

# Adaptive Discovering and Merging for Incremental Novel Class Discovery

Guangyao Chen<sup>1</sup>, Peixi Peng<sup>2, 3\*</sup>, Yangru Huang<sup>1</sup>, Mengyue Geng<sup>1</sup>, Yonghong Tian<sup>1, 2, 3\*</sup>

<sup>1</sup> National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, China.

<sup>2</sup> School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, China.

<sup>3</sup> Peng Cheng Laboratory, China.

{gy.chen, ppxpeng, yrhuang, mygeng, yhtian}@pku.edu.cn

## Abstract

One important desideratum of lifelong learning aims to discover novel classes from unlabelled data in a continuous manner. The central challenge is twofold: discovering and learning novel classes while mitigating the issue of catastrophic forgetting of established knowledge. To this end, we introduce a new paradigm called Adaptive Discovering and Merging (ADM) to discover novel categories adaptively in the incremental stage and integrate novel knowledge into the model without affecting the original knowledge. To discover novel classes adaptively, we decouple representation learning and novel class discovery, and use Triple Comparison (TC) and Probability Regularization (PR) to constrain the probability discrepancy and diversity for adaptive category assignment. To merge the learned novel knowledge adaptively, we propose a hybrid structure with base and novel branches named Adaptive Model Merging (AMM), which reduces the interference of the novel branch on the old classes to preserve the previous knowledge, and merges the novel branch to the base model without performance loss and parameter growth. Extensive experiments on several datasets show that ADM significantly outperforms existing class-incremental Novel Class Discovery (class-iNCD) approaches. Moreover, our AMM also benefits the class-incremental Learning (class-IL) task by alleviating the catastrophic forgetting problem. The source code is included in the supplementary materials.

## Introduction

A long-standing goal of machine learning is to build Artificial Intelligence (AI) systems that can mimic human-level performance in an open and online manner. Neural networks, originally inspired by human brain structures, should also be flexible enough to incrementally absorb novel concepts (or *classes*) after acquiring previous knowledge. To evaluate the ability of novel class discovery and memory in the open world, several tasks (Han et al. 2020; Han, Vedaldi, and Zisserman 2019; Zhong et al. 2021a,b; Fini et al. 2021; Roy et al. 2022) are proposed to continuously identify new classes while leveraging some previously learned knowledge. Among these task settings, the class-incremental Novel Class Discovery (class-iNCD) (Roy et al. 2022) aims at discovering novel concepts sequentially in a

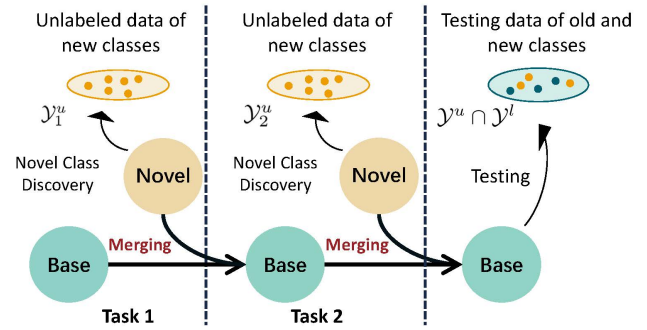


Figure 1: Illustration of incremental novel class discovery with an elastic learner. The novel model adaptively discovers novel classes from the unlabeled data of the current task based on the previous base model. The novel learner should not only affect the knowledge of the base model but also integrate with the base model.

continuous manner. Compared with other settings, models in class-iNCD can not access data in previously seen categories, while still having to perform well across all classes.

This paper focuses on developing efficient neural network models for the class-iNCD task. To achieve continuous learning without revisiting past knowledge, two significant challenges emerge as paramount. The first pertains to the discovery of new categories from unlabeled data, requiring the model to discern and learn novel patterns without explicit annotations. The second challenge is the notorious issue of catastrophic forgetting. As the model learns new concepts, it risks overwriting or diminishing prior knowledge.

To discover new categories and learn from unlabeled data, most existing methods (Han et al. 2020; Zhong et al. 2021a,b; Jia et al. 2021; Fini et al. 2021) rely on pseudo-labels for both data annotation and model supervision. However, the quality of the pseudo-label greatly affects the performance of novel category discovery. To address this issue, we propose to separate representation learning and category discovery in the incremental novel category discovery stage. Specifically, the model learns representations directly from unlabeled data through contrastive learning, while the classifiers are optimized independently with adaptive category discovery. To further enhance the quality of pseudo-label

\*Corresponding Author.

generation and cultivate a robust classifier, we introduce a Triplet Comparison (TC) technique to minimize intra-class distance and maximize inter-class distance. Additionally, a Probability Regularization (PR) technique is also developed to prevent the model from becoming trapped in a local optimal solution specific to a single class. By constraining the contrast and diversity of probability space, the learner could achieve self-adaptive novel category discovery from unlabelled data.

The class-iNCD also faces the well-known catastrophic forgetting problem (McCloskey and Cohen 1989). Several architectural approaches (Rusu et al. 2016; Liu, Schiele, and Sun 2021; Yan, Xie, and He 2021) address this problem by adding multiple new network components for novel data, but increase the model parameters and computations with each sequential task. Alternatively, a basic method of Incremental Model Merging (IMM) without parameter growth can be built based on model re-parameterization (Zagoruyko and Komodakis 2017; Ding et al. 2019; Cao et al. 2020; Guo, Alvarez, and Salzmann 2020; Ding et al. 2021). Specifically, a typical IMM method consists of a base branch and a novel branch, where the novel branch could be merged into the base model according to the additivity of convolution. However, IMM may disturb the base model as the novel branch is learned directly from the novel data. To reduce interference with the base model, we propose an Adaptive Feature Fusion (AFF) mechanism, which leverages the incompatibility of feature magnitude between old and new input of a network branch (Dhamija, Günther, and Boulton 2018; Chen et al. 2020a, 2021). Specifically, AFF uses a gated dynamic unit to maintain the important information of the feature from the base branch. In this way, the novel branch trained by AFF has a low magnitude on the old classes, and brings little influence to the base branch. Since AFF is performed on the fusion of features of the base branch and novel branch rather than their model parameters, we further improve the AFF to Adaptive Model Merging (AMM), where we alternatively approximate the dynamic gating information through the weight information of BN in the base branch to merge the novel branch to the base model.

