ARE WE FORGETTING OR JUST CONFUSED?

Anonymous authors
Paper under double-blind review

ABSTRACT

Class incremental learning enables models to continuously learn new categories sequentially. Nevertheless, mainstream incremental learning approaches follow a disjoint stage-wise training pipeline and inevitably suffer from severe **inter-stage class confusion** across different learning phases. To address this issue, this paper presents a **C**ross-stage **C**onfusion **S**uppression (**CCS**) framework built upon pre-trained vision models. The framework first develops a Confusion-Aware Connected Component Generation module, which leverages the Union-Find algorithm to dynamically cluster categories with high inter-stage confusion. On this basis, a Connected Component-Guided Differential Adapter mechanism is proposed to alleviate feature confusion induced by isolated task-specific feature spaces. In particular, we learn a shared adapter to capture common feature patterns within each confused class group, and further subtract the shared feature response from task-specific adapters to obtain more discriminative incremental representations. Moreover, to cope with the inherent unavailability of historical data in exemplar-free incremental learning, we develop a Noise-Driven Pseudo-Sample Generation strategy to compensate for the missing prior category information. Finally, we adopt a two-stage inference procedure to adaptively activate differential adapters during testing, which effectively bridges the gap between training and inference. Extensive experiments on four standard benchmarks verify that the proposed CCS method significantly reduces inter-stage category confusion and achieves superior performance in both classification accuracy and forgetting mitigation. Code will be made publicly available.

1 INTRODUCTION

Real-world visual systems process streaming data instead of a static corpus Wang & Chen (2021); Zhou et al. (2024), motivating class-incremental learning (CIL) that learns new categories sequentially while retaining old-class knowledge. A key obstacle for CIL is catastrophic forgetting, i.e., severe performance degradation on previous categories after learning new tasks Sun et al. (2022); Wu et al. (2025); Lai et al. (2026). Pre-trained models (PTMs) trained on large-scale corpora exhibit strong generalization and transfer capabilities, opening new avenues for incremental learning Zhang et al. (2026a). Accordingly, building robust incremental frameworks upon PTM priors has become a popular direction to mitigate forgetting. Most existing PTM-based CIL methods freeze the pre-trained backbone to preserve its general representations, and introduce lightweight modules for task adaptation. Representative prompt-based approaches such as L2P Wang et al. (2022c), DualPrompt Wang et al. (2022b) and CODA-Prompt Smith et al. (2023), learn task-aware prompts to capture task-specific semantics. As another dominant paradigm, adapter-based methods fine-tune PTMs through lightweight plug-in adapters and generally outperform prompt-based solutions Lu et al. (2026a); Sun et al. (2025b); Zhao et al. (2026); Deng et al. (2026).

Despite their promising performance, existing adapter-based methods suffer from a critical inherent limitation. Adapters trained at individual incremental stages exclusively focus on fitting the category features of the current task, while lacking a dedicated mechanism to suppress feature ambiguity across different stages. This leads to severe *inter-stage class confusion*, i.e., when prototypes of semantically similar classes from distinct learning stages are closely distributed in the embedding space, the model inevitably suffers from ambiguous inference and degraded classification accuracy. We illustrate this issue with a two-stage incremental learning example in Figure 1. The task-specific adapters A_1 and A_2 are trained for Task 1 and Task 2 separately, producing category prototypes

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

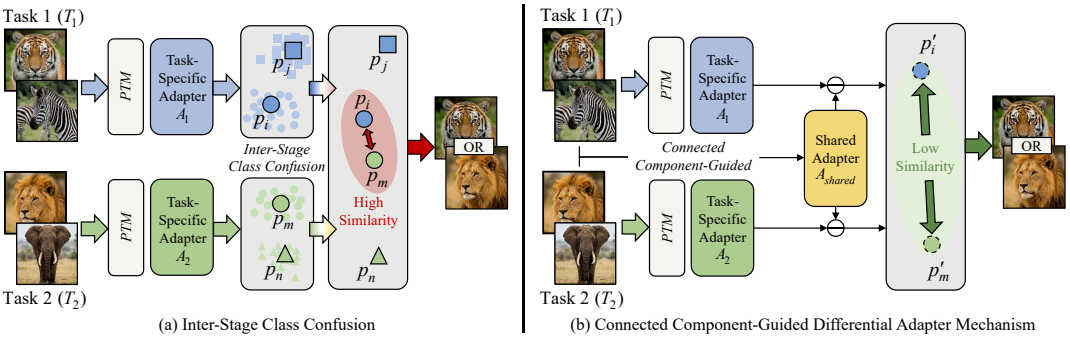


Figure 1: Inter-stage class confusion in CIL. (a) Similar cross-task prototypes (*e.g.*, tiger vs. lion) cause feature ambiguity. (b) Our differential adapter achieves discriminability enhancement.

for their corresponding classes. Although prototypes within a single task are well-separated, cross-task prototype discrimination is not guaranteed. For example, the tiger prototype p_i learned in Task 1 and the lion prototype p_m learned in Task 2 may maintain high embedding similarity. During inference, test samples of these two categories yield feature representations highly correlated with both prototypes, resulting in frequent misclassification.

Existing adapter-based methods typically mitigate inter-stage class confusion via rehearsal, yet the dependence on historical samples contradicts the exemplar-free learning principle and brings potential privacy risks. Meanwhile, limited preserved samples lead to biased feature optimization and weak cross-stage discriminative capability. To address these limitations, this paper proposes a novel Cross-stage Confusion Suppression (CCS) framework for exemplar-free class-incremental learning. We first design a **Confusion-Aware Connected Component Generation (CACG)** module that models class prototypes as graph nodes and identifies confused class groups via the Union-Find algorithm. To resolve feature ambiguity stemming from isolated task-specific spaces, we further present the **Connected Component-Guided Differential Adapter (CCDA)** mechanism, which learns component-level shared adapters to capture common patterns among confused classes and refines task-specific representations via feature difference calibration. Besides, we develop a **Noise-Driven Pseudo-Sample Generation (NPG)** strategy to compensate for the absence of historical exemplars, providing auxiliary supervision for robust shared adapter training and cross-stage confusion reduction. Finally, a two-stage inference pipeline is introduced for adaptive adapter activation, building a coherent training-inference optimization paradigm.

Our contributions are summarized as follows:

- We propose a novel PTM-based CCS framework for exemplar-free class-incremental learning. It effectively alleviates inter-stage class confusion and avoids the privacy and imbalance issues caused by historical sample rehearsal.
- A Connected Component-Guided Differential Adapter mechanism is designed to capture shared feature patterns of confused classes. The differential calibration operation explicitly enhances cross-stage feature discriminability and mitigates semantic ambiguity.
- We develop a Noise-Driven Pseudo-Sample Generation strategy to compensate for inaccessible historical data under the exemplar-free setting. The generated pseudo-samples supervise shared adapter training to robustly suppress cross-stage class confusion.
- Extensive experiments on four standard benchmarks demonstrate that the proposed CCS framework outperforms existing state-of-the-art methods on three datasets.

2 Related Work

Class-Incremental Learning (CIL). CIL targets continuous acquisition of novel visual categories while alleviating catastrophic forgetting of previously learned knowledge Zhou et al. (2025b; 2026); Li et al. (2026); Huang et al. (2026). Existing CIL solutions adopt diverse technical routes to combat catastrophic forgetting. Distillation-based methods Snell et al. (2017); Rebuffi et al. (2017); Deng et al. (2025); Bao et al. (2025) transfer discriminative knowledge from teacher to student models.

Replay-based strategies Li & Hoiem (2017); Wang et al. (2022a); Liu et al. (2024); Li et al. (2025) preserve past knowledge by replaying buffered historical exemplars. Regularization-based approaches Zhu et al. (2022); Wang et al. (2023); Zhai et al. (2024); Zhang et al. (2026b) constrain parameter or feature-manifold updates to avoid erasing learned information. To rectify spurious inductive bias accumulated throughout incremental training, rectification-based methods Wu et al. (2019); Yu et al. (2020); Zhao et al. (2020); Ahn et al. (2021); Pernici et al. (2021); Zhuang et al. (2022) calibrate spurious inductive bias accumulated over incremental phases, while expansion-based methods Douillard et al. (2022); Hu et al. (2023); Lu et al. (2026c) deploy task-specific subnetworks for knowledge isolation and ensemble them for inference-time multi-class calibration.

From the perspective of historical-data accessibility, exemplar-based CIL Hou et al. (2019); Ahmed et al. (2024); Yu et al. (2024); Qi et al. (2026) achieves decent performance at the cost of storing real old-class samples, which raises privacy concerns and violates incremental-learning desiderata Goswami et al. (2023); Xiao et al. (2025). This motivates exemplar-free CIL He et al. (2025); Jiang et al. (2026a); Yang et al. (2026) that learns new classes without real historical data. Representative works include weight-regularized EWC Kirkpatrick et al. (2017) and frozen-backbone FeTrIL Petit et al. (2023). A separate line of work combats representation drift, where ADC Goswami et al. (2024), LDC Gomez-Villa et al. (2024), and RoSE Lu et al. (2026b) mitigate feature-space, backbone, and semantic drift respectively. Considering the exemplar-free setting, we develop a Noise-Driven Pseudo-Sample Generation strategy in this paper. We involve the generated pseudo-samples in training to mitigate the class prototype confusion across incremental phases.

CIL with Pre-trained Models. Pre-trained models (PTMs) Han et al. (2021); Radford et al. (2021); Jiang et al. (2026b) are widely deployed in CIL under a freeze-backbone setting, with only lightweight modules updated. Prompt-based methods steer frozen PTMs via task-specific instructions. For example, L2P Wang et al. (2022c) leverages instance-specific prompts, DualPrompt Wang et al. (2022b) disentangles general and expert prompt branches, and CODA-Prompt Smith et al. (2023) performs prompt decomposition with attention weighting. Beyond prompt tuning as one popular lightweight paradigm, adapter-based fine-tuning represents another competitive line for CIL Lu et al. (2026a). Early efforts such as EASE Zhou et al. (2024) assign separate adapters for every incremental task. To retain old knowledge, MOS Sun et al. (2025b) merges task-specific adapters after model training, yet this practice brings growing parameter overhead with accumulating tasks. Further, RSIAT Zhao et al. (2026) adopts a single shared adapter and leverages an autoencoder-based projector to realize cross-task feature alignment. Nevertheless, embedding task-specific prototypes into a shared latent space inherently produces semantic overlap and inter-task confusion. To solve this, we introduce a Component-Guided Differential Adapter mechanism.

3 METHOD

To alleviate inter-stage class confusion, this work proposes the Cross-stage Confusion Suppression (CCS) framework, as visualized in Figure 2. CCS first detects mutually confusable cross-stage classes and learns group-level shared adapters to capture their intrinsic common feature patterns. To enable robust optimization under the exemplar-free setting, noise-driven pseudo-samples are leveraged to supervise shared adapter training. Finally, a two-stage inference strategy is adopted to refine model predictions.

