Passive to Proactive: Counterfactual Boundary Augmentation for Class-Incremental Learning

Anonymous submission

Abstract

Class-Incremental Learning (CIL) aims to enable models to continuously learn new classes without forgetting previously acquired knowledge. Existing CIL methods rely primarily on passive knowledge preservation strategies and lack proactive expansion of decision boundaries, leading to severe confusion between similar classes. To address these issues, we propose two proactive strategies: Counterfactual Boundary Augmentation (CBA) and Temporal Instability-Aware Weighting (TIAW) for replay-based CIL. CBA explicitly broadens the decision boundaries between old and new classes by synthesizing intermediate counterfactual samples for highly confusable new-old class pairs and training the model with an uncertainty-promoting constraint, thereby enhancing class separability. To further amplify the effect of CBA's boundary expansion, TIAW introduces a novel temporal instability metric to dynamically adjust the loss weights of training samples, placing greater emphasis on samples that exhibit high prediction instability, thus enhancing training effectiveness. These two modules are model-agnostic and can be seamlessly integrated into various replay-based CIL methods to enhance their resistance to forgetting. Experimental results on multiple datasets demonstrate that incorporating our approaches into various classic CIL methods yields significant performance improvements of over 2%.

Introduction

Class-Incremental Learning (CIL) aims to enable a model to continuously learn new classes over time while preserving performance on previously learned classes, which is crucial for open-world image recognition applications. However, due to the dynamic nature of data distribution and the training process, CIL models are prone to *catastrophic forgetting* (McCloskey and Cohen 1989), where recognition performance on previously learned classes deteriorates significantly. Given that each incremental step allows only a limited number of old-class samples to be stored for replay, it becomes challenging to maintain clear and robust decision boundaries between old and new classes.

Current mainstream approaches in CIL can be broadly categorized into six aspects: Parameter Regularization (e.g., EWC (Kirkpatrick et al. 2017)), Knowledge Distillation methods (e.g., iCaRL (Rebuffi et al. 2017)), Data Replay (e.g., RM (Bang et al. 2021)), Dynamic Architecture (e.g., DER (Yan, Xie, and He 2021)), Model Rectify (e.g., BiC (Wu et al. 2019)), and Template-Based Classification (e.g., ZSTCI (Wei et al. 2021)), which alleviate catastrophic forgetting through different mechanisms. However, the aforementioned categories are not mutually exclusive. In practice, most CIL methods integrate two or more of these strategies to achieve stronger resistance to forgetting.

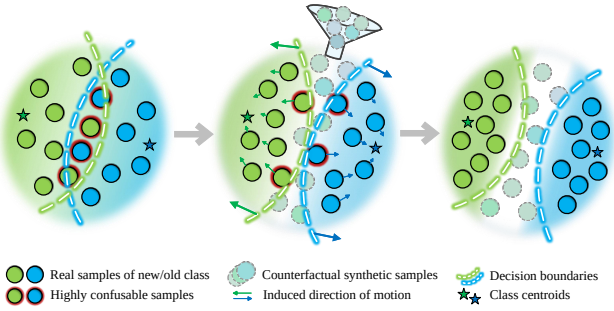

Figure 1: Illustration of decision boundary augmentation via counterfactual synthetic samples.

However, Although most existing replay-based methods employ various techniques to preserve learned knowledge and mitigate the overall forgetting issue, they largely adopt a passive strategy, without proactively addressing the ambiguity of decision boundaries between old and new classes. Specifically, when encountering semantically similar classes (e.g., "dog" and "wolf"), CIL models often struggle to distinguish between them due to insufficiently expanded decision boundaries. Psychological studies have demonstrated that humans tend to actively construct contrast-enhanced representations via counterfactual reasoning when processing semantic category boundaries (Murphy and Medin 1985). This cognitive mechanism is mathematically isomorphic to the principle of margin maximization in support vector machines (Cortes and Vapnik 1995), highlighting the importance of promoting boundary discrimination to enhance classification performance.

In this paper, We propose a paradigm shift termed "Passive to Proactive", which aims to achieve a new type of CIL strategy centered on proactively expanding the boundaries of discrimination (Figure 1). Specifically, we propose two key strategies: *Counterfactual Boundary Augmentation (CBA)* and *Temporal Instability-Aware Weighting (TIAW)*. Firstly,

CBA identifies the most confusing new-old class pairs using a prototype-based metric, and synthesizes intermediate counterfactual samples by combining image-level CutMix and latent-space interpolation with perturbation. These samples, assigned symmetric soft labels, are incorporated into training with an entropy-maximizing constraint, encouraging the model to maintain uncertainty near decision boundaries. This strategy explicitly expands class margins and improves the model's discriminative capability. Secondly, TIAW introduces the concept of "temporal instability", further amplifying the value of counterfactual samples in CBA and that of limited examplars. It dynamically adjusts samples' loss weights based on their prediction instability across training epochs, prompting the model to place more emphasis on hard samples and thereby improving training effectiveness.

The proposed CBA and TIAW modules are model-agnostic and can be flexibly embedded into various replay-based CIL frameworks. We integrate our method into several classic CIL baseline methods and conduct extensive experiments on common datasets. The results show that our CBA and TIAW can significantly improve the prediction accuracy and forgetting resistance. In summary, our main contributions include:

- We propose a proactive CIL method called **Counterfactual Boundary Augmentation (CBA)**, which explicitly expands the decision boundaries between highly confusable new-old class pairs by leveraging synthesized counterfactual samples, effectively enhancing inter-class discriminability and alleviating the forgetting issue.
- We propose **Temporal Instability-Aware Weighting (TIAW)**, which introduces a temporal instability metric to adaptively adjust the loss weights for samples with high instability scores, facilitating more effective learning from counterfactual samples and limited exemplars.
- Extensive experimental results demonstrate that the proposed CBA and TIAW, as model-agnostic plug-and-play modules, can significantly improve the performance of various replay-based CIL methods.