Capitalizing on both adaptive novel discovery and AMM, our solution forms a new paradigm called Adaptive Discovering and Merging (ADM). As shown in Fig. 1, ADM could discover novel categories adaptively in the incremental stage and integrate novel knowledge into the model without affecting the original knowledge, thereby yielding a marked enhancement in the model’s efficacy. To summarize, the main contributions of our paper are as follows: **(1)** We decouple representation learning and adaptive discovery, and propose TC and PR mechanisms for adaptive novel category assignment by constraining the probability difference and diversity. **(2)** We propose two novel AFF and AMM methods to effectively handle the catastrophic forgetting problem, especially the latter, which can make the novel branch mergeable without additional model parameters through sequential learning tasks. **(3)** Extensive experimental results show that our method not only achieves state-of-the-art performance on several class-iNCD datasets but also has a positive impact on the class-IL task.

## Related Works

**Novel Class Discovery** Novel Class Discovery (NCD) deals with the task of learning to discover novel classes in an unlabelled dataset by utilizing the knowledge acquired from another labeled base dataset (Han, Vedaldi, and Zisserman 2019). It is assumed that the classes in the labeled and unlabelled sets are disjoint. Several NCD methods use a stage-wise training scheme where the model is pre-trained on the labeled base dataset, followed by fine-tuning on the unlabelled data using an unsupervised clustering loss (Hsu, Lv, and Kira 2018; Hsu et al. 2019; Han, Vedaldi, and Zisserman 2019; Liu and Tuytelaars 2022). Barring (Liu and Tuytelaars 2022), none of the above methods consider tackling the forgetting issue, and as a result, the model loses the ability to classify the base classes. The second category comprises NCD methods that assume both the labeled and unlabelled data are available simultaneously, which are then trained jointly (Han et al. 2020; Zhong et al. 2021a,b; Jia et al. 2021; Fini et al. 2021). However, access to seen classes limits the practical applicability of NCD methods. Class-incremental novel class discoverers (class-iNCD) (Roy et al. 2022) is proposed to preserve the ability of the model to recognize previously seen base categories. Compared with class-incremental learning, the novel data of class-iNCD at every stage are not provided with any supervised labels.

**Incremental Learning with Multi-branch Model** Incremental Learning (IL) is a learning paradigm where a model is trained on a sequence of tasks such that data from only the current task is available for training, while the model is evaluated on all the observed tasks. To prevent catastrophic forgetting (McCloskey and Cohen 1989; Goodfellow et al. 2014), existing class-IL approaches with multi-branch models could be categorized into knowledge distillation and architectural methods. Knowledge distillation methods (Jung et al. 2016; Li and Hoiem 2017; Kang, Park, and Han 2022; Wang et al. 2022) achieve better generalization performance by transferring knowledge from a base model to a novel network through matching logits, activations, attention, and so on. Most architectural methods (Rusu et al. 2016; Liu, Schiele, and Sun 2021; Yan, Xie, and He 2021) adjust network capacity dynamically to handle a sequence of incoming tasks. Some methods even dedicate different model parameters for different incremental phases, to prevent model forgetting (caused by parameter overwritten). However, these approaches require creating and storing additional network components and performing multiple forward-pass computations for inference, which incur extra computational costs. Different from these architectural methods, our approach does not continuously increase the network size and even could achieve non-destructive incremental fusion, so as to be used as a plug-and-play unit without extra inference computational costs.

## Proposed Approach

**Problem Setting** Here, we follow the setting of class-incremental Novel Class Discovery (class-iNCD) (Roy et al. 2022), where the lifelong learner not only distinguishes the

