshold α as "*potentially*" old class data. The remaining samples belong to the new class sample set $\mathcal{D}_t^{\text{new}}$, i.e., $\mathcal{D}_t^{\text{new}} = \mathcal{D}_t \setminus \mathcal{D}_t^{\text{old}}$.

Candidate Label Re-Allocation. We re-allocate the candidate label sets according to our data separation procedure,

$$\mathcal{S}_i' = \begin{cases} \{c_i^*\} \cup \mathcal{Y}_t^{\text{new}} \cap \mathcal{S}_i^t & \text{if } \boldsymbol{x}_i \in \mathcal{D}_t^{\text{old}}, \\ \mathcal{Y}_t^{\text{new}} \cap \mathcal{S}_i^t & \text{otherwise,} \end{cases} \tag{4}$$

where c_i^* is a class that corresponds to the nearest prototype. Our rationale is that: (i)-for a sample is far away from the prototypes (i.e. from $\mathcal{D}_t^{\text{new}}$), it is highly probable to be a new class sample, and thus, we discard all its old candidate labels; (ii)-for those identified as from old classes, it is either an old class sample locates around one old class prototype (i.e. prototype of c_i^*) or actually a new class sample. Hence, we employ the old class prototypes as a disambiguator to remove confusing old classes, but preserve all candidates on new classes. After this process, the model bias turns into a *good property*—it allows pre-disambiguate old classes. Thus, the model can focus on addressing the ambiguity among new classes. Notably, we conduct the separation procedure once before training for each task to achieve differential guidance for the model.

Momentum-based Pseudo-labeling. After that, we initialize the pseudo-labels $\boldsymbol{p}_i$ by a uniform probability on the newly allocated candidate label set:

$$p_{ij} = \frac{1}{|\mathcal{S}_i'|}\mathbb{I}(j \in \mathcal{S}_i'). \tag{5}$$

In subsequent rounds, we assume the classifier can be increasingly accurate. Thus, we devise a momentum-based mechanism to update the $\boldsymbol{p}_i$:

$$\boldsymbol{p}_i \leftarrow \beta \boldsymbol{p}_i + (1-\beta)\boldsymbol{z}_i, \tag{6}$$

where $\boldsymbol{z}_i$ is a one-hot vector, i.e., $z_{ij} = \mathbb{I}(j = \arg\max_{j \in \mathcal{S}_i^t} f_j(\boldsymbol{x}_i))$, and β is a hyperparameter. Practically, we assign a large β at the beginning of each task due to the unreliable model prediction of new classes. As the prediction on new classes becomes more accurate, we set a smaller β value to facilitate the convergence of the pseudo-labels.

4.2 Representative and Diverse Memory Replay

While our disambiguation module prevents the model from label confusion, the current task is still dominated by new class samples. As the pseudo-labels become increasingly precise, the model tends to be reversely biased towards new patterns and quickly forgets the old knowledge. Without stabilizing the old knowledge, the training procedure may be fairly unstable, leading to degraded performance. To this end, we further introduce a memory replay module to alleviate forgetting, which comprises representative and diverse samples.

Distance-based Representativeness. We assume that most samples can be purified by our disambiguation module at the end of each task. Accordingly, we believe those samples around the prototypes are accurate and can represent well the whole clusters. Formally, denote the whole memory by $\mathcal{M}^t$ at the t-th round, with a maximum limit of m for storing samples. We first concatenate the training set with the previous memory by $\mathcal{D}_t' = \mathcal{D}_t \cup \mathcal{M}^{t-1}$. Then, we refer to prior research [Rebuffi *et al.*, 2017] and perform per-class distance-based selection that collects samples having the shortest distance to their prototype:

$$\begin{aligned} \mathcal{M}_r^t = \cup_{c \in \mathcal{Y}_t} \{\boldsymbol{x}_i | c = \arg\max_{j \in \mathcal{S}_i^t} f_j(\boldsymbol{x}_i), \\ \text{rank}(d_i^c) \le N_r, \text{and } \boldsymbol{x}_i \in \mathcal{D}_t'\}, \end{aligned} \tag{7}$$

where $d_i^c = ||\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_c||_2$. That is, we split the whole set according to the predicted categories and then select N_r most representative samples to the episodic memory.