Related Work

The key challenge of CIL is catastrophic forgetting – the model's drastic loss of old-class performance when trained on new classes. To address this issue, various methods have been proposed.

Parameter regularization methods impose penalties on changes in important weights, such as EWC (Kirkpatrick et al. 2017) and SI (Zenke, Poole, and Ganguli 2017), though these methods showed limited performance in large-scale visual CIL tasks. **Knowledge distillation** (Hinton, Vinyals, and Dean 2015) is widely adopted to preserve old knowledge. LwF (Li and Hoiem 2017) distills predictions of a fixed old model when exemplar storage is not allowed. Many exemplar-based approaches enhance this by integrating distillation with rehearsal, *e.g.,* iCaRL (Rebuffi et al. 2017) combines a distillation loss with a nearest-mean classifier. More advanced methods, such as LUCIR (Hou et al. 2019), introduces cosine-normalized classifiers along with a margin ranking loss to better separate representations of old and new classes. PODNet (Douillard et al. 2020) further preserves multi-scale intermediate representations via pooled output distillation, enabling stronger performance across longer task sequences. **Replay-based** methods maintain a small memory of past exemplars or use generative models to produce pseudo-old data. The seminal iCaRL (Rebuffi et al. 2017) utilizes *herding*-based strategy to select and store exemplars for old classes. Some methods employ generative models, such as conditional GAN and diffusion models, to generate replay samples, thereby reinforcing previously learned knowledge (Xiang et al. 2019; Gao and Liu 2023). BI-R (Van de Ven, Siegelmann, and Tolias 2020) argues that generating features instead of images offers advantages in terms of computational complexity and semantic preservation, and employs a variational autoencoder (VAE) to model feature representations. **Dynamic architecture** methods expand model capacity for new tasks. For instance, DER (Yan, Xie, and He 2021) introduces new model branches for each task and uses distillation to retain previous knowledge, while DyTox (Douillard et al. 2022) employs Transformer-based expandable tokens for each task to achieve task-specific representation learning. In addition, approaches based on **Model Rectify** (Wu et al. 2019; Zhao et al. 2020) and **Template-based Classification** (Wei et al. 2021; Zhou et al. 2022) have also demonstrated promising performance.

Replay-based methods have attracted considerable attention due to the intuitiveness and effectiveness. Below, we summarize some of the latest advancements in this category. TagFex (Zheng et al. 2025) injects task-agnostic self-supervised features to mitigate inter-task feature clash. AHR (Nori, Kim, and Wang 2025) introduces a dual-purpose hybrid autoencoder to that compresses past samples into latent codes and reconstructs them for replay. CCFA (Kim, Park, and Han 2024) generates old-class features via cross-class adversarial perturbations to bolster boundaries. MRFA (Zheng et al. 2024) applies layer-wise perturbations on exemplars to enlarge margins and curb forgetting. Gao et al.'s method (Gao et al. 2025) normalizes full-class logits before knowledge distillation, eliminating KD-CE conflict and ensuring fair old–new predictions. MTD (Wen et al. 2024) Builds diverse in-network teachers and applies multi-teacher distillation to balance old–new knowledge.

Most existing replay-based methods do not explicitly expand the decision boundaries between old and new classes. Instead, their boundaries are passively shaped by a limited set of stored exemplars, which often fails to provide sufficient discriminability for highly confusable classes. This limitation motivates us to explore proactive augmentation strategies targeted at enhancing class separation near decision boundaries.