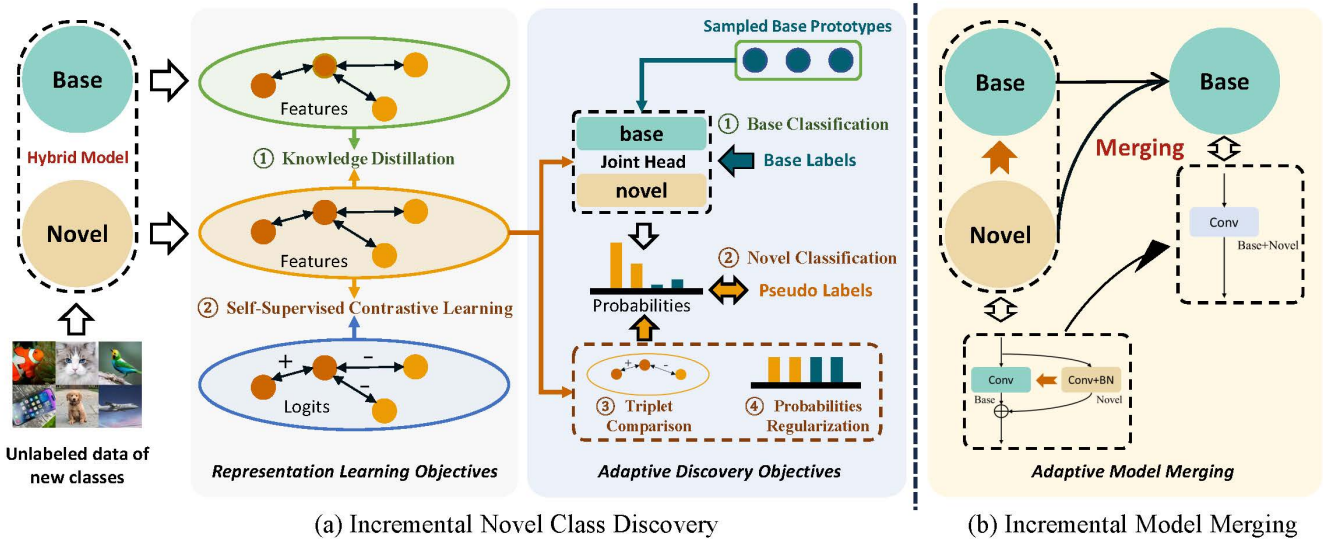


Figure 2: Illustration of incremental novel class discovery with adaptive discovering and merging. (a) For incremental novel class discovery, representation learning and feature-based adaptive novel class discovery are decoupled. Knowledge distillation is applied for representation learning to mitigate catastrophic forgetting, and self-supervised contrastive learning is employed to learn new categories. For adaptive novel discovery, the base head is trained with replayed base prototypes. For the category assignment of new classes, triplet comparison enhances the probability differentiation and probability regularization promotes the diversity to avoid overfitting of categories within a single class. Finally, pseudo labels are generated by the maximum probability. (b) Sketch of adaptive model merging. Each Conv + BN unit is decomposed into the base and novel branch, which could be merged based on the additivity of convolution.

novel classes but also preserves the performance on the previously seen classes, without having access to or storing data from the previous tasks. In the class-iNCD, a initial supervised task contains  $n^l$  instances of a labelled dataset  $\mathcal{D}_0 = \{(x_i^l, y_i^l)\}_{i=1}^{n^l}$ , where  $x^l \in \mathcal{X}^l$  represents the input data and  $y^l \in \mathcal{Y}^l$  as corresponding labels. The following task  $t$  only have the unlabelled dataset  $\mathcal{D}_t^u = \{(x_i^u)\}_{i=1}^{n^u}$ , where  $x^u \in \mathcal{X}^u$  are the unlabelled new data containing  $c^u$  classes. Here, it is assumed that the labels in  $\mathcal{Y}^l$  and  $\mathcal{Y}^u$  are disjoint, *i.e.*,  $\mathcal{Y}^l \cap \mathcal{Y}^u = \emptyset$ . The class-iNCD focuses on clustering the data in  $\mathcal{D}_t^u$  by just leveraging the learned information contained in the single model, while still behaving well on the previous tasks. For the evaluation of the class-iNCD, a joint (task-agnostic) head is adopted to distinguish the old and new classes by directly comparing the predictions with these re-assigned ground-truth labels. Here, HA (Kuhn 1955) is used to re-assign ground-truth IDs based on the predictions and ground-truth labels for the new classes only.

**Overview of ADM** The key challenge to solving class-iNCD is to learn unseen classes from unlabeled data and keep the distinguish for the seen classes. To circumvent this problem, we propose ADM, a novel paradigm that adaptively discovers and integrates new categories in the incremental stage without forgetting the original knowledge. As shown in Fig. 2, ADM consists of two stages: incremental novel class discovery and incremental model merging. The former adaptively discovers novel classes by constraining the probability differences and diversity, and the latter

merges the novel branch to the base model without performance loss and parameter growth. In the following sections, we describe the adaptive process of incremental novel class discovery and the network architectures for non-destructive incremental merging.

## Incremental Novel Class Discovery

**Representation Learning Objectives** Unlike most novel class discovery settings (Hsu, Lv, and Kira 2018; Hsu et al. 2019; Han, Vedaldi, and Zisserman 2019; Liu and Tuytelaars 2022), class-iNCD does not have any labeled data and only provides unlabeled data of new classes. Therefore, our representation learning adopts self-supervised contrastive learning (Chen et al. 2020b) on all samples.

• **Novel Representation Learning.** Formally, given two views (random augmentations)  $\{x_i^u, \hat{x}_i^u\}$  of the same image in a mini-batch  $B$ , the self-supervised contrastive loss is defined as:

$$\mathcal{L}_c^r = -\frac{1}{|B|} \sum_{i \in B} \log \frac{\exp(z_i^\top \hat{z}_i / \tau_u)}{\sum_{j \neq i} \exp(z_i^\top \hat{z}_j / \tau_u)} \quad (1)$$

where the feature  $z_i = g(f_n(x_i))$  and is  $\ell_2$ -normalised,  $f_n, g$  denote the backbone of novel model and the projection head, and  $\tau_u$  is a temperature value.

• **Base Representation Learning.** To further mitigate forgetting on the base classes from representation learning, we follow the FRoST (Roy et al. 2022) by using knowledge distillation (Hinton, Vinyals, and Dean 2015; Li and Hoiem 2017;

Roy et al. 2022) on the novel model to ensure the feature encoding for the old task does not drift too far while learning on the new task, which is formalized as:

$$\mathcal{L}_{\text{KD}}^r = -\mathbb{E}_{p(x^u)} \left\| f_b(x^u) - f_n(x^u) \right\|_2, \quad (2)$$

where  $f_b$  is the base feature extractor from the previous task and is kept frozen. The overall objective for representation learning is given as:  $\mathcal{L}^r = \mathcal{L}_{\text{KD}}^r + \mathcal{L}_c^r$ .

**Adaptive Discovery Objectives** Besides learning generic feature representations to facilitate the discovery of novel categories, another key challenge is to generate pseudo clusters/labels for unlabelled data to guide the learning of a classifier. Moreover, at any time during the training sessions, a single classification head is maintained for all the classes seen for the class-iNCD. Thus, in the novel class discovery phase, the classifier extends a novel head based on the base head to form a joint classifier.