Neighborhood-based Diversity. Apart from those representative samples, we further collect a few samples that can better preserve the diversity of old class distribution. In our IPLL setup, this should be done more carefully due to the label ambiguity. We propose selecting a sample with two characteristics. First, it should lie in a smooth local manifold to avoid including falsely disambiguated samples. Second, it does not lie in a local region near those already selected samples and carries mutually different patterns. Formally speaking, we calculate the sum of distance measure from a sample to its K-nearest neighbor:

$$a_i = \sum_{j \in \mathcal{N}_K(\boldsymbol{x}_i)} d_{ij}, \tag{8}$$

where $\mathcal{N}_K(\boldsymbol{x}_i) = \{\boldsymbol{x}_j | \text{rank}(d_{ij}) \leq K, j \neq i\}$. In other words, we believe those samples with lower a_i, which indicates they have many close neighbors, are typically located in a relatively smooth region.

Next, we select a sample with the smallest a_i while being not a part of the already chosen sample's neighbors:

$$\mathcal{M}_k^t = \mathcal{M}_k^t \cup \{\boldsymbol{x}_i | a_i = \min_{\boldsymbol{x}' \in \mathcal{D}_t'} a', \text{and } \boldsymbol{x}_i \notin \cup_{\boldsymbol{x}' \in \mathcal{M}_k^t} \mathcal{N}_K(\boldsymbol{x}')\}. \tag{9}$$

In our implementation, for each class, we first select up to N_d diverse samples and then, select $N_r = m/|\mathcal{Y}_t| - N_d$ representative samples, resulting in $\mathcal{M}^t = \mathcal{M}_r^t \cup \mathcal{M}_k^t$. This provides richer class information for subsequent learning, mitigating the forgetting of old classes. $\mathcal{L}_{\text{ce}}$ also includes this portion of samples

Knowledge Distillation Regularization. In the training phase, we utilize replay data $\mathcal{M}^{t-1}$ and the current data $\mathcal{D}_t$ for training. Following [Rebuffi *et al.*, 2017], we employ a knowledge distillation loss to alleviate forgetting,

$$\mathcal{L}_{\text{kd}} = -\frac{1}{|\mathcal{D}_t'|} \sum_{i=1}^{|\mathcal{D}_t'|} \sum_{j=1}^{|\mathcal{Y}_{t-1}|} f_j^{old}(\boldsymbol{x}_i) \log(f_j(\boldsymbol{x}_i)). \tag{10}$$

4.3 Practical Implementation

Robust Training with Self-Supervised Learning. In order to further enhance classification performance, we introduce self-supervised learning on the foundation of the disambiguation module and memory replay module. Specifically, we introduce consistency regularization $\mathcal{L}_{\text{cr}}$ [Berthelot *et al.*, 2019] to encourage smoother decision boundaries (details in Appendix A.1). Finally, the total loss is defined as,

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{kd}} + \mathcal{L}_{\text{cr}}. \tag{11}$$

Bias Elimination for the Testing Phase. In each task, despite having operations to mitigate forgetting, the influence of new class samples on model updates is more profound due to the significantly larger number of new class samples compared to old class samples. Therefore, the classifier exhibits a certain bias. In contrast, the feature prototypes retain rich old-class knowledge and are not substantially updated as training progresses. Consequently, we employ the feature prototypes for sample classification during testing,

$$y_i^* \leftarrow \arg\min_{j \in \mathcal{Y}_t} \|\phi(\boldsymbol{x}_i) - \boldsymbol{\mu}_j\|. \tag{12}$$

As demonstrated empirically, the feature prototype classifier does indeed outperform linear classification during the testing phase; see Appendix B.7.

5 Experiments

5.1 Experimental Settings

Datasets. We perform experiments on CIFAR100 [Krizhevsky *et al.*, 2009] and Tiny-ImageNet [Le and Yang, 2015]. Additionally, we further conduct experiments on CUB200 [Welinder *et al.*, 2010]. For the experimental setup of IPLL, we reference the settings of PLL [Lv *et al.*, 2020] and IL [Li and Hoiem, 2016; Zhao *et al.*, 2020]. We generate partially labeled datasets by manually flipping negative labels $\bar{y} \neq y$ to false-positive labels with probability $q = P(\bar{y} \in \mathcal{Y}_t | \bar{y} \neq y)$. In the t-th task, all $|\mathcal{Y}_t| - 1$ negative labels have a uniform probability to be false positive and we aggregate the flipped ones with the ground-truth to form the candidate set $\mathcal{S}_i^t$. The flipping probability q is set at 0.1 and 0.2. We randomly partition all classes into 10 tasks, i.e., $T = 10$. In the first task, there are exclusively new classes, while in the remaining $T - 1$ tasks, a substantial number of samples belong to new classes, with relatively fewer samples belonging to old classes. To be specific, for a certain category of samples, $W\%$ of the samples emerge as new class samples in the t-th task, while the remaining $(100 - W)\%$ of the samples uniformly appear as old class samples in each subsequent task $\{t + 1, t + 2, ..., T\}$. We consider the degree of blending new and old class data to be "(100-W)-blurry" and set the value of W to 90 (10-blurry) and 70 (30-blurry).