3.1 PRELIMINARIES

Class Incremental Learning. CIL learns a sequence of T tasks with training sets $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$. Each dataset $\mathcal{D}_t = \{(x_k, y_k)\}_{k=1}^{|\mathcal{D}_t|}$ corresponds to task t with a disjoint label space Y_t, such that $Y_i \cap Y_j = \emptyset$ for $i \neq j$. $\mathcal{D}_t$ is only accessible during stage t. The objective is to optimize a classifier $f : \mathbb{R}^D \to \mathcal{Y}^t$ that minimizes the risk over the accumulated label space $\mathcal{Y}^t = \bigcup_{i=1}^{t} Y_i$.

PTM-Based CIL. The impressive downstream performance of PTMs has motivated researchers to explore their application in CIL tasks. These methods typically freeze the pre-trained backbone to maintain general knowledge and mitigate catastrophic forgetting. Following this paradigm, a frozen PTM backbone is utilized as the foundation, denoted as $\phi : \mathbb{R}^D \to \mathbb{R}^d$. Task-specific adaptation is achieved by introducing lightweight trainable modules Λ (e.g., adapters), which modulate the feature extraction process to produce $\phi(x; \Lambda)$.

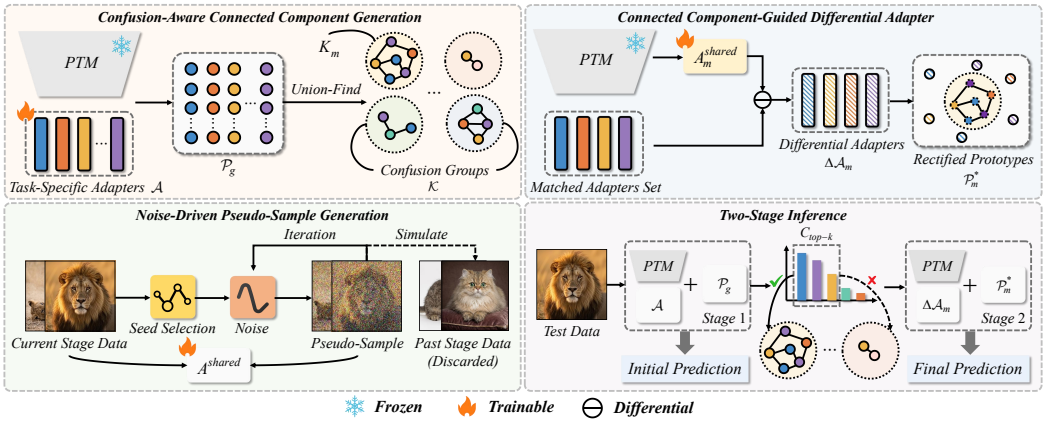


Figure 2: Overview of the proposed CCS framework. **(1) CACG:** Prototypes form connected components via confusion relationships. **(2) CCDA:** Discriminative features are extracted by differencing task-specific and component-shared adapters. **(3) NPG:** Noise perturbation synthesizes exemplar-free data. **(4) Two-Stage Inference:** Coarse prediction is refined for confusing samples.

The final model is defined as $f(x) = W^\top \phi(x;\Lambda)$, where $W \in \mathbb{R}^{d\times|\mathcal{Y}^t|}$ represents the linear classifier weights. During training on $\mathcal{D}_t$, W and Λ are updated by minimizing the classification loss:

$$\mathcal{L}_{base} = \sum_{(x_k,y_k)\in\mathcal{D}_t} \ell\left(W^\top \phi(x_k;\Lambda), y_k\right), \tag{1}$$

where $\ell(\cdot,\cdot)$ denotes the cross-entropy loss. The backbone ϕ is frozen throughout to preserve general knowledge and mitigate catastrophic forgetting.

PTM-Based CIL with Adapter Tuning. When the trainable module Λ is instantiated as an adapter, it is typically inserted as a side branch to capture task-specific information. Specifically, an adapter is placed after the Multi-Layer Perceptron (MLP) in each transformer block. It consists of a down-projection layer $W_{down} \in \mathbb{R}^{d\times r}$, a non-linear activation $\sigma(\cdot)$, and an up-projection layer $W_{up} \in \mathbb{R}^{r\times d}$, where $r \ll d$ is the bottleneck dimension. The adapter enhances the original MLP output via a residual connection:

$$x_o = \text{MLP}(x_i) + \sigma(x_i W_{down})W_{up}, \tag{2}$$

where x_i and x_o denote the input and output of the module, respectively.

3.2 Confusion-Aware Connected Component Generation

To identify inter-stage class ambiguity, we propose **Confusion-Aware Connected Component Generation (CACG)**. For each new task t, we allocate a task-specific adapter A_t and a classifier head $W_t \in \mathbb{R}^{d\times|Y_t|}$. Keeping the backbone $\phi(\cdot)$ frozen, we optimize A_t and W_t by minimizing the classification loss on the task dataset $\mathcal{D}_t$:

$$\min_{A_t,W_t} \sum_{(x_k,y_k)\in\mathcal{D}_t} \ell\big(W_t^\top \phi(x_k;A_t), y_k\big), \tag{3}$$

where $\phi(x;A_t)$ denotes the adapted PTM embedding. After T tasks, we obtain a set of task-specific adapters $\mathcal{A} = \{A_1,\ldots,A_T\}$. For each class i in stage t, its prototype $p_{i,t}$ is defined as the expected feature representation over its data distribution $\mathcal{X}_i$:

$$p_{i,t} = \mathbb{E}_{x\sim\mathcal{X}_i}\left[\phi(x;A_t)\right]. \tag{4}$$

Let $P_t = \{p_{i,t}\}_{i=1}^{\mathcal{C}_t}$ be the prototype set for stage t, where $\mathcal{C}_t$ denotes the number of classes. The global prototype set across T stages is defined as $\mathcal{P}_g = \bigcup_{t=1}^{T} P_t$. We model inter-class relationships using an undirected graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, where vertices $\mathcal{V}$ represent prototypes in $\mathcal{P}_g$. The edge set $\mathcal{E}$ is defined by the similarity between prototypes from different stages. An edge $(i,t_1) \sim (j,t_2) \in \mathcal{E}$ exists only if $t_1 \neq t_2$ and their cosine similarity exceeds a threshold θ:

$$(i, t_1) \sim (j, t_2) \iff \frac{p_{i,t_1}^\top p_{j,t_2}}{\|p_{i,t_1}\|_2 \|p_{j,t_2}\|_2} \geq \theta. \tag{5}$$

Using the Union-Find algorithmTarjan & Endre (1975), we decompose $\mathcal{G}$ into disjoint connected components, denoted as $\mathcal{K} = \{K_1, \ldots, K_M\}$. Each component K_m containing at least two nodes ($|K_m| \geq 2$) forms a **Confusion Group**. These groups identify classes that cause inter-stage confusion, thereby establishing the foundation for our subsequent suppression mechanism.

3.3 Connected Component-Guided Differential Adapter

To address the cross-stage confusion caused by classes within each confusion group, we propose the **Connected Component-Guided Differential Adapter (CCDA)**. Specifically, since nodes represent class prototypes, a connected component $K_m \in \mathcal{K}$ is formalized as a set of class-stage tuples:

$$K_m = \left\{ (y^{(j)}, t^{(j)}) \mid j = 1, 2, \ldots, |K_m| \right\}, \tag{6}$$

where $|K_m| \geq 2$ denotes the total number of confused classes within this component. Here, $y^{(j)}$ represents the class label and $t^{(j)}$ indicates its incremental stage. Each class $y^{(j)}$ is mapped to its task-specific adapter $A_{t^{(j)}}$, giving the set $S_m = \{A_{t^{(j)}} \mid (y^{(j)}, t^{(j)}) \in K_m\}$.

To address cross-stage feature confusion, we train a *shared adapter* A_m^{shared} to capture common patterns among all classes in K_m. We then derive a *differential adapter* for each class $y^{(j)}$ by subtracting the shared component from its task-specific adapter:

$$\Delta A_{t^{(j)}} = A_{t^{(j)}} - A_m^{shared}. \tag{7}$$

The set of differential adapters for K_m is denoted as $\Delta\mathcal{A}_m = \{\Delta A_{t^{(j)}} \mid (y^{(j)}, t^{(j)}) \in K_m\}$. This element-wise subtraction acts as a high-pass semantic filter, suppressing similar information while preserving the unique, discriminative residuals specific to each task. During this phase, we freeze the original task-specific adapters $A_{t^{(j)}}$ and exclusively optimize the shared adapter A_m^{shared}.

To mitigate inter-stage confusion, we align the feature space with rectified prototypes. For a class $y^{(k)}$ from stage $t^{(k)}$, the original and rectified prototypes are defined as:

$$p_{y^{(k)},t^{(k)}} = \mathbb{E}_{x \sim \mathcal{X}_{y^{(k)}}} \left[\phi(x; A_{t^{(k)}}) \right], \tag{8}$$

$$p^*_{y^{(k)},t^{(k)}} = \mathbb{E}_{x \sim \mathcal{X}_{y^{(k)}}} \left[\phi(x; \Delta A_{t^{(k)}}) \right]. \tag{9}$$

Here, $p^*_{y^{(j)},t^{(j)}}$ is the prototype of class $y^{(k)}$ in the feature space transformed by the differential adapter $\Delta A_{t^{(k)}}$. The set of rectified prototypes for K_m is $\mathcal{P}_m^* = \{p^*_{y^{(j)},t^{(j)}} \mid (y^{(j)}, t^{(j)}) \in K_m\}$.

For a training sample x of class $y^{(j)}$ in K_m, we extract its feature embedding via the differential adapter as $z = \phi(x; \Delta A_{t^{(j)}})$. To enhance discriminability, we minimize a prototype-based contrastive loss in the rectified feature space:

$$\mathcal{L}_{diff} = -\sum_{(x,y) \in \mathcal{D}_{K_m}} \log \frac{\exp(\text{sim}(z, p^*_{y^{(j)},t^{(j)}})/\tau)}{\sum_{(y^{(k)},t^{(k)}) \in K_m} \exp(\text{sim}(z, p^*_{y^{(k)},t^{(k)}})/\tau)}, \tag{10}$$

where $\mathcal{D}_{K_m}$ is the union of samples for all classes in K_m, sim denotes cosine similarity, and τ is the temperature hyperparameter. By optimizing A_m^{shared} to minimize $\mathcal{L}_{diff}$, the model filters out common patterns. Consequently, the rectified features extracted by the matched differential adapters form distinct, non-overlapping clusters.

3.4 Noise-Driven Pseudo-Sample Generation

Optimizing the shared adapter via Eq. 10 requires training data $\mathcal{D}_{K_m}$ for all classes within the connected component K_m. However, under the exemplar-free setting, historical samples are discarded.

To bridge this gap, we propose a **Noise-Driven Pseudo-Sample Generation (NPG)** strategy to synthesize data for previous classes $K_m^{prev} = \{(y,t) \in K_m \mid t < T\}$ by perturbing samples from the current stage $K_m^{curr} = \{(y,t) \in K_m \mid t = T\}$.