• **Novel Classification.** To learn novel classification, pseudo-labels are computed from novel probabilities  $p^u$  to train the joint classifier  $\mathcal{H}$ . Specifically, we employ the *Logit* output  $\mathcal{P}^u$  of novel head  $\mathcal{H}_n$  to compute the pseudo-label  $\hat{y}^u$  for an unlabelled data  $x^u$ . The  $\hat{y}^u$  is then used to supervise the training of  $\mathcal{H}$ . The self-training for novel classification is formalized as:

$$\mathcal{L}_{\text{self}} = -\mathbb{E}_{(x^u, \hat{y}^u)} \frac{1}{|C^A|} \sum_{k=1}^{|C^A|} \hat{y}_k^u \log \sigma_k(\mathcal{H}(f_n(x^u))), \quad (3)$$

where  $C^A$  in Eq.3 represents the total number of categories that are configured and  $\hat{y}^u = C_{\text{base}} + \arg \max_{k \in C_{\text{novel}}} \mathcal{H}_n(f_n(x^u))$ .

This part of the objective generates pseudo-labels for the unlabeled data to guide the training of the joint classifier.

• **Triplet Comparison.** However, pseudo-labels generated by unconstrained probabilities are inaccurate. Therefore, a similarity prediction function parameterized by a linear novel classifier should be learned such that instances from different classes are grouped into multisets. To achieve this, we reformulate the cluster learning problem as a triplet similarity prediction task. Specifically, to obtain the triplet instances for the unlabeled set, we compute the cosine distance between all pairs of feature representations  $f(x_i)$  in a mini-batch. We then rank the computed distances and for each instance generate the triplet instances for its most similar and dissimilar neighbors, to generate pseudo-labels from the most confident positive pair  $\{\mathcal{P}_a^u, \mathcal{P}_p^u\}$  and negative pairs  $\{\mathcal{P}_a^u, \mathcal{P}_n^u\}$  for each instance within the mini-batch. The objective for triplet comparison is designed as a refined version of the traditional mean absolute error loss, combining two modifications for greater effectiveness:

$$\mathcal{L}_{\text{triplet}} = \frac{1}{C^A} \sum (\sigma(\mathcal{P}_a^u) - \sigma(\mathcal{P}_p^u))^2 - \frac{1}{C^A} \sum (\sigma(\mathcal{P}_a^u) - \sigma(\mathcal{P}_n^u))^2. \quad (4)$$

Here,  $\sigma$  denotes the softmax function which assigns instances to one of the novel classes, and  $m$  is a nonnegative margin representing the minimum difference between the positive and negative distances and set as 1.0. By controlling the intra-class and inter-class variance of novel classes,

the triplet comparison improves the quality of the pseudo-labels.

• **Probabilities Regularization.** In the early stages of the training, the network could degenerate to a trivial solution in which all instances are assigned to a single class. To prevent this solution, we introduce a regularization term for the probabilities by applying a maximum entropy regularization that regularizes the mean of all probabilities of novel classes:

$$\mathcal{L}_r = -\frac{1}{|B|} \sum_{x_i \in \mathcal{D}_t^u} \sigma(\mathcal{H}_n(f_n(x_i))) \log \left( \frac{1}{|B|} \sum_{x_i \in \mathcal{D}_t^u} \sigma(\mathcal{H}_n(f_n(x_i))) \right). \quad (5)$$

Here, maximum entropy regularization for the mean of all probabilities of novel classes encourages diversity to avoid the overfitting of categories within a single class.

• **Base Classification.** To further mitigate forgetting on the base classes for the classifier, we follow the FRoST (Roy et al. 2022) by using *generative feature-replay* drawn from a Gaussian distribution of base feature prototypes  $p_c$  is used to preserve the performance of old classes. While learning on the new task  $t$ ; the weights of the joint classifier  $\mathcal{H}$ , corresponding to the base classes  $C_{\text{base}}$ , are trained by replaying features from the class-specific Gaussian distribution  $\mathcal{N}(\mu_c, \mathbf{v}_c^2)$ . The feature-replay loss is formalized as:

$$\mathcal{L}_{\text{base}}^c = -\mathbb{E}_{c \sim C} \mathbb{E}_{(z, y_c) \sim \mathcal{N}(\mu_c, \mathbf{v}_c^2)} \sum_{k=1}^{|C_{\text{base}}|} y_{kc} \log \sigma_k(\mathcal{H}^A(p)). \quad (6)$$

Then, the loss for discovering novel classes and having a single classifier for all the classes seen so far can be formalized as  $\mathcal{L}^c = \mathcal{L}_{\text{base}}^c + \{\mathcal{L}_{\text{triplet}}^c + \mathcal{L}_r^c + \omega(t)\mathcal{L}_{\text{self}}^c\}$ , where  $\omega(t)$  are ramp-up functions to ensure stability in learning. Finally, the total objective function is  $\mathcal{L} = \mathcal{L}^r + \mathcal{L}^c$ , which consists of two components: the representation learning loss  $\mathcal{L}^r$  and the classifier loss  $\mathcal{L}^c$ .

## Incremental Model Merging

To further integrate novel knowledge with the base knowledge, we further proposed a two-branch hybrid architecture that could make an only-novel-training ensemble of numerous models. In detail, the base approach of incremental model merging is first described, which could achieve a non-destructive incremental merging mechanism. Second, adaptive feature fusion is proposed to guide the novel branch to generate a lower magnitude to old categories and fuse two branches at the feature level. Then we further proposed adaptive model merging to approximate the dynamic gating output through the weight of BN in the base branch, and achieve linear additional fusion in the model level.