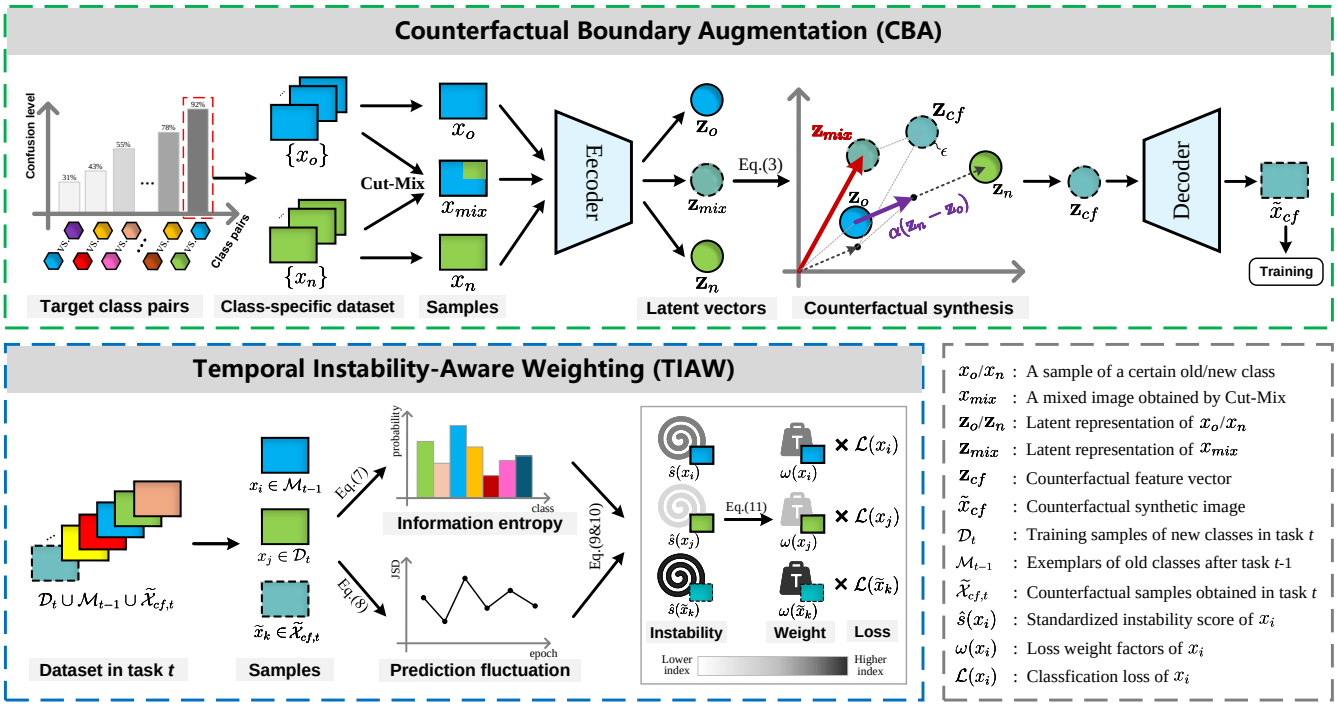

Figure 2: The overall framework of the proposed CBA and TIAW modules.

Method

Problem Formulation

In Class-Incremental Learning (CIL) setting, the models learn a series of class sets incrementally. We adopt the standard B-k Inc-t setting: at task t, we introduce the new class set $\mathcal{C}_t$ with training dataset $\mathcal{D}_t = \{(x_{t,i}, y_{t,i})\}_{i=1}^{N_t}$, where $y_{t,i} \in \mathcal{C}_t$. To alleviate catastrophic forgetting, our approach maintains a small exemplar memory $\mathcal{M}_{t-1}$ that stores a limited number of representative samples (exemplars) from previously learned classes. The model consists of a feature extractor $f_\theta^{(t)}(\cdot)$ and a classifier $g_\phi^{(t)}(\cdot)$. The prototype for class c is defined as $\mu_c^{(t)}$, and the covariance is $\Sigma_c^{(t)}$. While learning new classes at task t, the model combines new data with exemplars from old classes, retaining prior knowledge under limited storage.

We introduces two complementary modules that operate in tandem. CBA proactively expands the decision boundaries by synthesizing counterfactual boundary samples, while TIAW dynamically reweights the training loss based on temporal instability metrics to emphasize currently challenging samples. The overall framework of these two modules is illustrated in Figure 2. In the following sections, we provide detailed descriptions of each module.

Counterfactual Boundary Augmentation

Relying solely on a limited number of replayed exemplars is insufficient to ensure clear and robust classification boundaries between new and old classes. Particularly, when new classes closely resemble old ones, models tend to become confused, leading to overly confident misclassification near decision boundaries. To address this issue, we propose Counterfactual Boundary Augmentation (CBA) method, explicitly broadening decision boundaries by synthesizing samples situated between confusing class pairs. The core idea is to identify pairs of semantically similar samples from new and old classes, combine their semantic information in latent feature space with added perturbations, and generate intermediate counterfactual samples. These samples are then incorporated into model training with specially designed constraints, forcing the model to maintain uncertainty on these synthetic samples, thereby smoothing and relaxing the classification boundary and improving the separability between new and old classes.

Similar Sample pairs Retrieval. We first identify the most easily confused pairs among all new–old class combinations. To this end, we propose a prototype-based metric:

$$\Delta(c_n, c_o) = \|\mu_{c_n}^{(t)} - \mu_{c_o}^{(t-1)}\|_2 - \lambda_d \|\mu_{c_o}^{(t)} - \mu_{c_o}^{(t-1)}\|_2 - \lambda_s \cdot \mathrm{tr}\left(\Sigma_{c_n}^{(t)} + \Sigma_{c_o}^{(t-1)}\right), \quad (1)$$

where $\lambda_d, \lambda_s > 0$ are weighting coefficients. The first term of $\Delta(c_n, c_o)$ serves as the primary discriminative component, used to measure the similarity between prototypes of the old and new classes, with smaller value indicating greater confusion between them. The second term quantifies the extent of prototype drift for old classes caused by *Catastrophic Forgetting* during incremental training, with larger value indicating that the stability of the old class knowledge is more significantly affected by the new class. The third term is the trace of the covariance matrices, with larger value

indicating more blurred boundary. The computation of $\mu_{c_n}^{(t)}$, $\mu_{c_o}^{(t)}$ and $\Sigma_{c_n}^{(t)}$ in Equation 1 requires a preliminary warm-up stage prior to the commencement of the current incremental training task t. Specifically, the initial model is pre-trained using $\mathcal{D}_t$ and $\mathcal{M}_{t-1}$, with a small learning rate and a limited number of training epochs.

After obtaining the $\Delta(c_n, c_o)$ for for all new–old class pairs, we select the top K pairs with the lowest values, and K is set to the number of old classes in $\mathcal{M}_{t-1}$. For each selected class pair (c_n, c_o), we randomly select m exemplars of old class c_o from $\mathcal{M}_{t-1}$, and m samples of new class c_n from $\mathcal{D}_t$. These samples are paired one-to-one, resulting in m similar sample pairs, which are used to synthesis counterfactual samples for these two classes. To fully utilize the limited number of exemplars, we set m to the number of exemplars for each old class.