Baselines. We compare PGDR with two PLL methods: **PiCO** [Wang *et al.*, 2022c] and **PaPi** [Xia *et al.*, 2023]. We also discuss **PRODEN** [Lv *et al.*, 2020] in Appendix B.4. We compare PGDR with five incremental learning methods: 1) **iCaRL** [Rebuffi *et al.*, 2017]; 2) **BiC** [Wu *et al.*, 2019]; 3) **WA** [Zhao *et al.*, 2020]; 4) **ER-ACE** [Caccia *et al.*, 2022]; 5) **ANCL** [Kim *et al.*, 2023]. To ensure experimental fairness and credibility, we combine these five incremental learning methods with PiCO and PaPi to achieve basic label disambiguation. To objectively evaluate our method, we respectively plug the contrastive learning module of PiCO and the Kullback-Leibler divergence of PaPi into our method.

Evaluation metrics. We utilize the average incremental accuracy as the performance evaluation metric [Rebuffi *et al.*, 2017]. At the t-th task, the incremental accuracy represents the classification accuracy A_t of the model on the currently seen classes. The average incremental accuracy is denoted as $\bar{A} = (1/T) \sum_{i=1}^{T} A_i$.

Implementation Details. We employ ResNet-18 for feature extraction. The network model is trained using SGD with a momentum of 0.9. Trained for 200 epochs on the CIFAR100, and trained for 300 epochs on the Tiny-ImageNet. The learning rate starts at 0.1 for PiCO and 0.01 for PaPi. Batch sizes are set to 256 and 128 for the CIFAR100 and Tiny-ImageNet, with a maximum sample storage limit of m of 2000. For prototypes, a moving average coefficient γ of 0.5 is used. In the sample selection for the storage phase, we

Method	CIFAR100				Tiny-ImageNet			
	$q = 0.1$		$q = 0.2$		$q = 0.1$		$q = 0.2$	
	10-blurry	30-blurry	10-blurry	30-blurry	10-blurry	30-blurry	10-blurry	30-blurry
PiCO	28.00	28.90	18.66	19.34	22.62	19.02	10.85	9.96
+iCaRL	61.59	62.75	59.40	55.25	41.81	43.87	35.27	13.98
+BiC	64.11	64.78	53.17	52.93	33.43	34.71	25.34	23.41
+WA	67.31	66.86	64.12	59.93	44.15	44.33	33.80	31.37
+ER-ACE	62.98	65.32	61.32	61.29	43.14	43.01	40.83	40.79
+ANCL	65.66	66.10	56.04	48.81	35.11	30.31	23.53	22.59
+PGDR	**69.84**	**71.68**	**68.49**	**70.85**	**46.47**	**49.33**	**42.03**	**41.79**
PaPi	24.17	31.16	21.59	22.47	20.45	20.17	16.34	10.84
+iCaRL	60.11	62.52	57.69	58.08	38.89	37.46	22.42	20.46
+BiC	61.50	63.08	58.40	56.69	41.09	35.63	25.67	22.82
+WA	62.75	63.59	59.73	58.10	39.34	35.82	20.91	18.60
+ER-ACE	60.76	62.13	50.43	47.17	23.60	21.45	14.21	14.10
+ANCL	63.19	63.47	60.89	57.85	42.41	35.33	22.15	19.26
+PGDR	**69.09**	**69.64**	**68.47**	**66.84**	**47.85**	**49.06**	**44.49**	**44.64**

Table 1: Accuracy comparisons on CIFAR100 and Tiny-ImageNet. The best results are marked in bold and the second-best marked in underline. q represents the degree of label ambiguity. IL methods are equipped with the PLL method PiCO and PaPi.