**Incremental Model Merging** Since the multi-branch topology has drawbacks for inference but the base branch seems beneficial to novel discovery and learning, we employ base and novel branches to make an *only-novel-training* ensemble of numerous models as shown in Fig. 3(a). Based on the homogeneity of convolution (Ding et al. 2019, 2021), each CONV + BN unit the base model trained on labeled dataset  $\mathcal{D}^l$  is fused to a single convolution layer with weight

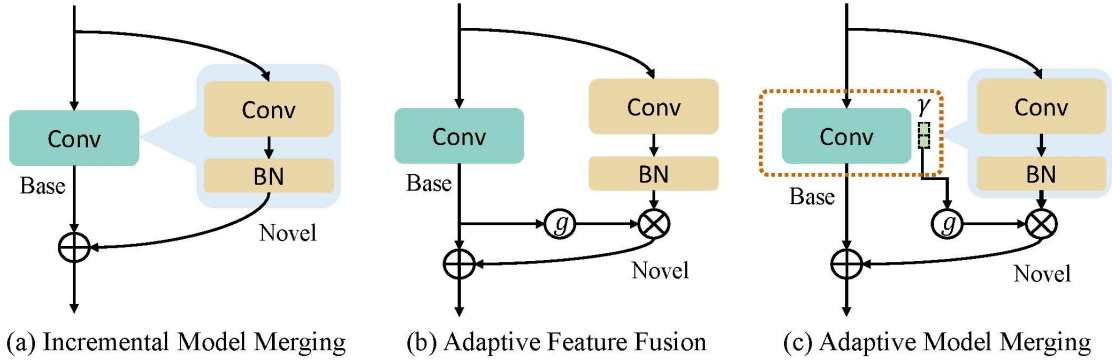


Figure 3: Conceptual illustrations of different merging methods. (a) Incremental Model Merging (IMM) (Zhu et al. 2022) decomposes each CONV + BN operation into the base and novel parts, then fuses these two parts based on the additivity of convolution. (b) Adaptive Feature Fusion (AFF) adopts the gated output of the base branch to guide the learning process and outputs of the novel branch. (c) Adaptive Model Merging (AMM) uses the weight of base BN to replace dynamic gated output, so as to achieve linear merging.

$W'_{(b)}$  and bias  $B'_{(b)}$ . Then, for the stage of novel class discovery, we extend a novel branch with a CONV + BN unit, where  $W'_{(n)} \in \mathbb{R}^{C_2 \times C_1}$  for the kernel with the same size of  $W'_{(b)}$ , and  $\mu_{(n)}, \sigma_{(n)}, \gamma_{(n)}, \beta_{(n)}$  for the BN following the novel convolution. For input  $X^{(1)}$  and output  $X^{(2)}$ , we have

$$\begin{aligned} X^{(2)} &= (X^{(1)} * W'_{(b)}) + B'_{(b)} \\ &+ \text{BN}(X^{(1)} * W_{(n)}, \mu_{(n)}, \sigma_{(n)}, \gamma_{(n)}, \beta_{(n)}). \end{aligned} \quad (7)$$

In the novel categories training stage, the base branch parameters  $\{W'_{(b)}, B'_{(b)}\}$  are fixed, and only the novel branch parameters  $\{W_{(n)}, \mu_{(n)}, \sigma_{(n)}, \gamma_{(n)}, \beta_{(n)}\}$  is trained, where the novel output is added to the base output, so as to integrate the base knowledge.

After novel class discovery, we will have two  $k \times k$  kernels, and two bias vectors. Then we obtain the final base kernel  $W' = W'_{(b)} + W'_{(n)}$  and bias  $B' = B'_{(b)} + B'_{(n)}$  by adding up the two kernels and two bias vectors. For input  $X^{(1)}$  and output  $X^{(2)}$ , we could get

$$\begin{aligned} X^{(2)} &= (X^{(1)} * W'_{(b)} + B'_{(b)}) + (X^{(1)} * W'_{(n)} + B'_{(n)}) \\ &= X^{(1)} * (W'_{(b)} + W'_{(n)}) + (B'_{(b)} + B'_{(n)}) \\ &= X^{(1)} * W' + B'. \end{aligned} \quad (8)$$

In the novel categories inference stage, the base and novel branches could be merged based on the additivity of convolution without any output losses. For the next incremental learning stage, a new novel branch could be extended again and continuously merge.

**Adaptive Feature Fusion** IMM could learn novel branches based on the base branch, but could not ensure the discrimination of final features for old tasks is not affected. Here, we argue that the old categories could be seen as the unknown for the novel branch, thus the novel branch would output relatively lower magnitude features due to the

absence of old training data (Dhamija, Günther, and Boulton 2018; Chen et al. 2020a, 2021). To further reduce the magnitude of the novel branch to the old classes, we propose an Adaptive Feature Fusion (AFF) mechanism to guide the learning process of the novel branches to focus on the unimportant parts of the base. As shown in Fig.3(b), we use the base outputs to control the propagated information through the novel branch. The larger the output value of the base branch is, the more important its corresponding neurons are. So we control the update of the novel branch by using negative gating functions of base outputs. With the gated function  $\mathcal{G}(\cdot)$ , the AFF with base branch  $f_b(\cdot)$  and novel branch  $f_n(\cdot)$  could be formalized as:

$$\begin{aligned} X^{(2)} &= f_b(X^{(1)}) + g(-f_b(X^{(1)})) \otimes f_n(X^{(1)}) \\ &= (X^{(1)} * W'_{(b)} + B'_{(b)}) \\ &- \mathcal{G}(X^{(1)} * W'_{(b)} + B'_{(b)}) \otimes (X^{(1)} * W'_{(n)} + B'_{(n)}), \end{aligned} \quad (9)$$

where  $\mathcal{G}(\cdot)$  is a *sigmoid* function. Moreover, the gradient of AFF for the novel branch is  $\frac{\partial X^{(2)}}{\partial f_n(X^{(1)})} = -\mathcal{G}(f_b(X^{(1)}))$ . Hence, the update of the novel branch is influenced by the outputs of the base branch, and the magnitude for old categories is further suppressed as shown in Fig. 4.