Counterfactual sample Synthesis. For each identified similar sample pair (x_n, x_o), we first perform Cut-Mix (Yun et al. 2019) to obtain the mixed image x_{mix} by blending x_o and x_n in a 3:1 ratio, which incorporates local features from both images. Then, we perform latent space nonlinear interpolation. Specifically, we first use a pre-trained VQ-VAE (Van Den Oord, Vinyals et al. 2017) encoder to obtain the latent representations of the original and the mixed images:

$$\mathbf{z}_n = Enc(x_n), \mathbf{z}_o = Enc(x_o), \mathbf{z}_{mix} = Enc(x_{mix}). \quad (2)$$

Subsequently, we synthesize counterfactual feature vectors in the latent space:

$$\mathbf{z}_{cf} = \mathbf{z}_{mix} + \alpha(\mathbf{z}_n - \mathbf{z}_o) + \epsilon, \quad (3)$$

where $\alpha \sim \mathcal{U}(0,1)$, $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. The first term $\mathbf{z}_{mix}$ preserves high-frequency local cues, preventing the decoder from producing blurred or artifact-prone reconstructions. The second term $\alpha(\mathbf{z}_n - \mathbf{z}_o)$ represents linear interpolation along the new-old feature vector direction, which conducts controlled traversal along the inter-class semantic vector to densely populate the decision boundary. The third term ϵ represents a nonlinear random perturbation, which injects stochastic deviations around this manifold, enriching sample diversity and smoothing the boundary to enhance classifier generalization.

In general, this synthesis method build a more rich and asymmetric transition region between two classes, enhancing the representativeness and diversity of counterfactual samples, thereby improving the model's generalization and ultimately optimizing the discrimination of the classification boundary. The counterfactual synthetic image is obtained by decoding the latent vector:

$$\tilde{x}_{cf} = Dec(\mathbf{z}_{cf}), \quad (4)$$

and we define $\tilde{\mathcal{X}}_{cf} = \{\tilde{x}_{cf}\}$ as the set of all counterfactual synthetic samples.

Boundary Augmentation Loss. For any synthesized sample $\tilde{x}_{cf}$ situated near the boundary between class c_n and c_o, we expect the model to remain uncertain about its classification, *i.e.*, to assign high probabilities to both c_n and c_o, thereby expanding the decision boundary with them. We assign a symmetric soft label y_{cf}^* to $\tilde{x}_{cf}$:

$$y_{cf}^*(c_n) = y_{cf}^*(c_o) = \frac{1}{2}, \quad y_{cf}^*(c) = 0 \ (c \notin \{c_n, c_o\}). \quad (5)$$

Then we minimize the cross-entropy between the model's prediction $\hat{p}(\cdot|\tilde{x}_{cf})$ and the soft label y_{cf}^*, and design a boundary augmentation loss function $\mathcal{L}_{\text{CBA}}$:

$$\begin{aligned}\mathcal{L}_{\text{CBA}}(\tilde{x}_{cf}) &= -\sum_{c\in\{c_n,c_o\}} y_{cf}^*(c)\log p(c \mid \tilde{x}_{cf}) \\ &= -\frac{1}{2}\big[\log p(c_n \mid \tilde{x}_{cf}) + \log p(c_o \mid \tilde{x}_{cf})\big],\end{aligned} \quad (6)$$

which can be interpreted as maximizing the entropy of the model's prediction at x_{cf}, thereby preventing overly confident decisions near the boundary. In this context, by imposing this uncertainty-promoting constraint on a large number of counterfactual samples located near the decision boundaries of every easily confused class pairs, the decision boundaries are effectively stretched, leading to an increased margin between new and old classes. Therefore, the proposed counterfactual-based boundary augmentation can help alleviate the forgetting of previously learned knowledge to a certain extent.

Temporal Instability-Aware Weighting

During the incremental training, the necessity of each sample for the model to learn is not constant. Some samples can be quickly and consistently recognized correctly by the model, while others may exhibit persistent prediction fluctuations throughout training. Assigning equal loss weights to both types of samples leads to inefficiencies: the former results in wasted gradient updates, while the latter suffers from under-fitting. To address this issue, we propose the Temporal Instability-Aware Weighting (TIAW) method, which leverages temporal instability, a unique aspect of incremental learning, to dynamically adjust the training focus based on this metric. The core idea is to identify "hard samples" in the current stage and then increase the learning emphasis on these samples in subsequent stages to improve overall training effectiveness.

Definition of Temporal Instability. Consider a sample x from the current training dataset composed of $\mathcal{D}_t$, $\mathcal{M}_{t-1}$ and the counterfactual synthetic dataset $\tilde{\mathcal{X}}_{cf}^t$ in task t. we track the SoftMax output distribution of x over training epochs using a sliding window to quantify its "Temporal Instability". Specifically, we maintain a sliding window of length w that stores the model's output probability distributions for x over the most recent w epochs (denoted as $p_\tau(x)$). By examining these consecutive historical distributions, we can characterize the fluctuations in the model's predictions for sample x over time.

Calculation of Instability score. We decompose the temporal instability of x at epoch i into two components: the *current prediction uncertainty* and the *historical prediction fluctuation*.

First, the *current prediction uncertainty* is measured by the information entropy $H_i(x)$ of the output distribution at epoch i. It is defined as:

$$H_i(x) = -\sum_{c\in\mathcal{C}_i} p_i(c \mid x)\log p_i(c \mid x), \quad (7)$$

where $p_i(c \mid x)$ denotes the model's predicted probability of class c for sample x at epoch i, and $\mathcal{C}_t$ represents all currently seen classes.