Rather than initializing from random noise, we leverage the semantic richness of the current data. For a target historical prototype $p_{y^{(k)},t^{(k)}}$, we select N "seed" samples $\mathcal{S}_{seed}$ with the highest feature affinity to the target from the current stage data $\mathcal{D}_{K_m^{curr}}$:

$$\mathcal{S}_{seed} = \underset{\{x_i\} \subset \mathcal{D}_{K_m^{curr}}, |\{x_i\}|=N}{\arg\min} \sum_{x \in \{x_i\}} \|\phi(x; A_{t^{(k)}}) - p_{y^{(k)},t^{(k)}}\|_2. \tag{11}$$

For each selected seed sample $x \in \mathcal{S}_{seed}$, we aim to synthesize a pseudo-sample $\hat{x}$ by injecting a perturbation δ such that its feature representation aligns with the target prototype. We initialize $\hat{x}^{(0)} = x$ and optimize it to minimize the mean squared error in the feature space:

$$\mathcal{L}_{gen}(\hat{x}, p_{y^{(k)},t^{(k)}}) = \|\phi(\hat{x}; A_{t^{(k)}}) - p_{y^{(k)},t^{(k)}}\|_2^2. \tag{12}$$

To bridge the domain gap, we update the sample iteratively. In each iteration i, we compute the gradient of the loss with respect to the input $\hat{x}^{(i)}$ and apply a normalized gradient update step:

$$\hat{x}^{(i+1)} \leftarrow \hat{x}^{(i)} - \eta \frac{\nabla_{\hat{x}} \mathcal{L}_{gen}(\hat{x}^{(i)}, p_{y^{(k)},t^{(k)}})}{\|\nabla_{\hat{x}} \mathcal{L}_{gen}(\hat{x}^{(i)}, p_{y^{(k)},t^{(k)}})\|_2}, \tag{13}$$

where η is the step size controlling the magnitude of noise injection. After n iterations, the final $\hat{x}^{(n)}$ serves as the pseudo-sample for class $y^{(k)}$. This process allows us to populate the historical set K_m^{prev} with synthesized data, thus enabling the calculation of the suppression loss $\mathcal{L}_{diff}$ in Eq. 10.

3.5 Two-Stage Inference

To balance intra-stage stability and inter-stage discriminability, we propose an adaptive **two-stage inference** procedure, activating differential adapters only when cross-stage confusion occurs.

Stage 1: Global Coarse Prediction. For a test sample x, we first extract features using task-specific adapters $\mathcal{A} = \{A_t\}_{t=1}^T$ and compute cosine similarities with global prototypes $\mathcal{P}_g$:

$$S_{\text{coarse}}(x, i, t) = \text{sim}\left(\phi(x; A_t), p_{i,t}\right). \tag{14}$$

Let (y^*, t^*) be the top-1 prediction among the retrieved top-k candidates C_{top-k}.

Confusion Detection and Triggering. Stage 2 is triggered if and only if C_{top-k} contains a candidate (y', t') that shares the same connected component K_m as the top-1 prediction (y^*, t^*) but originates from a different stage, i.e., $t^* \neq t'$. Otherwise, (y^*, t^*) is retained as the final prediction, as task-specific adapters are sufficient for non-confusing or intra-stage scenarios.

Stage 2: Fine-Grained Refinement. If triggered, we first obtain the shared adapter A_m^{shared} for component K_m and derive the differential adapters by subtracting it from each task-specific adapter in the matched set S_m, i.e., $\Delta A_t = A_t - A_m^{shared}$. These differential adapters $\Delta\mathcal{A}_m = \{\Delta A_t \mid (y,t) \in K_m\}$, together with the rectified prototypes $\mathcal{P}_m^* = \{p_{y,t}^* \mid (y,t) \in K_m\}$ defined in Eq. 9, are then used to re-evaluate candidates within K_m for a precise decision:

$$(y_{final}, t_{final}) = \underset{(y,t) \in K_m}{\arg\max}\ \text{sim}\left(\phi(x; \Delta A_t), p_{y,t}^*\right). \tag{15}$$

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on the CIFAR100Krizhevsky et al. (2009), ImageNet-RHendrycks et al. (2021a), ImageNet-AHendrycks et al. (2021b), and VTABZhai et al. (2019) datasets, containing 100, 200, 200, and 50 classes respectively. Following standard CIL protocols Sun et al. (2025b); Wang et al. (2025), each dataset is split into non-overlapping tasks denoted

Table 1: Performance comparison on four datasets using ViT-B/16-IN21K as the backbone. The best results are highlighted in red and the second-best in blue. "−" denotes unavailable results.

Method	CIFAR100 B0 Inc5		ImageNet-R B0 Inc20		ImageNet-R B0 Inc10		ImageNet-A B0 Inc20		VTAB B0 Inc10	
	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$
FinetuneSun et al. (2025a)	38.90	20.17	32.31	22.78	30.28	20.74	24.28	14.51	34.95	21.25
L2PWang et al. (2022c)	85.94	79.93	75.46	69.77	74.23	69.58	49.39	41.71	77.11	77.10
DualPromptWang et al. (2022b)	87.87	81.15	73.10	67.18	72.45	65.43	53.71	41.67	83.36	81.23
ACIL Zhuang et al. (2022)	90.22	88.79	81.21	77.38	79.43	75.55	70.93	60.90	–	–
CODA-PromptSmith et al. (2023)	89.11	81.96	77.97	72.27	73.48	67.92	53.54	42.73	83.90	83.02
LAE Gao et al. (2023)	88.50	82.76	75.42	69.83	72.85	65.57	58.56	47.73	–	–
SLCAZhang et al. (2023)	92.49	88.55	81.17	77.00	84.32	79.37	68.66	58.74	90.94	90.76
DS-AL Zhuang et al. (2024)	88.82	85.91	78.37	77.48	75.90	74.05	63.47	52.67	–	–
EASEZhou et al. (2024)	91.51	85.80	81.74	76.17	79.12	72.83	65.34	55.04	93.61	93.55
SimpleCILZhou et al. (2025a)	87.57	81.26	61.26	54.55	66.89	60.87	59.77	48.91	85.99	84.38
APER+AdapterZhou et al. (2025a)	90.65	85.15	75.82	67.95	77.17	70.45	60.47	49.37	85.95	84.35
MoAL Gao et al. (2025)	93.23	88.41	84.45	79.33	82.94	76.85	67.58	56.77	–	–
MOSSun et al. (2025b)	93.30	89.25	82.96	77.93	81.32	75.63	69.13	59.12	92.62	92.79
TUNAWang et al. (2025)	94.44	90.74	84.22	79.42	82.67	77.45	73.78	64.78	93.72	93.47
IM+LCATran et al. (2026)	92.77	88.10	77.93	72.96	72.28	65.20	64.78	57.32	95.20	91.07
RAA Deng et al. (2026)	94.14	90.77	84.35	80.23	83.19	79.03	73.68	63.66	93.33	93.56
GOD Jia et al. (2026)	–	–	85.41	79.20	–	–	70.50	59.38	–	–
DPTA Xu et al. (2026)	92.90	88.60	84.57	78.03	82.54	74.85	69.56	58.67	94.38	93.85
CCS	94.84	91.55	86.49	81.68	84.92	79.48	74.53	66.75	94.95	94.95

by "B-m Inc-n", where m and n denote the numbers of classes in the base and incremental tasks, respectively, and $m = 0$ indicates equal division across all tasks.

Implementation Details. All experiments are implemented in PyTorch on a single NVIDIA RTX 3090 GPU, utilizing a self-supervised ViT-B/16-IN21K backbone pre-trained on ImageNet-21K. Complete experimental details can be found in Appendix A.1.

Metrics. Following standard practice, we report the final top-1 accuracy $\mathcal{A}_N$ and the average incremental accuracy $\bar{\mathcal{A}} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{A}_i$, where $\mathcal{A}_i$ is the accuracy after stage i. To validate the mitigation of inter-stage confusion, we propose two new metrics: **Confusion Component Accuracy (CCA)**, which measures the correct classification rate within the confusion groups, and **Within-Component Error Rate (WCER)**, which calculates the proportion of samples misclassified as another class within the same group. A higher CCA and a lower WCER indicate effective discrimination. Formal mathematical definitions are deferred to Appendix A.2.

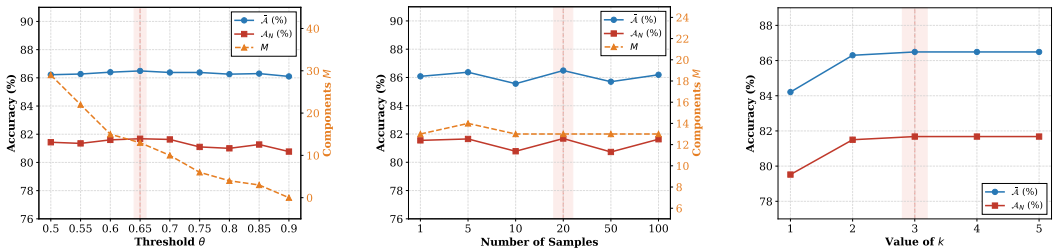
4.2 Experimental Results

Main Results. We compare CCS with state-of-the-art methods across four benchmarks using the ViT-B/16-IN21K backbone. As shown in Table 1, CCS consistently outperforms existing approaches such as TUNA and RAA, establishing new SOTA results on three datasets. By mitigating inter-stage class confusion, our framework achieves the best final accuracy on all five settings and the best average accuracy on four, surpassing the runner-up by up to 2% in final accuracy.

Ablation Analysis. As shown in Table 2, we conduct ablation studies to evaluate the key mechanisms of CCS on four benchmark datasets, with stage-wise results deferred to Appendix C.1. **CCS** denotes our full method. The variant without CACG generates random connected components instead of using similarity graphs and the Union-Find algorithm, while preserving identical component counts and sizes for fairness. The variant without CCDA replaces the differential adapter with a standard one. The variant without both CACG and CCDA removes the two-stage inference procedure. Removing any component degrades performance across all benchmarks, confirming that each proposed mechanism contributes to the overall improvement.

Table 2: Ablation study of different mechanisms in the CCS framework across four benchmark datasets.

CACG	CCDA	CIFAR100		ImageNet-R		ImageNet-A		VTAB	
		$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$
×	×	92.31	87.49	84.21	79.52	72.88	63.20	93.92	93.61
×	✓	92.75	87.65	85.38	80.30	72.35	63.27	94.36	93.60
✓	×	92.90	88.15	85.60	80.78	72.48	63.73	94.55	93.80
✓	✓	**94.84**	**91.55**	**86.49**	**81.68**	**74.53**	**66.75**	**94.95**	**94.95**



(a) Threshold θ (b) Number of samples (c) Value of k

Figure 3: Ablation studies on key hyperparameters. Red shaded bands indicate the optimal settings adopted in our experiments.

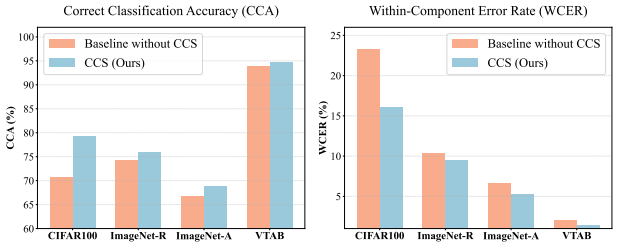


Figure 4: Effectiveness of CCS in suppressing inter-stage class confusion across four datasets.