**Adaptive Model Merging** The gating process of AFF is one that varies dynamically according to the sample, which could not implement the merging operation according to the additivity of convolution, as shown in Eq. (9). To solve this problem, we rethink the importance of BN. For a certain BN layer,  $\mu$  and  $\sigma$  compute the mean and the standard deviation, respectively, of all activations overall pixel locations for the current mini-batch data;  $\gamma$  and  $\beta$  are the trainable scaling factor and offset, respectively. The factor  $\gamma$  evaluates the correlation between the input and output during training. The gradient of the loss will approach 0 if  $\gamma \rightarrow 0$  at one training step, implying that the input will almost lose its influence on the final prediction and become redundant

---

**Algorithm 1: Adaptive Discovering and Merging**

---

- 1: **Input:** current base model  $M_b^t(\cdot)$ , current task dataset  $\mathcal{D}_t^u$ , and current base weight  $\gamma_{(b)}^t$
  - 2: **Novel Branch Expansion:**  $M^t = \{M_b^t(\cdot), M_n^t(\cdot)\}$
  - 3: **for** all  $(x_1, x_2, \dots, x_n) \in \mathcal{D}_t^u$  **do**
  - 4:   Compute output of  $M^t$  based on Eq.(10).
  - 5:   Compute adaptive novel class discovery loss  $\mathcal{L} = \mathcal{L}^r + \mathcal{L}^c$  for mini-batch.
  - 6:   Perform backward through  $M_n^t(\cdot)$ .
  - 7: **end for**
  - 8: CONV + BN fusion of  $M_n^t(\cdot)$ .
  - 9: **Adaptive Model Merging** of base-novel parameters  $\{M_b^t(\cdot), M_n^t(\cdot)\}$  based on Eq.(11).
  - 10: **Weight update** of  $\gamma_{(b)}^{t+1}$  based on Eq.(12).
  - 11: **Output:** The base model  $M_b^{t+1}(\cdot)$  and weight  $\gamma_{(b)}^{t+1}$  for task  $t + 1$ .
- 

thereby at this training step. That means that if the scaling factor of one channel (with sparsity constraints) is lower than the small threshold at one training step, this channel will hardly recover and almost become redundant during the later training process. Therefore, it motivates us to replace the input of the gated function with  $\gamma_{(b)}$  of the base BN. As shown in Fig.3(c), we adopt  $\gamma_{(b)}$  of the base BN to control the propagated information through the novel branch, which could be formalized as:

$$X^{(2)} = f_b(X^{(1)}) + \mathcal{G}(-\gamma_{(b)}) \otimes f_n(X^{(1)}). \quad (10)$$

Then, the base and novel branches could be merged based on linear additivity:

$$X^{(2)} = X^{(1)} * (W'_{(b)} + g(-\gamma_{(b)}) \otimes W'_{(n)}) + (B'_{(b)} + \mathcal{G}(-\gamma_{(b)}) \otimes B'_{(n)}). \quad (11)$$

Since the base parameter is invariant during the incremental learning phase,  $\gamma_{(b)}$  corresponds to a constant, allowing merging to be implemented. Moreover, the merging process of  $\gamma$  during base-novel merging could also be added linearly as:

$$\gamma' = \gamma'_{(b)} + \mathcal{G}(-\gamma_{(b)}) \otimes \gamma'_{(n)}. \quad (12)$$

Overall, Algorithm 1 summarizes the details of incremental category discovery with gated linear merging.

## Experiments

### Experimental Setup

**Datasets.** We employ three datasets to conduct experiments for class-iNCD: CIFAR-10 (Krizhevsky and Hinton 2009), CIFAR-100 (Krizhevsky and Hinton 2009) and Tiny-ImageNet (Le and Yang 2015). The CIFAR10 and CIFAR100 datasets contain 50,000 and 10,000  $32 \times 32$  color images for training and testing. The Tiny-ImageNet contains 100,000 images of 200 classes downsized to  $64 \times 64$  colored images. Each dataset is split into the old and new classes following (Han et al. 2020; Zhong et al. 2021a; Liu and Tuytelaars 2022; Roy et al. 2022). For CIFAR10, 5 base classes

and 5 novel classes are randomly sampled. 20 novel classes are sampled for CIFAR100 and TinyImageNet and the rest are base classes.

**Evaluation Metrics.** We used the evaluation protocol in (Roy et al. 2022) to evaluate the performance of the test data for all the classes. Three classification accuracies are reported, denoted as **Old**, **New**, and **All**. They represent the accuracy obtained from the joint classifier head on the samples of the old, new, and old+new classes, respectively. We used ResNet-18 (He et al. 2016) as the backbone in all the experiments. We have adopted most of the implementation following (Roy et al. 2022).