Method	CIFAR100-H				CUB200			
	$q = 0.1$		$q = 0.2$		$q = 0.1$		$q = 0.2$	
	10-blurry	30-blurry	10-blurry	30-blurry	10-blurry	30-blurry	10-blurry	30-blurry
PiCO	22.85	26.50	19.41	18.16	21.12	22.12	15.25	13.08
+iCaRL	51.69	47.28	51.97	42.51	46.16	41.14	28.58	22.37
+BiC	52.75	53.74	36.65	37.32	43.67	41.28	28.11	22.49
+WA	53.25	54.50	49.26	46.53	44.84	43.02	30.40	24.14
+ER-ACE	54.22	54.99	51.75	52.00	39.09	39.48	23.83	23.11
+ANCL	55.59	55.73	38.73	37.37	44.70	42.18	31.16	24.86
+PGDR	**60.35**	**59.78**	**56.85**	**54.93**	**48.83**	**49.26**	**36.11**	**34.04**
PaPi	18.25	26.06	15.87	17.13	20.82	21.87	16.40	14.09
+iCaRL	47.06	48.71	38.37	38.93	44.61	38.81	29.54	23.62
+BiC	49.14	49.01	41.88	38.79	44.70	40.60	33.04	26.47
+WA	49.15	47.03	38.82	37.41	45.57	41.78	32.29	25.74
+ER-ACE	47.27	44.19	28.41	24.80	43.23	37.59	25.97	22.22
+ANCL	51.73	51.09	42.56	37.89	45.27	42.56	33.50	25.45
+PGDR	**58.70**	**58.50**	**56.75**	**55.46**	**46.30**	**47.06**	**33.72**	**27.19**

Table 2: Accuracy comparisons on CIFAR100-H and CUB200. The best results are marked in bold and the second-best marked in underline. q represents the degree of label ambiguity. IL methods are equipped with the PLL method PiCO and PaPi.

set the number of nearest neighbors K to 10, with a maximum limit N_d for diverse sample storage of $0.67 * m/|\mathcal{Y}_t|$. In the label disambiguation stage, the threshold α is set to 0.8. We linearly ramp down β from 0.8 to 0.6 to ensure the full utilization of differential labels in the early stages of training. This helps mitigate the introduction of noisy information resulting from inaccurate early predictions. The experiments regarding PiCO and PaPi reference their experimental parameters.

5.2 Comparative Results

Comparison under IPLL. As shown in Table 1, our method significantly outperforms baselines in the IPLL. The PLL methods, PiCO and PaPi, are lower than other solutions due to their difficulty in mitigating forgetting. In addition, on the CIFAR100 dataset with 10 tasks and $q = 0.1$, our method outperforms the best baseline incorporating PiCO by **2.53%** (10-blurry) and **4.82%** (30-blurry). Additionally, compared to the best baseline using PaPi, our method shows improvements of **5.90%** (10-blurry) and **6.05%** (30-blurry). Furthermore, in the case of more noisy candidate labels with $q = 0.2$, our method achieves a maximum performance improvement of **9.56%**. We also validate the effectiveness of our method on the more challenging Tiny-ImageNet. Our method still demonstrates significant advantages, providing evidence of the superiority of our method in addressing disambiguation and mitigating forgetting.

Comparison on fine-grained datasets. When semantically similar classes are concentrated in a particular stage, disam-

Ablation	Disambiguation	Memory Replay	10-blurry	30-blurry
PGDR	✓	✓	**63.83**	**64.58**
PGDR w MP	Model Prediction	✓	57.94	59.13
PGDR w PP	Prototype Prediction	✓	59.92	62.60
PGDR w/o Memory	✓	×	23.18	23.94
PGDR w Random	✓	Random	61.34	62.60
PGDR w Distance	✓	Prototype Distance	61.69	62.52

Table 3: Ablation study of PGDR on CIFAR100 with $q = 0.2$ at IPLL 10 tasks.