Second, the *historical prediction fluctuation* $B_i(x)$ quantifies the difference between the current prediction and recent predictions. We employ the Jensen–Shannon Divergence (JSD) to measure the difference between two probability distributions, and define $B_i(x)$ as the average JSD between the current distribution and all distributions in the sliding window:

$$B_i(x) = \frac{1}{|W_i|} \sum_{\tau \in W_i} \mathrm{JSD}\big(p_i(x) \big\| p_\tau(x)\big), \tag{8}$$

where $\mathcal{W}_i = \{i-w, \ldots, i-1\}$ denotes the set of indices of the w most recent epochs preceding epoch i (if $i < w$, then $\mathcal{W}_i = \{1, 2, \ldots, i-1\}$).

According to $H_i(x)$ and $B_i(x)$, we define the overall instability score of sample x at epoch i as their weighted sum:

$$s_i(x) = \lambda_t \cdot H_i(x) + (1 - \lambda_t) \cdot B_i(x), \tag{9}$$

where $\lambda_t \in [0, 1]$ is a hyperparameter that balances the current uncertainty and the historical fluctuation. Since the numerical range of $s_i(x)$ varies depending on the dataset, model, and hyperparameters, we apply Min-Max Normalization to convert it into a standardized instability score:

$$\hat{s}_i(x) = \frac{s_i(x) - s_{\min,i}}{s_{\max,i} - s_{\min,i} + \varepsilon}, \tag{10}$$

where $s_{\max,i}$ and $s_{\min,i}$ represent the maximum and minimum values of $s_i(x)$ across all samples in the current mini-batch, and ε is a small regularization constant.

Adaptive Classification Loss Weighting. In order to dynamically adjust the loss weight of sample x according to its instability level, we maps the instability score to a gradient weight factor $\omega_i(x)$:

$$\omega_i(x) = 1 - (1 - \omega_{\min}) e^{-\beta_k \cdot \hat{s}_i(x)}, \tag{11}$$

where $\omega_{\min}$ is a constant that defines the lower bound of the sample loss weight, while $\beta_k \in \mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_K\}$ represents the exponential slope, which is adjusted dynamically according to current learning rate. The value range of $\omega_i(x)$ is $[\omega_{\min}, 1)$. We apply this factor to weight the loss term of sample x, thereby adjusting its contribution to the overall gradient:

$$\mathcal{L}_{\mathrm{TIAW}}(x) = \frac{1}{N} \sum_{j=1}^{N} \Big[\omega_i(x_j) \cdot \mathcal{L}_{\mathrm{cls}}\big(f_\theta(x_j),\, y_j\big)\Big] \tag{12}$$

where $\mathcal{L}_{\mathrm{cls}}$ represents the normal classification loss (e.g., cross-entropy), and N is the number of samples in the current mini-batch.

Notably, TIAW can further improve the effectiveness of the CBA method and enhance the utility of exemplars in replay-based approaches. On one hand, the counterfactual samples generated by CBA inherently exhibit high entropy $H_i(x)$ in the label space, indicating their boundary-like nature. On the other hand, due to the small proportion in the training data, exemplars from old classes are highly vulnerable to gradient interference from new classes, often resulting in more pronounced historical prediction fluctuation, *i.e.*, a higher expected value of $B_i(x)$. By increasing the gradient weights of these two types of samples, TIAW demonstrates unique advantages in enhancing decision boundaries and mitigating catastrophic forgetting.

Algorithm 1: Training with CBA & TIAW

Require: New data $\mathcal{D}_t$, memory $\mathcal{M}_{t-1}$, model parameters θ

$\widetilde{\mathcal{X}}_{cf} \leftarrow$ **CBA_Generate**$(\mathcal{D}_t, \mathcal{M}_{t-1})$:
  pick top-K confusable class pairs using $\Delta(c_n, c_o)$;
  CutMix (x_n, x_o) and latent interpolation$\rightarrow \tilde{x}$;
  collect $\tilde{x}$ (with soft label y^*) into $\widetilde{\mathcal{X}}_{cf}$
for epoch $i = 1$ **to** E **do**
  for all mini-batch $\mathcal{B} \subset (\mathcal{D}_t \cup \mathcal{M}_{t-1} \cup \widetilde{\mathcal{X}}_{cf,t})$ **do**
    Split $\mathcal{B}$ into real $\mathcal{B}_{\mathrm{r}}^{(i)}$ and counterfactual $\mathcal{B}_{\mathrm{cf}}^{(i)}$
    for all sample $x \in \mathcal{B}$ **do**
      $\hat{s}_i(x) \leftarrow$ **Temporal Instability**(x):
        append current pred p to buffer;
        $s_i(x) \leftarrow$ entropy $H_i(x)$ and average JSD $B_i(x)$;
        normalize $s_i(x)$ to $\hat{s}_i(x)$
      $\omega_i(x) \leftarrow 1 - (1 - \omega_{\min}) \exp\big(-\beta\, \hat{s}_i(x)\big)$
    end for
    Compute total loss $\mathcal{L}_{\text{total}}$ via Equation (13)
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{total}}$
  end for
end for
$\mathcal{M}_t \leftarrow$ **UpdateMemory**$(\mathcal{D}_t, \mathcal{M}_{t-1})$

Integrated Objective

We integrate the CBA and TIAW modules to formulate the integrated optimization objective of the model. At the i-th epoch of incremental training task t, for each real sample x from $\mathcal{D}_t$ and $\mathcal{M}_{t-1}$ in the current mini-batch, its classification loss is dynamically scaled according to the TIAW weighting factor $\omega_i(x)$, as defined in Equation (12). For each counterfactual sample $\tilde{x}$ synthesized by CBA module, we calculate its boundary augmentation loss using Equation (6), and then also apply TIAW to reweight the loss accordingly.