## Results

**One-Step Class-iNCD** As shown in Table 1, all the NCD methods (AutoNovel (Han et al. 2020), ResTune (Liu and Tuytelaars 2022), NCL (Zhong et al. 2021a), DTC (Han, Vedaldi, and Zisserman 2019)) fail to achieve a good balance between the old and new classes, implying that the baseline methods can maintain the performance in old classes. FRoST (Roy et al. 2022) can balance the old and new categories after learning the new category without severe catastrophic forgetting. We first apply AFF and AMM to the SOTA class-iNCD method FRoST (Roy et al. 2022). AFF could be applicable to FRoST because it only has one incremental novel discovery task to learn. The combination of AFF and FRoST not only further enhances the performance of the old categories, but also boosts the performance of the new categories. The improvement of the old categories is mainly attributed to the gating mechanism that constrains the learning parameters of the novel branch. Meanwhile, the gated dynamic units enable the novel branch to exploit the feature diversity of the base branch. Moreover, AMM replaces the dynamic gating mechanism with the  $\gamma$  weight of BN, which achieves comparable performance to AFF and even outperforms AFF in overall performance. This also demonstrates the effectiveness of the  $\gamma$  weight of BN as an important assessment. Furthermore, when combining adaptive novel class discovery with AMM, we achieve the best performance in all settings.

## Further Analysis

**The Plug-and-play Architectural Approach** We evaluate the effectiveness of AMM in class-incremental learning with different numbers of steps. We apply AMM to five state-of-the-art methods: iCaRL (Rebuffi et al. 2017), PODNet (Douillard et al. 2020), NECIL (Zhu et al. 2022), AFC (Kang, Park, and Han 2022) and DyTox (Douillard et al. 2022). As shown in Table 2, our approach significantly improves the performance of all methods in all incremental stages. Moreover, the improvement increases as the number of stages grows, indicating that our method is more robust to challenging scenarios than the previous methods. Notably, DyTox (Douillard et al. 2022) is a class-IL method based on Transformer architecture, which can be combined with AMM. We add AMM to the *Linear* layer in MLP and achieve better performance than DyTox alone.

Methods	CIFAR-10			CIFAR-100			Tiny-ImageNet			Average		
	Old	New	All	Old	New	All	Old	New	All	Old	New	All
AutoNovel	27.5	3.5	15.5	2.6	15.2	5.1	2.0	26.4	4.5	10.7	15.0	8.4
ResTune	91.7	0.0	45.9	<b>73.8</b>	0.0	59.0	44.3	0.0	39.9	<b>69.9</b>	0.0	48.3
NCL	<b>92.0</b>	1.1	46.5	73.6	10.1	60.9	0.8	6.5	1.4	55.5	5.9	36.3
DTC	64.0	0.0	32.0	55.9	0.0	44.7	35.5	0.0	32.0	51.8	0.0	36.2
FRoST	77.5	49.5	63.4	64.6	<b>45.8</b>	59.2	54.5	33.7	52.3	65.5	43.1	58.3
FRoST+AFF	77.3	51.5	64.1	65.5	<b>45.8</b>	61.0	56.4	35.0	54.3	66.4	44.1	59.9
FRoST+AMM	76.8	<b>52.6</b>	64.2	65.6	43.7	61.1	57.0	35.0	54.7	66.5	43.8	60.2
<b>ADM</b>	78.4	51.9	<b>65.2</b>	67.9	45.7	<b>63.2</b>	<b>57.8</b>	<b>36.0</b>	<b>55.6</b>	68.0	<b>44.5</b>	<b>61.3</b>

Table 1: Comparison with state-of-the-art methods in the one-step setting of class-iNCD. All values are percentages and the best results are indicated in bold.

Methods	Param	25 stages	10 stages
iCaRL	469K	50.60	53.78
iCaRL+AAANets	530K	56.43	60.26
iCaRL+AMM	469K	<b>60.59</b>	<b>62.20</b>
PODNet	469K	60.72	63.19
PODNet+AAANet	530K	62.31	64.31
PODNet+AMM	469K	<b>63.45</b>	<b>64.32</b>
NECIL†	11.22M	61.01	61.07
NECIL + AMM	11.22M	<b>61.49</b>	<b>61.83</b>
AFC	497K	63.89	64.98
AFC+AMM	497K	<b>65.31</b>	<b>66.01</b>
DER	112.27M	-	<b>72.81</b>
FOSTER	11.22M	63.83	67.95
DyTox+†	10.73M	64.01	69.28
DyTox+ + AMM	10.73M	<b>65.67</b>	<b>70.52</b>

Table 2: Performance comparison with state-of-the-art class-IL methods on CIFAR100. Param is evaluated for the model in the novel inference time. † denotes the reproduction results based on the official code.

**Comparison with Architectural Methods** DER (Yan, Xie, and He 2021) is a two-stage learning approach that uses a dynamically expandable representation, which increases the computation and number of parameters with the incremental stages. In contrast, AMM achieves model fusion without introducing additional parameters or requiring any training, thus avoiding performance loss. FOSTER (Wang et al. 2022) also has base and novel models, but its model fusion relies on knowledge distillation, which could degrade the performance. NECIL (Zhu et al. 2022) uses Re-Parameterization (RP) as a combination of our IR and a prototype-based method. However, IR negatively affects the base branch without considering any base architectural prior. To address this problem, AMM adds a gating mechanism to IR to limit the influence of the novel branch on the base branch and preserve the lossless fusion property of RP. As shown in Table 2, AMM outperforms NECIL by replacing its RP with ours.

**The Role of the Gated Unit** Most unknown samples have lower deep magnitude features from the neural network trained with known samples (Dhamija, Günther, and Boulton

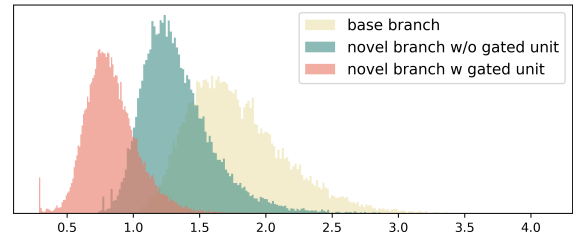


Figure 4: The max magnitude distribution of old categories on the base and novel branches.