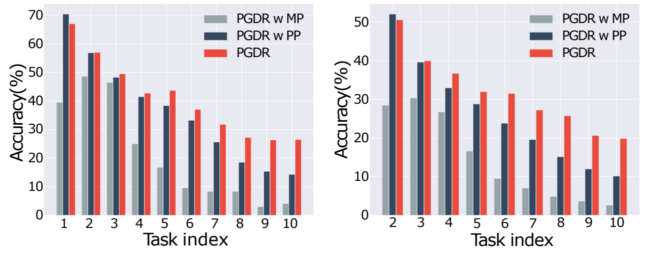
biguation becomes more challenging. To address this, we reference [Wang *et al.*, 2022c; Wang *et al.*, 2022b] and evaluate our method on two fine-grained datasets: 1) CIFAR100 dataset with hierarchical labels, i.e., CIFAR100-H. There are five categories under each superclass, and there is no intersection between different superclasses; 2) CUB200 [Welinder *et al.*, 2010] dataset with 200 species of birds. For CIFAR100-H, the new classes in each task are derived from two unseen superclasses in IPLL. The flipping probabilities q are 0.1 and 0.2. As shown in Table 2, all solutions exhibited a noticeable decrease in performance in the more challenging setting. On the CUB200 dataset, when q is set to 0.1, the gaps between our method and the best baseline that incorporates PiCO are **2.67%** (10-blurry) and **6.24%** (30-blurry). This thoroughly validates the efficacy of our method, even under highly challenging data settings.

5.3 Ablation Study

The additional introduction of the loss function in PGDR hinders visualizing the effects of each module, reducing the interpretability of ablation results. Thus, we present ablation results on PGDR without contrastive learning and Kullback-Leibler divergence to demonstrate our method's effectiveness. More experiments can be found in Appendix B.7.

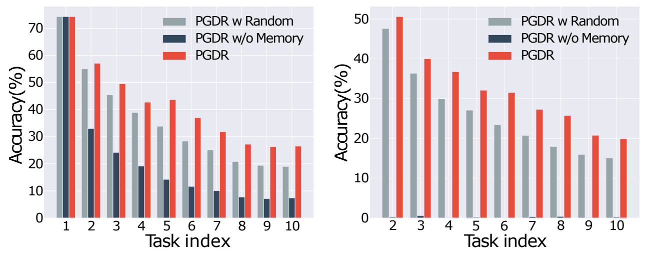
Effect of Label Disambiguation. To validate the effectiveness of the label disambiguation module, we choose the variants 1) *PGDR w MP* and 2) *PGDR w PP*; see Appendix A.2. The first variant utilizes the PRODEN label computation method. The second variant utilizes PiCO's labeling method. The Table 3 shows that our disambiguation solution is significantly better than the two variants. To explore further, we visualized category changes under the Tiny-ImageNet 10-blurry (Figure 4). *PGDR w PP* frequent updates to prototypes accumulate model bias, leading to a noticeable decline in classifications when new class data is introduced. In comparison, our solution alleviates knowledge confusion, balancing the model's perception of categories, especially in old classes.

Different Memory Replay Strategies. We further try different memory update strategies to validate the superiority of our sample memory replay strategy. We compare PGDR with three variants: 1) *PGDR w/o Memory* that does not store samples; 2) *PGDR w Random* randomly selects some samples for storage; 3) *PGDR w Distance* entails selecting only representative samples. As shown in Table 3, our method outperforms all the variants, which demonstrates the effectiveness of our representative and diverse selection procedure. We detail the comparison results of each task in Figure 5, where our method outperforms other variants on all-class average accuracy. We



(a) All-class average accuracy. (b) Old-class average accuracy.

Figure 4: Ablation experiment about disambiguation module on Tiny-ImageNet (q=0.2)



(a) All-class average accuracy. (b) Old-class average accuracy.

Figure 5: Ablation experiment about memory replay module on Tiny-ImageNet (q=0.2).

can also observe that *PGDR w Random* struggles to overcome forgetting old classes due to the lack of high-quality replay data, and *PGDR w/o Memory* suffers from close to zero old-class accuracies with no memory replay.

6 Conclusion

We propose and investigate a novel scenario called *incremental partial label learning* (IPLL), where data appears sequentially, and each sample is associated with a set of candidate labels. This presents even more challenging obstacles, involving the intertwining of label disambiguation and catastrophic forgetting. To address this, we propose a prototype-guided disambiguation and replay algorithm (PGDR), which utilizes class prototypes as proxies to balance the model's perception of different categories. Moreover, we establish the first IPLL benchmark to lay the foundation for future research. Through extensive experiments, our method consistently outperformed baselines, demonstrating the superiority of PGDR. We hope our work will draw more attention from the community towards a broader view of tackling the IPLL.

Acknowledgements

This paper is supported by NSFC under Grants (No. 62402424). Chang Yao is partially supported by the Key Research and Development Program of Zhejiang Province No. 2024C03270.