The overall loss at each training step is then given by:

$$\begin{aligned} \mathcal{L}_{\text{total}}^{(i)} = \frac{1}{N_{\mathrm{r}}} \sum_{k=1}^{N_{\mathrm{r}}} \Big[\omega_i(x_k) \cdot \mathcal{L}_{\mathrm{cls}}\big(f_\theta(x_k),\, y_k\big)\Big] \\ + \lambda \cdot \frac{1}{N_{\mathrm{cf}}} \sum_{j=1}^{N_{\mathrm{cf}}} \Big[\omega_i(\tilde{x}_j)) \cdot \mathcal{L}_{\mathrm{cls}}(f_\theta(\tilde{x}_j), y_j^*)\Big] \end{aligned} \tag{13}$$

where N_{r} is the number of real samples in current mini-batch, N_{cf} is that of counterfactual samples, and λ is a hyperparameter that balances the contribution of the CBA term. As shown in Equation (13), CBA and TIAW, as plug-in CIL strategies, introduce no modifications to the original model architecture or loss design. The complete procedure is summarized in the pseudocode of Algorithm 1.

Method	CIFAR100			ImageNet-Subset			ImageNet-Full	
	$T=5$	$T=10$	$T=25$	$T=5$	$T=10$	$T=25$	$T=5$	$T=10$
Upper Bound	73.24	73.56	73.77	83.21	82.53	82.82	70.81	69.88
iCaRL	57.12	52.66	48.22	65.44	59.88	52.97	51.50	46.89
BiC	59.36	54.20	50.00	70.07	64.96	57.73	62.65	58.72
Mnemonics	63.34	62.28	60.96	72.58	71.37	69.74	64.63	63.01
TPCIL	65.34	63.58	-	76.27	74.01	-	64.89	62.88
GeoDL(+LUCIR)	65.14	65.03	63.12	73.87	73.55	71.72	65.23	64.46
AANet(+PODNet)	66.31	64.31	62.31	76.96	75.58	71.78	67.73	64.85
LUCIR	63.17	60.14	57.54	70.84	68.32	61.44	64.45	61.57
LUCIR*	63.32	61.07	57.83	70.91	67.90	62.06	66.30	63.22
***w/* CBA+TIAW**	**65.11** ↑1.79	**63.20** ↑2.13	**59.92** ↑2.09	**72.68** ↑1.77	**69.53** ↑1.63	**65.03** ↑2.97	**66.97** ↑0.67	**64.64** ↑1.42
PODNet	64.83	63.19	60.72	75.54	74.33	68.31	66.95	64.13
PODNet*	64.59	62.88	60.11	75.05	73.84	67.24	66.11	63.78
***w/* CBA+TIAW**	**65.83** ↑1.24	**64.67** ↑1.79	**61.74** ↑1.63	**75.90** ↑0.85	**75.06** ↑1.22	**69.06** ↑1.82	**67.32** ↑1.21	**64.89** ↑1.11
AFC(CNN)	66.49	64.98	63.89	76.87	75.75	73.34	68.90	67.02
AFC(CNN)*	66.02	64.37	62.50	76.94	75.29	72.88	69.01	66.96
***w/* CBA+TIAW**	**67.16** ↑1.14	**65.42** ↑1.05	**63.49** ↑0.99	**77.25** ↑0.31	**75.96** ↑0.67	**73.56** ↑0.68	**70.03** ↑1.02	**67.78** ↑0.82

Table 1: The *average incremental accuracy* (%) of different methods on CIFAR100, ImageNet-Subset and ImageNet-Full under the T-task setting. Models marked with an asterisk (*) denote results reproduced using the official code. Bold numbers indicate the best results under the same baseline and benchmark. The symbol ↑ followed by a number indicates the performance gain achieved by integrating our proposed methods (CBA + TIAW).

Experiment

Experimental Settings

Datasets. We conduct experiments on three widely adopted benchmark datasets in CIL. **CIFAR100** (Krizhevsky, Hinton et al. 2009) is a medium-scale natural image dataset consisting of 100 classes, each containing 500 training images and 100 test images of size 32 × 32. **ImageNet-Full** (Russakovsky et al. 2015) is a large-scale dataset consisting of 1000 classes, each containing about 1250 training images and 50 test images of size 224 × 224. **ImageNet-Subset** is constructed by 100 selected classes from ImageNet-Full with random seed 1993 (Rebuffi et al. 2017).

Baselines and Benchmarks. We evaluate our proposed CBA and TIAW modules integrated into three representative baselines, LUCIR (Hou et al. 2019), PODNet (Douillard et al. 2020), and AFC (Kang, Park, and Han 2022), and compare their performance with several state-of-the-art CIL methods, including iCaRL (Rebuffi et al. 2017), BiC (Wu et al. 2019), Mnemonics (Liu et al. 2020), TPCIL (Tao et al. 2020), GeoDL (Simon, Koniusz, and Harandi 2021), and AANet (Liu, Schiele, and Sun 2021).

We adopt the widely used "B-half Inc-T" protocol, where the learning process begins with an initial task that includes half of the total classes for training. The remaining classes are then uniformly divided into T groups, each introduced sequentially in T incremental steps (denoted as "T tasks"). For CIFAR100 and ImageNet-Subset, we evaluate the model under three different task configurations of $T = \{5, 10, 25\}$, while two configurations of $T = \{5, 10\}$ for ImageNet-Full.