2018), which is verified by open-set recognition (Chen et al. 2020a, 2021). Therefore, the novel branch should produce low-magnitude output for old categories, to reduce the influence of old features. As shown in Fig. 4, the novel branch with the proposed gated unit achieves a lower magnitude, thus having less impact on the base branch. Hence, the proposed AFF and AMM preserve important base features. This also demonstrates the effectiveness of the gated unit. The gated unit has two advantages: first, it reduces the interference of the new branch output to the important features of the base in the input phase, and leverages the discriminative power of the base features; second, it guides the novel branch to focus on the learning of unimportant parameters in the base branch, thus minimizing the influence of old categories.

## Conclusion

In this paper, we tackle two challenges in incremental novel class discovery: how to leverage novel unlabeled data sets for effective training guidance and how to prevent catastrophic forgetting of previous knowledge. We propose Triple Comparison and Probability Regularization to control the probability discrepancy and diversity of categories for adaptive category assignment. In addition, we design a hybrid structure, Adaptive Model Merging, which preserves the previous knowledge by reducing the novel branch’s interference with the old classes. Extensive experiments on class-iNCD demonstrate that our method can significantly outperform the existing methods without increasing the computational cost.

## Acknowledgements

The study was funded by the National Natural Science Foundation of China under contracts No. 62332002, No. 62027804, No. 62088102, No. 61825101, No. 62372010, and No. 62202010, and the major key project of the Peng Cheng Laboratory (PCL2021A13). Computing support was provided by Pengcheng Cloudbrain.

## References

- Cao, J.; Li, Y.; Sun, M.; Chen, Y.; Lischinski, D.; Cohen-Or, D.; Chen, B.; and Tu, C. 2020. DO-Conv: Depthwise Over-parameterized Convolutional Layer. *arXiv preprint arXiv:2006.12030*.
- Chen, G.; Peng, P.; Wang, X.; and Tian, Y. 2021. Adversarial Reciprocal Points Learning for Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Chen, G.; Qiao, L.; Shi, Y.; Peng, P.; Li, J.; Huang, T.; Pu, S.; and Tian, Y. 2020a. Learning open set network with discriminative reciprocal points. In *European Conference on Computer Vision*, 507–522. Springer.
- Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*.
- Dhamija, A. R.; Günther, M.; and Boulton, T. 2018. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems*, 9157–9168.
- Ding, X.; Guo, Y.; Ding, G.; and Han, J. 2019. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE International Conference on Computer Vision*, 1911–1920.
- Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13733–13742.
- Douillard, A.; Cord, M.; Ollion, C.; and Robert, T. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *ECCV*.
- Douillard, A.; Ramé, A.; Couairon, G.; and Cord, M. 2022. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9285–9295.
- Fini, E.; Sangineto, E.; Lathuilière, S.; Zhong, Z.; Nabi, M.; and Ricci, E. 2021. A unified objective for novel class discovery. In *ICCV*.
- Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2014. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *ICLR*.
- Guo, S.; Alvarez, J. M.; and Salzmann, M. 2020. ExpandNets: Linear Over-parameterization to Train Compact Convolutional Networks. *Advances in Neural Information Processing Systems*, 33.
- Han, K.; Rebuffi, S.-A.; Ehrhardt, S.; Vedaldi, A.; and Zisserman, A. 2020. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*.
- Han, K.; Vedaldi, A.; and Zisserman, A. 2019. Learning to discover novel visual categories via deep transfer clustering. In *ICCV*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.
- Hsu, Y.-C.; Lv, Z.; and Kira, Z. 2018. Learning to cluster in order to transfer across domains and tasks. In *ICLR*.
- Hsu, Y.-C.; Lv, Z.; Schlosser, J.; Odom, P.; and Kira, Z. 2019. Multi-class classification without multi-class labels. In *ICLR*.
- Jia, X.; Han, K.; Zhu, Y.; and Green, B. 2021. Joint representation learning and novel category discovery on single-and multi-modal data. In *ICCV*.
- Jung, H.; Ju, J.; Jung, M.; and Kim, J. 2016. Less-forgetting Learning in Deep Neural Networks. *arXiv preprint arXiv:1607.00122*.
- Kang, M.; Park, J.; and Han, B. 2022. Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16071–16080.
- Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images.
- Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*.
- Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*.
- Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *TPAMI*.
- Liu, Y.; Schiele, B.; and Sun, Q. 2021. Adaptive Aggregation Networks for Class-Incremental Learning. In *CVPR*.
- Liu, Y.; and Tuytelaars, T. 2022. Residual tuning: Toward novel category discovery without labels. *TNNLS*.
- McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The sequential Learning Problem. In *Psychology of Learning and Motivation*. Elsevier.
- Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *CVPR*.
- Roy, S.; Liu, M.; Zhong, Z.; Sebe, N.; and Ricci, E. 2022. Class-incremental Novel Class Discovery. In *European Conference on Computer Vision*, 317–333. Springer.
- Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*.
- Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022. Foster: Feature boosting and compression for class-incremental learning. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, 398–414. Springer.

Yan, S.; Xie, J.; and He, X. 2021. DER: Dynamically Expandable Representation for Class Incremental Learning. In *CVPR*.

Zagoruyko, S.; and Komodakis, N. 2017. Diracnets: Training very deep neural networks without skip-connections. *arXiv preprint arXiv:1706.00388*.

Zhong, Z.; Fini, E.; Roy, S.; Luo, Z.; Ricci, E.; and Sebe, N. 2021a. Neighborhood Contrastive Learning for Novel Class Discovery. In *CVPR*.

Zhong, Z.; Zhu, L.; Luo, Z.; Li, S.; Yang, Y.; and Sebe, N. 2021b. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *CVPR*.

Zhu, K.; Zhai, W.; Cao, Y.; Luo, J.; and Zha, Z.-J. 2022. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9296–9305.