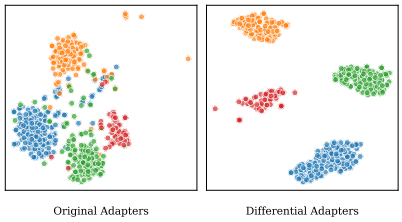


Figure 5: Visualization of 2D feature representations using t-SNE.

Furthermore, we ablate NPG with Mixup, CutMix, and Copy-Paste for pseudo-sample generation in CCS. Table 3 shows that NPG outperforms these methods across all benchmarks in $\bar{\mathcal{A}}$ and $\mathcal{A}_N$, demonstrating its ability to generate higher-fidelity representations.

Table 3: Ablation study on the pseudo-sample generation strategy within the CCS framework. Best results are highlighted in **bold**.

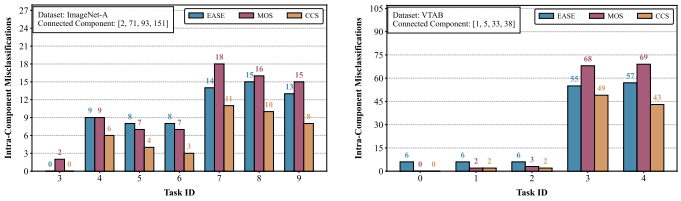
Setting	CIFAR100		ImageNet-R		ImageNet-A		VTAB	
	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$
CCS w/ Mixup Zhang et al. (2018)	94.74	91.43	86.13	81.60	74.43	66.29	94.55	94.86
CCS w/ CutMix Yun et al. (2019)	94.73	91.39	86.03	81.18	74.18	66.18	94.50	94.88
CCS w/ Copy-Paste Ghiasi et al. (2021)	94.51	90.68	85.99	80.42	74.31	66.16	93.78	93.86
CCS w/ NPG	**94.84**	**91.55**	**86.49**	**81.68**	**74.53**	**66.75**	**94.95**	**94.95**

We analyze the impact of key hyperparameters on ImageNet-R, with other datasets in Appendix A.3. As shown in Figure 3a, CCS performs best with $\theta = 0.65$, yielding 13 connected components. Lower thresholds erroneously group non-confusable classes and degrade accuracy. Figure 3b shows that 20 pseudo-samples per historical class maximize performance. To minimize overhead, we set the top-k candidate size to $k = 3$ for two-stage inference, where performance saturates (Figure 3c).

Effectiveness in Suppressing Inter-Stage Class Confusion. We first evaluate CCS from an overall metric perspective. Using CCA and WCER on four datasets, with task-specific adapters and prototype-based top-1 inference as the baseline, CCS consistently improves CCA and reduces WCER across all datasets, as shown in Figure 4. This confirms that CCS suppresses inter-stage interference and yields more accurate intra-component classification.

To verify that the improvement comes from suppressing misclassifications between confusable classes within the same connected component, we compare CCS with EASE and MOS. Figures 6a and 6b take representative components from ImageNet-A and VTAB as examples, showing that CCS consistently reduces intra-component misclassifications. Detailed results on other datasets are provided in the Appendix. To understand how CCS achieves this suppression, we visualize the feature space. Figure 5 shows t-SNE visualization on CIFAR100, where confusable classes overlap in the original feature space but are effectively separated by the differential adapters.

To examine whether feature separation yields more reliable predictions, Figure 7 visualizes prediction confidence before and after CCS. The baseline confuses semantically similar classes within the same component, e.g., Labrador Retriever vs. Golden Retriever and Hammer vs. Hatchet in Figures 7a and 7b, whereas CCS boosts confidence for ground-truth classes and suppresses confusing ones.

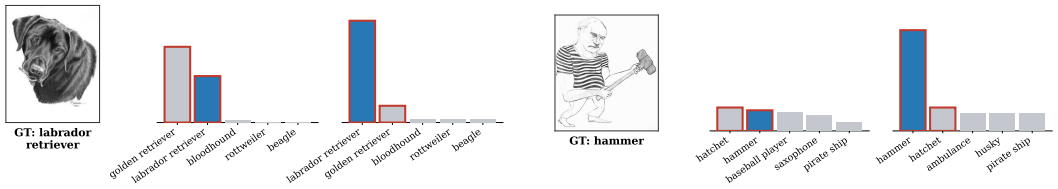


(a) ImageNet-A (b) VTAB

Figure 6: Intra-component misclassification counts across tasks.

Table 4: Parameter and time-consuming analysis. Best results are highlighted in **bold**.

Method	$\Delta\mathcal{P}$ (M)	Time (ms)
SLCAZhang et al. (2023)	85.81	970.72
CODA-PromptWang et al. (2022b)	3.99	213.31
RAADeng et al. (2026)	1.20	272.91
CCS	**0.79**	**186.33**



(a) "Labrador Retriever" and "Golden Retriever" (b) "Hammer" and "Hatchet"

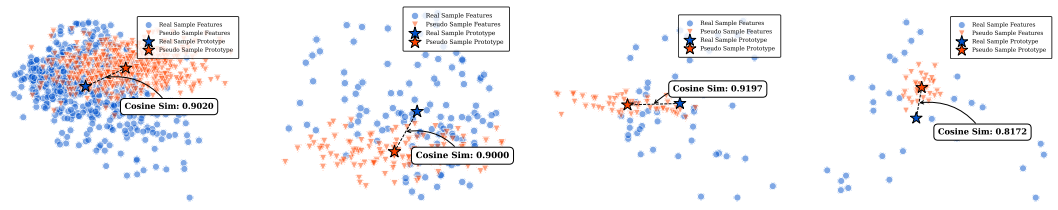
Figure 7: Visualizations of confidence scores before and after applying our CCS framework. Red boxes indicate categories in the same confusion group; blue bars denote the ground-truth (GT) class.

Quality of Generated Pseudo Samples. Figure 8 visualizes the feature distributions of real and pseudo-samples across four benchmarks. Their prototype cosine similarities consistently exceed 0.8 and reach 0.9 on three benchmarks. On CIFAR100 and ImageNet-R, the distributions overlap substantially in Figure 8a and 8b. For the more challenging ImageNet-A and VTAB, despite limited data, the high cosine similarity in Figure 8c and 8d confirms strong semantic alignment. These results demonstrate the fidelity and robustness of the NPG strategy.

Parameter-Time Analysis. Table 4 compares the updated parameters $\Delta\mathcal{P}$ and per-batch training time on ImageNet-R. Full fine-tuning methods such as SLCA incur high overhead in both metrics. The prompt-based CODA-Prompt trains faster but requires more parameters (3.99M), while the adapter-based RAA uses fewer parameters (1.20M) but trains slower (272.91 ms). In contrast, CCS minimizes both: it updates only 0.79M parameters and achieves the fastest training speed at 186.33 ms, striking a favorable balance between parameter size and computational cost.

5 Conclusion

This study addresses the critical challenge of inter-stage class confusion in CIL. We propose Cross-stage Confusion Suppression (CCS), a novel PTM-based framework that explicitly disentangles overlapping features. By integrating a Connected Component-Guided Differential Adapter mechanism with Noise-Driven Pseudo-Sample Generation, CCS effectively isolates discriminative class-specific representations without historical data. Coupled with an adaptive two-stage inference procedure, our approach refines the training-to-inference pipeline. Extensive experiments demonstrate that CCS achieves SOTA performance on three datasets.



(a) CIFAR100 (b) ImageNet-R (c) ImageNet-A (d) VTAB

Figure 8: t-SNE visualization of real and pseudo sample features on four datasets. The blue circles (○) and orange downward triangles (▽) denote the features of real and pseudo samples, respectively. The large star markers (☆) represent their prototypes.

AI USE STATEMENT

In this work, we have not used generative AI tools for developing theoretical models, formulating mathematical claims, proposing hypotheses, designing research methodology or experiments, implementing methods, interpreting results, cleaning datasets, or supporting qualitative data analysis. These tasks were performed entirely by the authors. Additionally, we used generative AI tools for language polishing and grammar correction. We have reviewed all AI-assisted work. Specifically, we checked AI-suggested edits line-by-line to ensure they did not alter technical meaning or introduce inaccuracies. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study excludes any use of human participants, sensitive information, or personally identifiable data. All empirical evaluations rely exclusively on open-source public benchmarks in compliance with their respective licenses. The proposed CCS framework focuses purely on resolving cross-stage class confusion in exemplar-free class-incremental learning. As a fundamental algorithmic development, this methodology poses no foreseeable risks of societal harm or malicious application. This submission strictly aligns with the ICLR Code of Ethics and upholds the principles of academic integrity and transparency.

REFERENCES

Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *CVPR*, pp. 28762–28771, 2024.

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *ICCV*, pp. 844–853, 2021.

Kexin Bao, Daichi Zhang, Hansong Zhang, Yong Li, Yutao Yue, and Shiming Ge. Cdˆ 2: Constrained dataset distillation for few-shot class-incremental learning. In *IJCAI*, pp. 4761–4769, 2025.

Peihua Deng, Jiehua Zhang, Xichun Sheng, Chenggang Yan, Yaoqi Sun, Ying Fu, and Liang Li. Multi-granularity class prototype topology distillation for class-incremental source-free unsupervised domain adaptation. In *CVPR*, pp. 30566–30576, 2025.

Yao Deng, Xiang Xiang, and Jiaqi Gui. Random amalgamation of adapters for flatter loss landscapes: towards class-incremental learning with better stability. In *AAAI*, pp. 20746–20754, 2026.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, pp. 9285–9295, 2022.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *ICCV*, pp. 11449–11459, 2023.

Zijian Gao, Wangwang Jia, Xingxing Zhang, Dulan Zhou, Kele Xu, Feng Dawei, Yong Dou, Xinjun Mao, and Huaimin Wang. Knowledge memorization and rumination for pre-trained model-based class-incremental learning. In *CVPR*, pp. 20523–20533, 2025.

Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, pp. 2918–2928, 2021.

Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D Bagdanov, Bartlomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *ECCV*, pp. 473–490. Springer, 2024.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *NeurIPS*, pp. 6582–6595, 2023.

Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bart Twardowski, Joost Van De Weijer, et al. Resurrecting old classes with new data for exemplar-free continual learning. In *CVPR*, pp. 28525–28534, 2024.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *Ai Open*, 2:225–250, 2021.

Run He, Di Fang, Yicheng Xu, Yawen Cui, Ming Li, Cen Chen, Ziqian Zeng, and Huiping Zhuang. Semantic shift estimation via dual-projection and classifier reconstruction for exemplar-free class-incremental learning. In *ICML*, volume 267, pp. 22392–22406, 2025.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pp. 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pp. 15262–15271, 2021b.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pp. 831–839, 2019.

Zhiyuan Hu, Yunsheng Li, Jiancheng Lyu, Dashan Gao, and Nuno Vasconcelos. Dense network expansion for class incremental learning. In *CVPR*, pp. 11858–11867, 2023.