References

[Asadi *et al.*, 2023] Nader Asadi, MohammadReza Davari, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Prototype-sample relation distillation: Towards replay-free continual learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*. PMLR, 2023.

[Bang *et al.*, 2021] Jihwan Bang, Heesu Kim, Youngjoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proc. of CVPR*, 2021.

[Bang *et al.*, 2022] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with blurry task boundaries. In *Proc. of CVPR*, 2022.

[Bao *et al.*, 2024] Wei-Xuan Bao, Yong Rui, and Min-Ling Zhang. Disentangled partial label learning. In *Proc. of AAAI*, 2024.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Proc. of NeurIPS*, 2019.

[Bhat *et al.*, 2023] Prashant Shivaram Bhat, Bahram Zonooz, and Elahe Arani. Task-aware information routing from common representation space in lifelong learning. In *Proc. of ICLR*, 2023.

[Bian *et al.*, 2024] Ang Bian, Wei Li, Hangjie Yuan, Chengrong Yu, Zixiang Zhao, Mang Wang, Aojun Lu, and Tao Feng. Make continual learning stronger via c-flat. In *Proc. of NeurIPS*, 2024.

[Caccia *et al.*, 2022] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *Proc. of ICLR*, 2022.

[Cha *et al.*, 2024] Sungmin Cha, Kyunghyun Cho, and Taesup Moon. Regularizing with pseudo-negatives for continual self-supervised learning. In *Proc. of ICML*, 2024.

[Cichon and Gan, 2015] Joseph Cichon and Wen-Biao Gan. Branch-specific dendritic ca2+ spikes cause persistent synaptic plasticity. *Nature*, 2015.

[Dong *et al.*, 2023] Ruo-Jing Dong, Jun-Yi Hang, Tong Wei, and Min-Ling Zhang. Can label-specific features help partial-label learning? In *Proc. of AAAI*, 2023.

[Fan *et al.*, 2024] Yan Fan, Yu Wang, Chen Dongyue Zhu, Pengfei, and Qinghua Hu. Persistence homology distillation for semi-supervised continual learning. In *Proc. of NeurIPS*, 2024.

[Feng *et al.*, 2020] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Proc. of NeurIPS*, 2020.

[He *et al.*, 2022] Shuo He, Lei Feng, Fengmao Lv, Wen Li, and Guowu Yang. Partial label learning with semantic label representations. In *Proc. of KDD*, 2022.

[Jia *et al.*, 2024] Yuheng Jia, Xiaorui Peng, Ran Wang, and Min-Ling Zhang. Long-tailed partial label learning by head classifier and tail classifier cooperation. In *Proc. of AAAI*, 2024.

[Kim *et al.*, 2023] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning. In *Proc. of CVPR*, 2023.

[Kim *et al.*, 2024] Taehoon Kim, Jaeyoo Park, and Bohyung Han. Cross-class feature augmentation for class incremental learning. In *Proc. of AAAI*, 2024.

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

[Lange and Tuytelaars, 2021] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021.

[Le and Yang, 2015] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015.

[Li and Hoiem, 2016] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Proc. of ECCV*, 2016.

[Li *et al.*, 2020] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proc. of ICLR*, 2020.

[Li *et al.*, 2023] Ximing Li, Yuanzhi Jiang, Changchun Li, Yiyuan Wang, and Jihong Ouyang. Learning with partial labels from semi-supervised perspective. In *Proc. of AAAI*, 2023.

[Li *et al.*, 2024] Depeng Li, Tianqi Wang, Junwei Chen, Qining Ren, Kenji Kawaguchi, and Zhigang Zeng. Towards continual learning desiderata via hsic-bottleneck orthogonalization and equiangular embedding. In *Proc. of AAAI*, 2024.

[Liu *et al.*, 2024] Yangfan Liu, Jiaqi Lv, Xin Geng, and Ning Xu. Learning with partial-label and unlabeled data: A uniform treatment for supervision redundancy and insufficiency. In *Proc. of ICML*, 2024.

[Lv *et al.*, 2020] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proc. of ICML*, 2020.

[Lv *et al.*, 2023] Jiaqi Lv, Biao Liu, Lei Feng, Ning Xu, Miao Xu, Bo An, Gang Niu, Xin Geng, and Masashi

Sugiyama. On the robustness of average losses for partial-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[Lyu *et al.*, 2021] Gengyu Lyu, Songhe Feng, Tao Wang, Congyan Lang, and Yidong Li. GM-PLL: graph matching based partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Lyu *et al.*, 2022] Gengyu Lyu, Yanan Wu, and Songhe Feng. Deep graph matching for partial label learning. In *Proc. of IJCAI*, 2022.