Evaluation Metric. At each incremental stage, we evaluate the model using the test samples from all classes learned fo far. The final performance is reported as the average accuracy over all stages, a metric commonly referred to as the *average incremental accuracy* (Rebuffi et al. 2017).

Implementation Details. To ensure a fair comparison and accurately assess the effectiveness of our methods, we follow all shared implementation setting of the four baselines we chose. First, the hyper-parameters, including the number of epochs, batch size, learning rate, etc. Then, the choice of backbone: ResNet-32 for CIFAR100, and ResNet-18 for ImageNet-Subset and ImageNet-Full. Finally, the herding-based exemplar selection strategy and the fixed number of 20 exemplars per learned class.

The hyper-parameters and implementation setting related to CBA and TIAW are as follows. Specifically, For the CBA module, we set the prototype–drift weight λ_d to 0.5, 0.5 and 1.0 for the training on CIFAR100, ImageNet-Subset and ImageNet-Full, respectively, and set the covariance–trace weight λ_s to 1.0, 0.5 and 0.5. The latent Gaussian perturbation ϵ adopts 0.1, 0.2 and 0.2 for the three datesets, respectively, following the order listed above. For the warm-up process, we train 30 epochs with a smaller learning rate of 0.01. For the TIAW module, we set the sliding window length w to 5 and the balance factor λ_t to 0.5 for all datasets. The minimum loss weight $\omega_{\min}$ is kept at 0.3 for all datasets to retain at least 30% of the gradient on easy samples. The exponential slope β_k is adjusted in accordance with the learning rate decay schedule, with $\mathcal{B}$ set to $\{1, 3, 5\}$ during training on CIFAR100, and to $\{1, 5, 8\}$ on ImageNet-Subset and ImageNet-Full. The boundary augmentation loss weight $\mathcal{L}_{\text{CBA}}$ is set to 1.0 for CIFAR100, and to 2.0 for the other two datasets.

Method	CIFAR100		
	$T = 5$	$T = 10$	$T = 25$
LUCIR*	63.32	61.07	57.83
LUCIR* + CBA	64.68	62.71	59.28
LUCIR* + TIAW	63.77	61.39	58.14
LUCIR* + CBA + TIAW	**65.11**	**63.20**	**59.92**

Table 2: Ablative results of CBA and TIAW on CIFAR100 using LUCIR as the baseline. We report the *average incremental accuracy* (%), and LUCIR* denotes our reproduced results.

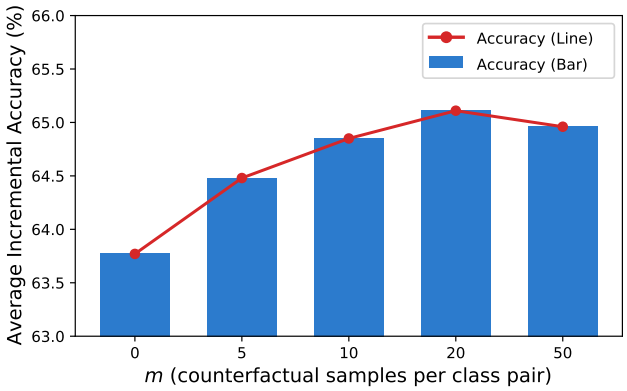

Figure 3: Effect of varying the number of counterfactual samples per class pair on CIFAR100 (with 5-task, LUCIR baseline).

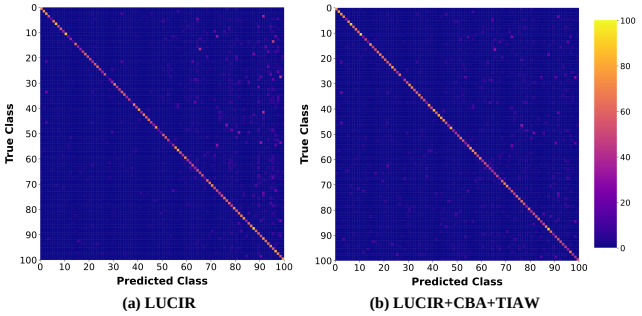

Figure 4: Confusion matrices of the baseline LUCIR and LUCIR combined with CBA and TIAW on CIFAR100 under the 5-task setting.

Performance Experiments

Results on CIFAR100. As shown in Table 1, The combination of integrating CBA+TIAW into AFC achieves the highest average incremental accuracies of 67.16%, 65.42%, and 63.49% under the 5-, 10-, and 25-task incremental settings on CIFAR100, surpassing all compared methods. Meanwhile, incorporating CBA and TIAW yields consistent performance gains for all baseline models. For instance, our approach improves the *average incremental accuracy* of LUCIR by 1.79%, 2.13%, and 2.09% under these three settings, respectively. For PODNet, the improvements are 1.24%, 1.79%, and 1.63%. Even for the best-performing baseline AFC, an additional gain of around 1% is achieved. Notably, in the challenging 25-task scenario, our method still maintains a significant advantage, demonstrating robust performance in long-sequence increments.

Results on ImageNet-Subset. ImageNet-Subset contains a larger number of higher-resolution images than CIFAR100, making it more complex and challenging. In this scenario, methods like PODNet and AFC that employ multi-proxy classifiers exhibit strong baseline performance. As shown in Table 1, our approaches (with AFC) obtain the best performance of 77.25%, 75.96% and 73.56% respectively. Furthermore, CBA and TIAW deliver performance gains across all baseline methods: for LUCIR, the accuracy increases by 1.77%, 1.63%, 2.97% in three cases, respectively. PODNet sees improvements of 0.85%, 1.22%, 1.82%. and AFC is also boosted by an additional 0.31%–0.68%.