Libo Huang, Xiangqi Li, Jiarui Zhao, Zhulin An, Chuanguang Yang, Boyu Diao, Fei Wang, Yan Zeng, Zhifeng Hao, and Yongjun Xu. Preprompt: Predictive prompting for class incremental learning. In *ACM SIGKDD*, pp. 1886–1897, 2026.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, pp. 35554–35584, 2023.

Wangwang Jia, Zijian Gao, Tianjiao Wan, Yuan Cao, Yong Dou, and Kele Xu. Geometry-driven ood detectors are class-incremental learners. In *CVPR*, pp. 34638–34649, 2026.

Kai Jiang, Zisong Lin, Hongyuan Zhang, Xueru Bai, and Xuelong Li. Miles: Metric learning with expandable subspace for pre-trained model-based class-incremental learning. *arXiv preprint arXiv:2607.17593*, 35:8291–8305, 2026a.

Kai Jiang, Zhengyan Shi, Dell Zhang, Hongyuan Zhang, and Xuelong Li. Mixture of noise for pre-trained model-based class-incremental learning. In *NeurIPS*, pp. 44776–44802, 2026b.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf*, 2009.

Guannan Lai, Da-Wei Zhou, Xin Yang, and Han-Jia Ye. The lie of the average: How class incremental learning evaluation deceives you? In *ICLR*, pp. 110530–110562, 2026.

Lan Li, Tao Hu, Da-Wei Zhou, Jia-Qi Yang, Han-Jia Ye, and De-Chuan Zhan. Bofa: Bridge-layer orthogonal low-rank fusion for clip-based class-incremental learning. In *AAAI*, pp. 22967–22975, 2026.

Yuhang Li, Guoxu Zhou, Zhenhao Huang, Xinqi Chen, Yuning Qiu, and Qibin Zhao. Tensor decomposition based memory-efficient incremental learning. In *ICML*, volume 267, pp. 36906–36920, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE TPAMI*, 40(12):2935–2947, 2017.

Lei Liu, Li Liu, and Yawen Cui. Prior-free balanced replay: Uncertainty-guided reservoir sampling for long-tailed continual learning. In *ACM MM*, pp. 2888–2897, 2024.

Aojun Lu, Tao Feng, Hangjie Yuan, Chunhui Ding, and Yanan Sun. Adapt before continual learning. In *AAAI*, volume 40, pp. 24061–24069, 2026a.

Haori Lu, Xusheng Cao, Linlan Huang, Enguang Wang, Fei Yang, and Xialei Liu. Restoring forgotten knowledge in non-exemplar class incremental image classification through test-time semantic evolution. *IEEE TMM*, pp. 1–13, 2026b.

Yue Lu, Shizhou Zhang, De Cheng, Yinghui Xing, Guoqiang Liang, Nannan Wang, Peng Wang, and Yanning Zhang. Continual low-rank adaptation via cumulative unified optimization. *IEEE TPAMI*, pp. 1–18, 2026c.

Federico Pernici, Matteo Bruni, Claudio Baecchi, Francesco Turchini, and Alberto Del Bimbo. Class-incremental learning with pre-allocated fixed classifiers. In *ICPR*, pp. 6259–6266, 2021.

Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *WACV*, pp. 3911–3920, 2023.

Zhuang Qi, Ying-Peng Tang, Lei Meng, Han Yu, Xiaoxiao Li, and Xiangxu Meng. Class-wise balancing data replay for federated class-incremental learning. In *NeurIPS*, pp. 145309–145334, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763, 2021.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, volume 30, pp. 506–516, 2017.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pp. 11909–11919, 2023.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, volume 30, pp. 4077–4087, 2017.

Hai-Long Sun, Da-Wei Zhou, De-Chuan Zhan, and Han-Jia Ye. Pilot: A pre-trained model-based continual learning toolbox. *Science China Information Sciences*, 68(4):147101–147102, 2025a.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning. In *AAAI*, pp. 20699–20707, 2025b.

Qing Sun, Fan Lyu, Fanhua Shang, Wei Feng, and Liang Wan. Exploring example influence in continual learning. In *NeurIPS*, pp. 27075–27086, 2022.

Tarjan and Robert Endre. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975.

Tung Tran, Danilo Vasconcellos Vargas, and Khoat Than. Lca: Local classifier alignment for continual learning. In *ICLR*, volume 2026, pp. 127572–127593, 2026.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *ECCV*, pp. 398–414. Springer, 2022a.

Shaokun Wang, Weiwei Shi, Yuhang He, Yifan Yu, and Yihong Gong. Non-exemplar class-incremental learning via adaptive old class reconstruction. In *ACM MM*, pp. 4524–4534, 2023.

Yan Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *ICCV*, pp. 806–816, 2025.

Yu Wang and Shiwei Chen. Multi-agent trajectory prediction with spatio-temporal sequence fusion. *IEEE TMM*, 25:13–23, 2021.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, pp. 631–648. Springer, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pp. 139–149, 2022c.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. In *ICLR*, pp. 58337–58353, 2025.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pp. 374–382, 2019.

Yang Xiao, Peng Tianyi, Rohan Kumar Das, Yuchen Hu, and Huiping Zhuang. Analytickws: towards exemplar-free analytic class incremental learning for small-footprint keyword spotting. In *ACL*, pp. 14147–14158, 2025.

Zhiming Xu, Suorong Yang, Baile Xu, Furao Shen, and Jian Zhao. Dual prototypes for adaptive pre-trained model in class-incremental learning. *Neural Networks*, 196:108389, 2026.

Yuanbo Yang, Jiahui Qu, Wenqian Dong, Ling Huang, and Yunsong Li. Prototype-based meta-prompt tuning: Toward rehearsal-free few-shot class-incremental learning for multimodal remote sensing image. *IEEE TIP*, 35:434–448, 2026.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *CVPR*, pp. 23219–23230, 2024.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, pp. 6982–6991, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pp. 6023–6032, 2019.

Jiang-Tian Zhai, Xialei Liu, Lu Yu, and Ming-Ming Cheng. Fine-grained knowledge selection and restoration for non-exemplar class incremental learning. In *AAAI*, pp. 6971–6978, 2024.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *ICCV*, pp. 19148–19158, 2023.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, pp. 2866–2878, 2018.

Huan Zhang, Shenghua Fan, Shuyu Dong, Yujin Zheng, Dingwen Wang, and Fan Lyu. Sparse tuning enhances plasticity in ptm-based continual learning. In *AAAI*, volume 40, pp. 28230–28238, 2026a.

Weichao Zhang, Shuai Zheng, Yeyu Yan, Zhizhe Liu, Zhenfeng Zhu, and Yao Zhao. Subspace-aware feature reshaping for open-set graph class-incremental learning. In *ICML*, 2026b. URL https://openreview.net/forum?id=QKo90eNH3l.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pp. 13208–13217, 2020.

Jiarui Zhao, Libo Huang, Xiangqi Li, Zhulin An, Chuanguang Yang, Yu Wang, Boyu Diao, and Yongjun Xu. Representation-steered incremental adapter-tuning for class-incremental learning with pre-trained models. In *CVPR*, pp. 18010–18020, 2026.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, pp. 23554–23564, 2024.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *IJCV*, 133(3):1012–1032, 2025a.

Da-Wei Zhou, Yuanhan Zhang, Yan Wang, Jingyi Ning, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Learning without forgetting for vision-language models. *IEEE TPAMI*, 47(6):4489–4504, 2025b.

Xinyu Zhou, Jing Yang, Xiaoli Ruan, Yuling Chen, Yong Yao, Xu Wang, and Zhidong Su. Caff-cil: Causality-aware freedom forgetting approach for class-incremental learning. *IEEE TNNLS*, pp. 1–15, 2026.

Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *CVPR*, pp. 9296–9305, 2022.

Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. Acil: Analytic class-incremental learning with absolute memorization and privacy protection. In *NeurIPS*, pp. 11602–11614, 2022.

Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. Ds-al: A dual-stream analytic learning for exemplar-free class-incremental learning. In *AAAI*, pp. 17237–17244, 2024.

APPENDIX

In this appendix, we provide additional experimental settings, discussions, and experimental results. The appendix is organized as follows:

A. More Experimental Settings

1. Implementation Details
2. Metrics
3. Hyperparameter Selection

B. More Discussions

1. Trigger Rate and Overhead of Stage 2
2. Comprehensive Graph Topology Analysis
3. Intra-Component Confusion Analysis
4. Discussion on Weight-Space Subtraction
5. Limitation and Bad Cases

C. More Experimental Results

1. Ablation Study on Key Mechanisms
2. Robustness across Random Seeds
3. Baseline Performance of the Pre-Trained Model

A MORE EXPERIMENTAL SETTINGS

A.1 IMPLEMENTATION DETAILS

All experiments are implemented in PyTorch on a single NVIDIA RTX 3090 GPU, utilizing a self-supervised ViT-B/16-IN21K backbone pre-trained on ImageNet-21K. In the proposed CCS method, the SGD optimizer with momentum is adopted. The batch size is set to 48, and the training spans 20 epochs. The initial learning rate is 0.01, decaying via a cosine annealing schedule, while the differential adapter is fine-tuned for 10 epochs. The choice of the similarity threshold θ and the number of pseudo samples is analyzed in Appendix A.3. The adapter projection dimension r, temperature hyperparameter τ, step size η, iteration number n, and top-k selection k are set to 16, 0.1, 20, 3, and 3, respectively.

In Table1, we evaluate CCS against state-of-the-art PTM-based CIL methods: L2P Wang et al. (2022c), DualPrompt Wang et al. (2022b), ACIL Zhuang et al. (2022), CODA-Prompt Smith et al. (2023), LAE Gao et al. (2023), SLCAZhang et al. (2023), DS-AL Zhuang et al. (2024), EASE Zhou et al. (2024), SimpleCILZhou et al. (2025a), APER+AdapterZhou et al. (2025a), MoAL Gao et al. (2025), MOS Sun et al. (2025b), TUNAWang et al. (2025), IM+LCA Tran et al. (2026), RAA Deng et al. (2026), GOD Jia et al. (2026), and DPTA Xu et al. (2026). As a reference baseline, we also report the performance of sequentially fine-tuning the pre-trained model, denoted as FinetuneSun et al. (2025a). For a fair comparison, all methods are implemented with identical PTM backbones and protocols.

A.2 METRICS

To validate the mitigation of inter-stage confusion, we propose two metrics: Confusion Component Accuracy (CCA) and Within-Component Error Rate (WCER). Given M confusion groups $\{K_m\}_{m=1}^{M}$, let $\mathcal{X}_m$ be the sample set for K_m, with ground-truth y and prediction $\hat{y}$. CCA measures the correct classification rate within these groups:

$$\text{CCA} = \frac{\sum_{m=1}^{M} \sum_{x \in \mathcal{X}_m} \mathbb{I}\big(\hat{y}(x) = y(x)\big)}{\sum_{m=1}^{M} |\mathcal{X}_m|}. \tag{16}$$

WCER measures the proportion of samples misclassified as another class within the same group:

$$\text{WCER} = \frac{\sum_{m=1}^{M} \sum_{x \in \mathcal{X}_m} \mathbb{I}\big(\hat{y}(x) \neq y(x) \text{ and } \hat{y}(x) \in K_m\big)}{\sum_{m=1}^{M} |\mathcal{X}_m|}, \tag{17}$$

Table 5: **Sensitivity analysis of hyperparameters θ, number of samples, and top-k on CIFAR100.** The study systematically assesses the threshold, sample size, and candidate selection. Best results are highlighted in **bold**. Our selected hyperparameters ($\theta = 0.70$, 20 samples, $k = 3$) are highlighted with the gray background.