[Marczak *et al.*, 2024] Daniel Marczak, Bartlomiej Twardowski, Tomasz Trzcinski, and Sebastian Cygert. MAGMAX: leveraging model merging for seamless continual learning. In *Proc. of ECCV*, 2024.

[Marouf *et al.*, 2024] Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. In *Proc. of ECCV*, 2024.

[Moon *et al.*, 2023] Jun-Yeong Moon, Keon-Hee Park, Jung Uk Kim, and Gyeong-Moon Park. Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. In *Proc. of ICCV*, 2023.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proc. of CVPR*, 2017.

[Szatkowski *et al.*, 2023] Filip Szatkowski, Mateusz Pyla, Marcin Przewiezlikowski, Sebastian Cygert, Bartlomiej Twardowski, and Tomasz Trzcinski. Adapt your teacher: Improving knowledge distillation for exemplar-free continual learning. *CoRR*, 2023.

[Tian *et al.*, 2024] Shiyu Tian, Hongxin Wei, Yiqun Wang, and Lei Feng. Crosel: Cross selection of confident pseudo labels for partial-label learning. In *Proc. of CVPR*, 2024.

[Wang *et al.*, 2022a] Deng-Bao Wang, Min-Ling Zhang, and Li Li. Adaptive graph guided disambiguation for partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[Wang *et al.*, 2022b] Haobo Wang, Mingxuan Xia, Yixuan Li, Yuren Mao, Lei Feng, Gang Chen, and Junbo Zhao. Solar: Sinkhorn label refinery for imbalanced partial-label learning. In *NeurIPS*, 2022.

[Wang *et al.*, 2022c] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico: Contrastive label disambiguation for partial label learning. In *Proc. of ICLR*, 2022.

[Welinder *et al.*, 2010] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical report, 2010.

[Wu *et al.*, 2019] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proc. of CVPR*, 2019.

[Xia *et al.*, 2023] Shiyu Xia, Jiaqi Lv, Ning Xu, Gang Niu, and Xin Geng. Towards effective visual representations for partial-label learning. In *Proc. of CVPR*, 2023.

[Yan and Guo, 2023] Yan Yan and Yuhong Guo. Mutual partial label learning with competitive label noise. In *Proc. of ICLR*, 2023.

[Yoo *et al.*, 2024] Jinsoo Yoo, Yunpeng Liu, Frank Wood, and Geoff Pleiss. Layerwise proximal replay: A proximal point method for online continual learning. In *Proc. of ICML*, 2024.

[Yoon *et al.*, 2018] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *Proc. of ICLR*, 2018.

[Yoon *et al.*, 2020] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. In *Proc. of ICLR*, 2020.

[Yu and Zhang, 2017] Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. *Mach. Learn.*, 2017.

[Yu *et al.*, 2024] Xiang-Ru Yu, Deng-Bao Wang, and Min-Ling Zhang. Partial label learning with emerging new labels. *Mach. Learn.*, pages 1549–1565, 2024.

[Yuan *et al.*, 2022] Hangjie Yuan, Jianwen Jiang, Samuel Albanie, Tao Feng, Ziyuan Huang, Dong Ni, and Mingqian Tang. RLIP: relational language-image pre-training for human-object interaction detection. In *NeurIPS*, 2022.

[Zenke *et al.*, 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proc. of ICML*, 2017.

[Zhang *et al.*, 2020] Yabin Zhang, Guang Yang, Suyun Zhao, Peng Ni, Hairong Lian, Hong Chen, and Cuiping Li. Partial label learning via generative adversarial nets. In *Proc. of ECAI*, 2020.

[Zhang *et al.*, 2023] Yabin Zhang, Hairong Lian, Guang Yang, Suyun Zhao, Peng Ni, Hong Chen, and Cuiping Li. Inaccurate-supervised learning with generative adversarial nets. *IEEE Trans. Cybern.*, 2023.

[Zhao *et al.*, 2020] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proc. of CVPR*, 2020.

[Zheng *et al.*, 2024] Bowen Zheng, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Multi-layer rehearsal feature augmentation for class-incremental learning. In *Proc. of ICML*, 2024.