Results on ImageNet-Full. ImageNet-Full is the most demanding benchmark. As shown in Table 1, The combination with AFC attains the best results of 70.03% and 67.78% under the 5-task and 10-task settings, respectively, substantially outperforming the other methods. Meanwhile, our approach still provides noticeable performance improvements for baselines in this large-scale scenario. We improve the accuracy of LUCIR by 0.67% and 1.42% for 5- and 10-task settings, respectively, and boost the performance of PODNet by 1.21% and 1.11%.

Ablation study and Analysis

Effects of the CBA module. CBA consistently improves model performance across a variety of incremental learning settings, as shown in Table 2. With LUCIR as the baseline on CIFAR100, integrating CBA improves the average incremental accuracy from 63.32% to 64.68% under the 5-task setting, with comparable absolute gains of approximately 1.5% also observed in the 10-task (61.07% to 62.71%) and 25-task (57.83% to 59.28%) settings. Such improvements indicate the general efficacy of CBA in mitigating the forgetting problem.

Moreover, we evaluate the effect of varying the number m of counterfactual samples synthesized by CBA. As shown in Figure 3, increasing m from 0 to 20 leads to a consistent improvement in accuracy, demonstrating the effectiveness of CBA in expanding decision boundaries and mitigating misclassification between confusable classes. However, further increasing m introduces performance degradation. This is because excessive synthetic samples, especially when surpassing the number of real exemplars, provide diminishing gains in boundary discriminability, while injecting redundancy and overfitting risk, which ultimately impairs the model's generalization to real data.

Effects of the TIAW module. Compared to CBA, TIAW alone offers limited performance gains (typically less than 0.5%). However, its core contribution lies in its temporal instability-aware reweighting mechanism, which increases the weight of samples with high prediction fluctuation and entropy. Without the presence of abundant high-uncertainty

counterfactual samples generated by CBA, the effectiveness of TIAW is inherently constrained. Thus, its full potential is best realized when used in conjunction with CBA, where the two modules synergistically amplify boundary information and reinforce old class knowledge retention.

Combined Effect Analysis. As shown in Table 2, the combination of CBA and TIAW yields a clear synergistic effect. Across all evaluation settings, "LUCIR+CBA+TIA" consistently achieves the highest average incremental accuracy, with performance gains that are comparable to or even slightly surpass the sum of their individual contributions. Mechanistically, CBA enhances inter-class discriminability by explicitly expanding decision boundaries and generating hard counterfactual samples near class borders, while TIAW reinforces learning on these samples by assigning greater loss weights to those with high temporal instability. The two modules are complementary: CBA supplies broader margins and valuable training samples, whereas TIAW ensures their effective exploitation, jointly mitigating class confusion and alleviating catastrophic forgetting. Figure 4 presents the visualized confusion matrices, clearly demonstrating the improvement in reducing inter-class confusion achieved by integrating CBA and TIAW into the baseline model.

Conclusion

References

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.-W.; and Choi, J. 2021. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8218–8227.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning*, 20(3): 273–297.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, 86–102. Springer.

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.

Gao, R.; and Liu, W. 2023. Ddgr: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, 10744–10763. PMLR.

Gao, Z.; Han, S.; Zhang, X.; Xu, K.; Zhou, D.; Mao, X.; Dou, Y.; and Wang, H. 2025. Maintaining fairness in logit-based knowledge distillation for class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 16763–16771.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 831–839.

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.

Kim, T.; Park, J.; and Han, B. 2024. Cross-class feature augmentation for class incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13168–13176.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Liu, Y.; Schiele, B.; and Sun, Q. 2021. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2544–2553.

Liu, Y.; Su, Y.; Liu, A.-A.; Schiele, B.; and Sun, Q. 2020. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 12245–12254.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Murphy, G. L.; and Medin, D. L. 1985. The role of theories in conceptual coherence. *Psychological review*, 92(3): 289.

Nori, M. K.; Kim, I.-M.; and Wang, G. 2025. Autoencoder-Based Hybrid Replay for Class-Incremental Learning. *arXiv preprint arXiv:2505.05926*.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Simon, C.; Koniusz, P.; and Harandi, M. 2021. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 1591–1600.

Tao, X.; Chang, X.; Hong, X.; Wei, X.; and Gong, Y. 2020. Topology-preserving class-incremental learning. In *European conference on computer vision*, 254–270. Springer.

Van de Ven, G. M.; Siegelmann, H. T.; and Tolias, A. S. 2020. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1): 4069.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Wei, K.; Deng, C.; Yang, X.; and Li, M. 2021. Incremental embedding learning via zero-shot translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10254–10262.

Wen, H.; Pan, L.; Dai, Y.; Qiu, H.; Wang, L.; Wu, Q.; and Li, H. 2024. Class incremental learning with multi-teacher distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 28443–28452.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 374–382.

Xiang, Y.; Fu, Y.; Ji, P.; and Huang, H. 2019. Incremental learning using conditional adversarial networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6619–6628.

Yan, S.; Xie, J.; and He, X. 2021. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3014–3023.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6023–6032.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13208–13217.

Zheng, B.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2024. Multi-layer rehearsal feature augmentation for class-incremental learning. In *Forty-first international conference on machine learning*.

Zheng, B.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2025. Task-Agnostic Guided Feature Expansion for Class-Incremental Learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 10099–10109.

Zhou, D.-W.; Ye, H.-J.; Ma, L.; Xie, D.; Pu, S.; and Zhan, D.-C. 2022. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11): 12816–12831.