θ	Samples	Top-k	Metrics (%)		Fluctuations (Δ)	
			$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\Delta\bar{\mathcal{A}}$	$\Delta\mathcal{A}_N$
0.60	20	3	**94.88**	91.31	**+0.04**	−0.24
0.65	20	3	94.84	91.34	0.00	−0.21
0.70	20	3	94.84	**91.55**	0.00	**0.00**
0.75	20	3	94.60	91.25	−0.24	−0.30
0.80	20	3	94.60	90.93	−0.24	−0.62
0.70	5	3	94.60	91.14	−0.24	−0.41
0.70	10	3	94.60	91.14	−0.24	−0.41
0.70	20	3	94.84	**91.55**	0.00	**0.00**
0.70	50	3	94.84	**91.55**	0.00	**0.00**
0.70	100	3	94.60	91.14	−0.24	−0.41
0.70	20	1	92.82	87.49	−2.02	−4.06
0.70	20	2	94.76	91.43	−0.08	−0.12
0.70	20	3	94.84	**91.55**	0.00	**0.00**
0.70	20	4	94.84	**91.55**	0.00	**0.00**
0.70	20	5	94.84	**91.55**	0.00	**0.00**

where $\mathbb{I}(\cdot)$ is the indicator function. High CCA and low WCER indicate effective discrimination.

A.3 Hyperparameter selection

We conduct a comprehensive hyperparameter sensitivity study on CIFAR100, ImageNet-R, ImageNet-A, and VTAB, with results summarized in Table 5, Table 6, Table 7, and Table 8. Overall, CCS exhibits strong robustness across all benchmarks, with evaluation metrics showing minor fluctuations under different configurations. We detail the selection of three key hyperparameters: the similarity threshold θ, the number of pseudo-samples per class, and the top-k selection value.

Similarity Threshold θ. The threshold θ governs the formation of connected components. A lower threshold groups more classes into each component, but a larger number of connected components is not always beneficial. Excessively low thresholds erroneously group non-confusable classes together, which can degrade accuracy rather than improve it. Therefore, we select thresholds that optimally balance performance and computational efficiency.

For CIFAR100, we choose $\theta = 0.70$. It delivers the best final accuracy of 91.55% and produces fewer connected components than $\theta = 0.60$. This offers a clear computational advantage with negligible accuracy loss. On ImageNet-A and VTAB, we select $\theta = 0.60$ and $\theta = 0.25$ respectively, where performance saturates and further lowering the threshold introduces excessive components without meaningful gains.

Number of Pseudo-Samples. On CIFAR100, ImageNet-R, and VTAB, performance peaks at 20 samples per class, achieving 94.84%, 86.49%, and 94.95% average accuracy respectively. For ImageNet-A, we choose 5 samples as it suffices to reach the best performance of 74.53%. Across all datasets, performance remains stable under varying sample sizes, with average accuracy fluctuations within 1%, demonstrating strong robustness.

Top-k Selection. We uniformly adopt $k = 3$ across all datasets, as performance saturates at this point. On CIFAR100, $k = 3$, $k = 4$, and $k = 5$ yield identical results of 94.84% average accuracy and 91.55% final accuracy. The same saturation occurs on ImageNet-R at 86.49%, ImageNet-A at 74.53%, and VTAB at 94.95%. We choose $k = 3$ to minimize computational overhead while retaining full performance. When $k = 1$, a substantial accuracy drop is observed across all benchmarks;

Table 6: **Sensitivity analysis of hyperparameters θ, number of samples, and top-k on ImageNet-R (B0 Inc20).** The study systematically assesses the threshold, sample size, and candidate selection, revealing robust performance across different configurations and identifying the optimal setting ($\theta = 0.65$, 20 samples, $k = 3$) for task performance. Best results are highlighted in **bold**. Our selected hyperparameters are highlighted with the gray background.

θ	Samples	Top-k	Metrics (%)		Fluctuations (Δ)	
			$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\Delta\bar{\mathcal{A}}$	$\Delta\mathcal{A}_N$
0.50	20	3	86.22	81.43	−0.27	−0.25
0.55	20	3	86.27	81.35	−0.22	−0.33
0.60	20	3	86.40	81.60	−0.09	−0.08
0.65	20	3	**86.49**	**81.68**	**0.00**	**0.00**
0.70	20	3	86.38	81.63	−0.11	−0.05
0.75	20	3	86.38	81.10	−0.11	−0.58
0.80	20	3	86.26	81.00	−0.23	−0.68
0.85	20	3	86.30	81.27	−0.19	−0.41
0.90	20	3	86.10	80.77	−0.39	−0.91
0.65	1	3	86.08	81.55	−0.41	−0.13
0.65	5	3	86.37	81.65	−0.12	−0.03
0.65	10	3	85.56	80.78	−0.93	−0.90
0.65	20	3	**86.49**	**81.68**	**0.00**	**0.00**
0.65	50	3	85.70	80.73	−0.79	−0.95
0.65	100	3	86.19	81.63	−0.30	−0.05
0.65	20	1	84.21	79.52	−2.28	−2.16
0.65	20	2	86.30	81.50	−0.19	−0.18
0.65	20	3	**86.49**	**81.68**	**0.00**	**0.00**
0.65	20	4	**86.49**	**81.68**	**0.00**	**0.00**
0.65	20	5	**86.49**	**81.68**	**0.00**	**0.00**

for instance, CIFAR100 average accuracy falls from 94.84% to 92.82%. This decline is **not due to poor robustness**, as $k = 1$ is equivalent to ablating the two-stage inference procedure and disabling our CACG and CCDA modules.

B More Discussions

B.1 Trigger Rate and Overhead of Stage 2

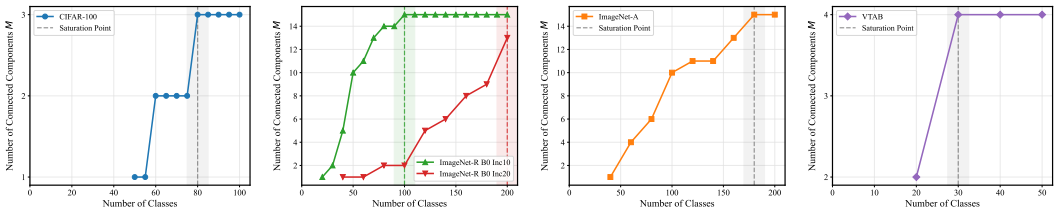
Table 9 presents the trigger rates, accuracy gains, and computational overhead associated with the Stage 2 refinement process. Across the evaluated datasets, the trigger rate ranges from 9.05% to 15.77%. The additional computational cost remains marginal, incurring only 2.24 to 11.20 ms of extra inference time per image and requiring 0.61 to 1.53 M additional parameters. Crucially, this modest overhead consistently yields clear accuracy improvements (up to +2.53%), validating that Stage 2 efficiently rectifies challenging samples. These results demonstrate that the proposed two-stage inference mechanism achieves a favorable trade-off between performance and computational cost.

B.2 Comprehensive Graph Topology Analysis

Overall Growth Trend. To provide a better understanding of cross-stage confusion mitigation, we analyze how connected components $K_m \in \mathcal{K}$ evolve during incremental learning. We first show the growth of the total component count M across learned classes in Figure 9. These line plots show that confusion components increase steadily for CIFAR100 in Figure 9a, ImageNet-R in Figure 9b, ImageNet-A in Figure 9c, and VTAB in Figure 9d, before reaching their saturation points.

Table 7: **Sensitivity analysis of hyperparameters θ, number of samples, and top-k on ImageNet-A.** Best results are highlighted in **bold**. Our selected hyperparameters ($\theta = 0.60$, 5 samples, $k = 3$) are highlighted with the gray background. "-" denotes that an excessively low threshold introduces too many non-confusable classes, causing a substantial performance degradation, so the results are omitted.

θ	Samples	Top-k	Metrics (%)		Fluctuations (Δ)	
			$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\Delta\bar{\mathcal{A}}$	$\Delta\mathcal{A}_N$
0.50	5	3	-	-	-	-
0.55	5	3	74.29	**66.82**	−0.24	**+0.07**
0.60	5	3	**74.53**	66.75	**0.00**	0.00
0.65	5	3	73.95	65.24	−0.58	−1.51
0.70	5	3	74.05	64.78	−0.48	−1.97
0.75	5	3	74.27	65.70	−0.26	−1.05
0.60	1	3	74.40	66.56	−0.13	−0.19
0.60	2	3	74.30	66.23	−0.23	−0.52
0.60	5	3	**74.53**	66.75	**0.00**	0.00
0.60	10	3	74.29	66.69	−0.24	−0.06
0.60	20	3	74.48	66.49	−0.05	−0.26
0.60	5	1	72.35	63.27	−2.18	−3.48
0.60	5	2	73.96	65.24	−0.57	−1.51
0.60	5	3	**74.53**	66.75	**0.00**	0.00
0.60	5	4	**74.53**	66.75	**0.00**	0.00
0.60	5	5	**74.53**	66.75	**0.00**	0.00



(a) CIFAR100 (b) ImageNet-R (c) ImageNet-A (d) VTAB

Figure 9: **Evolution of the number of connected components across incremental stages.** On all four benchmarks, the number of confusion components M steadily increases as the model continually learns new classes, eventually reaching an upper bound indicated by the dashed line.

Structural Evolution of Confusion Components. To further dissect these trends, Tables 10–13 track the global graph properties and specific connected components for the four benchmark datasets.

As the model continually learns, the Union-Find algorithm effectively captures transitive confusion chains. In early tasks, confusion is often strictly pairwise, indicated by smaller component sizes and a chain length of 1, as explicitly demonstrated for Task 4 of ImageNet-A in Table 12. As training progresses, these components naturally expand into complex, multi-hop confusion topologies rather than trivial dense cliques. For instance, in Table 10 for CIFAR100, we observe a steady growth in the maximum component size from 3 to 4 across tasks, accurately capturing newly introduced interfering classes.

Notably, the semantic purity remains highly competitive across all datasets, confirming that our connected component generation aligns with genuine semantic similarities rather than indiscriminately merging isolated classes. This is particularly evident for ImageNet-R (B0 Inc20) in Table 11, where

Table 8: **Sensitivity analysis of hyperparameters θ, number of samples, and top-k on VTAB.** Best results are highlighted in **bold**. Our selected hyperparameters ($\theta = 0.25$, 20 samples, $k = 3$) are highlighted with the gray background.

θ	Samples	Top-k	Metrics (%)		Fluctuations (Δ)	
			$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\Delta\bar{\mathcal{A}}$	$\Delta\mathcal{A}_N$
0.20	20	3	**95.02**	94.74	**+0.07**	−0.20
0.25	20	3	94.95	**94.94**	0.00	**0.00**
0.30	20	3	94.87	94.52	−0.08	−0.42
0.35	20	3	94.33	93.61	−0.62	−1.33
0.40	20	3	94.33	93.61	−0.62	−1.33
0.25	1	3	94.94	94.88	−0.01	−0.06
0.25	2	3	94.93	94.84	−0.02	−0.10
0.25	5	3	94.95	94.88	0.00	−0.06
0.25	10	3	94.92	94.81	−0.03	−0.13
0.25	20	3	94.95	**94.94**	0.00	**0.00**
0.25	20	1	94.33	93.60	−0.62	−1.34
0.25	20	2	94.39	94.80	−0.56	−0.14
0.25	20	3	94.95	**94.94**	0.00	**0.00**
0.25	20	4	94.95	**94.94**	0.00	**0.00**
0.25	20	5	94.95	**94.94**	0.00	**0.00**

Table 9: Trigger rate, accuracy gain, and computational overhead of the Stage 2 refinement process.

Dataset	Trigger Rate (%)	Acc. Gain (%)	Extra Time (ms)	Extra Params (M)
CIFAR100	9.36	+2.53	2.24	0.92
ImageNet-R	9.05	+2.28	4.09	0.61
ImageNet-A	12.90	+1.65	6.67	1.22
VTAB	15.77	+1.03	11.20	1.53

despite the total number of valid components nearly tripling from Task 5 to Task 9, the semantic purity is strictly maintained above 95%.

Finally, as detailed in Table 13, the cross-domain VTAB benchmark exhibits a uniquely distinct topological evolution. Confusion components manifest fully in the later task stages with larger cluster sizes up to 5 and longer chains up to 3, reflecting the severe distribution shifts inherent to VTAB's diverse domains. Guided by these diverse and dynamically evolving topological structures, the proposed CCDA can precisely target and resolve the interference within each specific K_m.

B.3 Intra-Component Confusion Analysis

To demonstrate the necessity and effectiveness of dynamically constructing connected components, we analyze the intra-component misclassification behavior of CCS against state-of-the-art adapter-based methods, EASE and MOS. Tables 14, 15, 16, and 17 report fine-grained mutual misclassification statistics among the constituent classes of representative components across the four benchmark datasets. In EASE and MOS, these classes frequently misclassify into one another as incremental learning progresses. For instance, the highly entangled pair of Class 33 and Class 38 in VTAB suffers catastrophic mutual confusion, as reported in Table 17, while severe error accumulation plagues Classes 55, 57, 62, and 74 in CIFAR100, as shown in Table 14. Similar cross-confusion is evident between the tracked classes in ImageNet-R and ImageNet-A, as detailed in Tables 15 and 16, respectively. This persistent confusion confirms that these categories share severely overlapping feature subspaces across incremental stages, and thus require explicit grouping for targeted disambiguation.

Table 10: Comprehensive analysis of connected component evolution across incremental tasks on **CIFAR100**. Global graph properties report the total number of valid components (M), maximum component size (Max Size), average edge density (Density), semantic purity (Purity), and maximum chain length (Chain). Representative connected components ($K_1, K_2, K_3 \in \mathcal{K}$) illustrate cross-task transitive merging. Newly added classes within each component at a given task are highlighted with **blue**.

Task	Graph Properties (Global)					Comp (K_m)	Class IDs within the Component
	M	Max Size	Density	Purity	Chain		
Task 10	1	3	0.67	100.0%	2	K_1	**35**, **45**, **54**
Task 12	2	3	0.67	83.3%	2	K_1	35, 45, 54
						K_2	**55**, **57**, **62**
Task 14	2	4	0.67	88.9%	2	K_1	35, 45, 54
						K_2	55, 57, 62, **74**
Task 16	3	4	0.83	92.3%	2	K_1	35, 45, 54, **76**
						K_2	55, 57, 62, 74
						K_3	**47**, **83**

Table 11: Comprehensive analysis of connected component evolution across incremental tasks on **ImageNet-R (B0 Inc20)**. Newly added classes within each component at a given task are highlighted with **green**.

Task	Graph Properties (Global)					Comp (K_m)	Class IDs within the Component
	M	Max Size	Density	Purity	Chain		
Task 4	2	2	1.00	100.0%	1	K_1	**47**, **79**
Task 5	5	2	1.00	100.0%	1	K_1	47, 79
						K_2	**2**, **93**
Task 7	8	3	0.97	95.2%	2	K_1	47, 79
						K_2	2, 93
						K_3	**31**, **123**
Task 9	13	3	0.98	95.5%	2	K_1	47, 79
						K_2	2, 93, **178**
						K_3	31, 123
						K_4	**8**, **25**, **145**

By identifying these confusing categories through the CACG module, CCS applies the CCDA mechanism to directly mitigate cross-stage feature confusion. The longitudinal impact of this mechanism is visually summarized in our task-wise bar charts. As shown in Figure 6b, the total intra-component misclassifications for MOS on VTAB surge to 68 at Task 3 and reach 69 at Task 4. Similarly, Figure 10a illustrates how EASE experiences a massive error spike on CIFAR100 at Task 14, recording 121 mutual misclassifications. Across all task sequences, CCS consistently yields fewer total intra-component misclassifications, effectively suppressing these sudden error surges. On ImageNet-R, as shown in Figure 10b, when EASE and MOS errors jump to 15 and 9 respectively at Task 7, CCS reduces the count to just 6. Furthermore, in the final stage of ImageNet-A, as illustrated in Figure 6a, CCS restricts the intra-component errors to 8, establishing a clear advantage over EASE with 13 and MOS with 15. By effectively untangling decision boundaries, CCS prevents the severe accumulation of mutual prediction errors from early stages into later tasks.

Table 12: Comprehensive analysis of connected component evolution across incremental tasks on **ImageNet-A**. Newly added classes within each component at a given task are highlighted with **orange**.

Task	Graph Properties (Global)					Comp (K_m)	Class IDs within the Component
	M	Max Size	Density	Purity	Chain		
Task 4	6	2	1.00	100.0%	1	K_1	**2**, **93**
Task 6	11	3	0.97	92.3%	2	K_1	2, 93
						K_2	**74**, **102**, **131**
						K_3	**112**, **137**
Task 7	11	4	0.91	80.0%	2	K_1	2, **71**, 93, **151**
						K_2	74, 102, 131
						K_3	112, 137, **155**
Task 9	15	4	0.91	79.3%	2	K_1	2, 71, 93, 151
						K_2	74, 102, 131
						K_3	112, 137, 155, **187**

Table 13: Comprehensive analysis of connected component evolution across incremental tasks on **VTAB**. Confusion components uniquely manifest fully in the later tasks of this benchmark's task sequence. Newly added classes within each component at a given task are highlighted with **purple**.

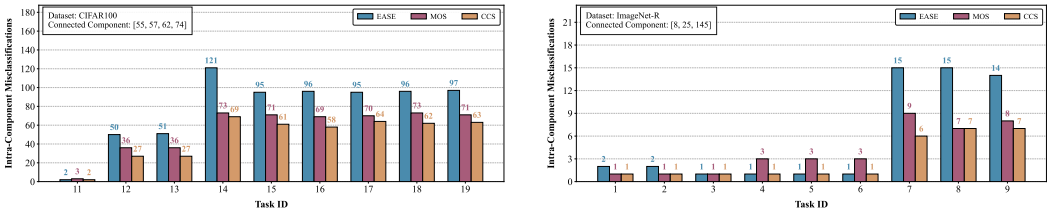
Task	Graph Properties (Global)					Comp (K_m)	Class IDs within the Component
	M	Max Size	Density	Purity	Chain		
Task 3	4	5	0.73	66.7%	3	K_1	**6**, **9**, **11**, **32**, **37**
						K_2	**1**, **5**, **33**, **38**
						K_3	**7**, **34**
Task 4	4	5	0.73	66.7%	3	K_1	6, 9, 11, 32, 37
						K_2	1, 5, 33, 38
						K_3	7, 34

Nevertheless, distinguishing fine-grained semantic differences among classes within the same component remains inherently challenging in class-incremental learning without historical exemplars. While CCS substantially alleviates structural confusion and strictly bounds error escalation, it does not fully eliminate all intra-component errors. For example, residual confusion still occurs in the final stages of CIFAR100, where CCS records 63 misclassifications compared to 97 in EASE and 71 in MOS. Future work could explore finer-grained subspace orthogonalization to further isolate these highly entangled representations.

B.4 Discussion on Weight-Space Subtraction

In this work, we adopt weight-space subtraction to extract task-specific residuals. We do not claim this operation as a mathematically exact or even approximate surrogate for feature-space decoupling. Due to the non-linear activation $\sigma(\cdot)$ and the scaling factor, higher-order Taylor terms are empirically non-negligible. Instead, we position weight-space subtraction as a *task-arithmetic-inspired, computationally efficient inductive bias*, justified by the following considerations.

Optimization Geometry and Shared Initialization. Both A_t and A_{shared} are initialized from the same checkpoint and optimized via SGD under identical hyperparameters. In this sense, A_t and A_{shared} reside in the same loss landscape and share a common parameter basis. Consequently, their difference $\Delta A = A_t - A_{shared}$ captures the *relative displacement* from the shared center, which



(a) CIFAR100 (b) ImageNet-R

Figure 10: Intra-component misclassification counts across tasks.

Table 14: Intra-component misclassification evolution on **CIFAR100** (Target Component Classes: 55, 57, 62, 74). We report the mutual prediction errors at representative incremental stages (Task 14, Task 16, and Task 19). CCS effectively mitigates cross-stage confusion from early stages and prevents severe error accumulation.

Misclassification ($A \rightarrow B$)	**Task 14** (Early Stage)			**Task 16** (Middle Stage)			**Task 19** (Final Stage)		
	EASE	MOS	**CCS**	EASE	MOS	**CCS**	EASE	MOS	**CCS**
Class 55 → Class 57	0	0	0	0	0	0	0	0	0
Class 55 → Class 62	19	6	4	10	5	4	10	4	4
Class 55 → Class 74	12	3	4	7	2	0	7	2	0
Class 57 → Class 55	0	2	0	0	2	1	0	2	1
Class 57 → Class 62	10	3	5	10	4	3	9	4	3
Class 57 → Class 74	19	13	11	19	13	8	20	14	7
Class 62 → Class 55	4	9	10	2	10	10	2	10	11
Class 62 → Class 57	4	7	2	3	8	1	4	8	2
Class 62 → Class 74	31	13	14	32	9	16	32	10	17
Class 74 → Class 55	4	5	8	0	1	4	0	1	4
Class 74 → Class 57	7	6	7	6	7	8	6	7	9
Class 74 → Class 62	11	6	4	7	8	3	7	9	5
Total Intra-Component Errors	121	73	**69**	96	69	**58**	97	71	**63**

inherently encodes task-specific directional information. This is analogous to recent findings on task arithmetic Ilharco et al. (2023), where weight differences between fine-tuned models have been shown to encode semantic shifts without requiring feature-space linearity.

System-Level Efficiency. In class-incremental learning, inference requires extracting features for test samples and comparing them with the prototypes of all seen classes. Feature-level subtraction would demand two forward passes per sample (one for A_t and one for A_{shared}), resulting in doubled computational cost. Weight-space subtraction, by contrast, merges the two adapters into a single equivalent module, enabling a **single forward pass**. This efficiency advantage becomes increasingly critical as the number of tasks and classes grows.

Implicit Regularization and Empirical Superiority. Beyond efficiency, weight-space subtraction imposes a hard parameter constraint. It forces the shared adapter to capture common patterns while compelling the differential adapter to absorb only the residual, task-specific information. This restriction effectively shrinks the hypothesis space and prevents overfitting to the current task. This is a desirable inductive bias in incremental settings. To validate this empirically, we compare three variants on CIFAR-100 and ImageNet-R, as shown in Table 18. Our weight-level subtraction achieves slightly better $\bar{A}$ than feature-level subtraction on both datasets, along with competitive A_N and half the inference cost. This confirms that weight-space subtraction serves as an effective and efficient heuristic for residual extraction.

B.5 Limitation and Bad Cases

Despite its overall effectiveness, CCS faces a limitation from the two-stage inference bottleneck. If GT classes fall outside the Stage 1 top-k candidates, samples fail to trigger Stage 2 error correction. This typically occurs under severe domain shifts or extreme stylization. As shown in Figure 11, anthropomorphic features in a stylized "broccoli" and severe domain shifts in a rough "hyena" sketch completely overpower their intrinsic semantics. Relying on superficial shape biases, the model misclassifies them as a "mushroom" and a "missile", respectively. Because the GT class is absent

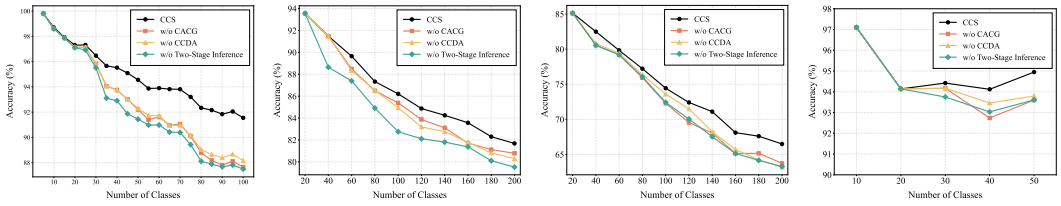
Table 15: Intra-component misclassification evolution on **ImageNet-R** (Target Component Classes: 8, 25, 145). We report the mutual prediction errors at representative incremental stages (Task 4, Task 7, and Task 9). CCS effectively mitigates cross-stage confusion from early stages and prevents severe error accumulation.

Misclassification ($A \rightarrow B$)	**Task 4** (Early Stage)			**Task 7** (Middle Stage)			**Task 9** (Final Stage)		
	EASE	MOS	**CCS**	EASE	MOS	**CCS**	EASE	MOS	**CCS**
Class 8 → Class 25	1	1	1	1	1	1	1	1	1
Class 8 → Class 145	0	0	0	6	2	3	6	2	3
Class 25 → Class 8	0	0	0	0	0	0	0	0	0
Class 25 → Class 145	0	0	0	6	0	0	5	0	0
Class 145 → Class 8	0	0	0	2	3	1	2	3	2
Class 145 → Class 25	0	0	0	0	3	1	0	2	1
Total Intra-Component Errors	**1**	**1**	**1**	15	9	**6**	14	8	**7**

Table 16: Intra-component misclassification evolution on **ImageNet-A** (Target Component Classes: 2, 71, 93, 151). We report the mutual prediction errors at representative incremental stages (Task 4, Task 7, and Task 9). CCS effectively mitigates cross-stage confusion from early stages and prevents severe error accumulation.

Misclassification ($A \rightarrow B$)	**Task 4** (Early Stage)			**Task 7** (Middle Stage)			**Task 9** (Final Stage)		
	EASE	MOS	**CCS**	EASE	MOS	**CCS**	EASE	MOS	**CCS**
Class 2 → Class 71	0	1	0	0	2	0	0	1	0
Class 2 → Class 93	8	4	2	7	3	2	6	2	1
Class 2 → Class 151	0	0	0	1	3	5	1	3	5
Class 71 → Class 2	0	1	0	0	0	0	0	0	0
Class 71 → Class 93	0	0	1	0	0	0	0	0	0
Class 71 → Class 151	0	0	0	0	1	1	0	1	1
Class 93 → Class 2	1	2	3	0	2	2	0	3	1
Class 93 → Class 71	0	1	0	0	1	0	0	0	0
Class 93 → Class 151	0	0	0	0	1	0	0	1	0
Class 151 → Class 2	0	0	0	1	1	0	1	0	0
Class 151 → Class 71	0	0	0	2	1	0	2	1	0
Class 151 → Class 93	0	0	0	3	3	1	3	3	0
Total Intra-Component Errors	9	9	**6**	14	18	**11**	13	15	**8**

from the top-k pool, the refinement mechanism is entirely bypassed. Addressing this limitation remains a key direction for future work.



(a) CIFAR-100 (b) ImageNet-R (c) ImageNet-A (d) VTAB

Figure 12: Stage-wise accuracy trends of the CCS framework and its ablated variants across four benchmark datasets. Note that "w/o Two-Stage Inference" is equivalent to the combination of "w/o CACG" and "w/o CCDA" .

C More Experimental Results

C.1 Ablation Study on Key Mechanisms

Figure 12 presents the stage-wise results of the ablation study on CIFAR100, ImageNet-R, ImageNet-A, and VTAB. Removing CACG leads to a noticeable drop in both average accuracy

Table 17: Intra-component misclassification evolution on **VTAB** (Target Component Classes: 1, 5, 33, 38). The statistics are recorded across Task 2, Task 3, and Task 4. The highly entangled pair (Class 33 and Class 38) experiences catastrophic confusion in EASE and MOS, whereas CCS dynamically suppresses these overlapping boundaries.

Misclassification ($A \rightarrow B$)	**Task 2** (Early Stage)			**Task 3** (Middle Stage)			**Task 4** (Final Stage)		
	EASE	MOS	**CCS**	EASE	MOS	**CCS**	EASE	MOS	**CCS**
Class 1 → Class 5	2	2	1	1	1	1	1	1	1
Class 1 → Class 33	0	0	0	2	2	3	2	1	3
Class 1 → Class 38	0	0	0	0	1	0	0	1	0
Class 5 → Class 1	4	0	4	3	0	0	3	0	2
Class 5 → Class 33	0	0	0	2	1	3	3	0	2
Class 5 → Class 38	0	0	0	0	2	2	0	2	2
Class 33 → Class 1	0	0	0	0	0	7	0	0	4
Class 33 → Class 38	0	0	0	14	25	10	15	29	8
Class 38 → Class 1	0	0	0	0	0	1	0	0	1
Class 38 → Class 5	0	0	0	0	0	0	0	0	1
Class 38 → Class 33	0	0	0	33	36	20	33	35	22
Total Intra-Component Errors	6	2	**5**	55	68	**47**	57	69	**46**

Table 18: **Comparison between feature-level and weight-level subtraction methods across two benchmark datasets.** The best and next best results are **bolded** and underlined, respectively. Our results are highlighted with the gray background.

Method	CIFAR-100		ImageNet-R		Inference Cost
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	
Base Adapter	92.90	88.15	85.60	80.78	1×
Feature-Level Subtraction	94.75	91.12	86.16	**81.87**	2×
Weight-Level Subtraction (Ours)	**94.84**	**91.55**	**86.49**	81.68	**1**×

and final accuracy, as random grouping fails to capture meaningful inter-class relationships. Replacing the differential adapter with a standard one also diminishes discriminability, confirming the effectiveness of our CCDA mechanism. The variant without both modules suffers the most significant degradation, highlighting the necessity of the entire two-stage inference pipeline. These results confirm that every proposed mechanism plays an essential role in achieving optimal performance.

C.2 Robustness across Random Seeds

To further validate the stability and reproducibility of our proposed CCS framework, we conduct repeated experiments across four benchmark datasets using three different random seeds (1024, 1993, and 2027). Table 19 reports the mean and standard deviation of both the average incremental accuracy ($\bar{\mathcal{A}}$) and the final accuracy ($\mathcal{A}_N$). The consistently low variance demonstrates that our method is highly robust to different initialization states and exemplar-free data orderings.

Table 19: Performance robustness of the CCS framework across three random seeds (1024, 1993, 2027). We report the mean and standard deviation for both average incremental accuracy ($\bar{\mathcal{A}}$) and final accuracy ($\mathcal{A}_N$).

Dataset	$\bar{\mathcal{A}}(\uparrow)$	$\mathcal{A}_N(\uparrow)$
CIFAR100 B0 Inc5	93.90 ± 0.82	90.04 ± 1.30
ImageNet-R B0 Inc20	85.49 ± 0.87	80.18 ± 1.29
ImageNet-A B0 Inc20	74.64 ± 0.12	65.54 ± 1.08
VTAB B0 Inc10	93.94 ± 0.87	92.67 ± 0.97

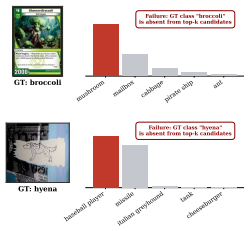


Figure 11: Illustration of typical bad cases.

Beyond performance metrics, we also investigate the structural stability of the dynamically constructed connected components. Figure 13 visualizes the number of identified components M generated under the three random seeds across the four benchmark datasets. As observed, the value of M remains remarkably consistent, fluctuating by at most one component across different initialization states, e.g., strictly bounding between 13 and 14 on ImageNet-R and between 15 and 16 on ImageNet-A. This structural consistency confirms that our CACG mechanism robustly captures the intrinsic semantic entanglement among overlapping classes, rather than being biased by random data permutations or initialization variances.

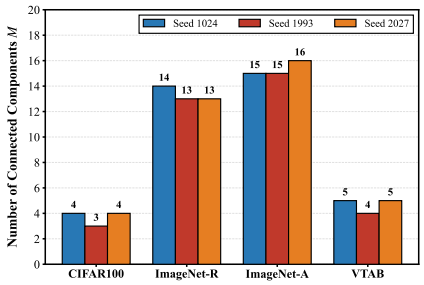


Figure 13: Number of connected components M under three random seeds across four benchmark datasets.

C.3 Baseline Performance of the Pre-Trained Model

We report this baseline in Table 20, where the baseline (PTM) evaluates the zero-shot or direct classification performance of the pre-trained model on the entire dataset without incremental adaptation. As shown in the table, CCS significantly outperforms it across all datasets (e.g., **+32.16%** on ImageNet-R), confirming that CCS learns new task-specific knowledge rather than merely inheriting pre-trained representations.

Table 20: Performance of the PTM baseline and CCS.

Method	CIFAR100	ImageNet-R	ImageNet-A	VTAB
Baseline (PTM)	81.26	54.33	48.45	84.46
CCS (Ours)	**94.84** (+13.58)	**86.49** (+32.16)	**74.48** (+26.03)	**94.95** (+10.49)