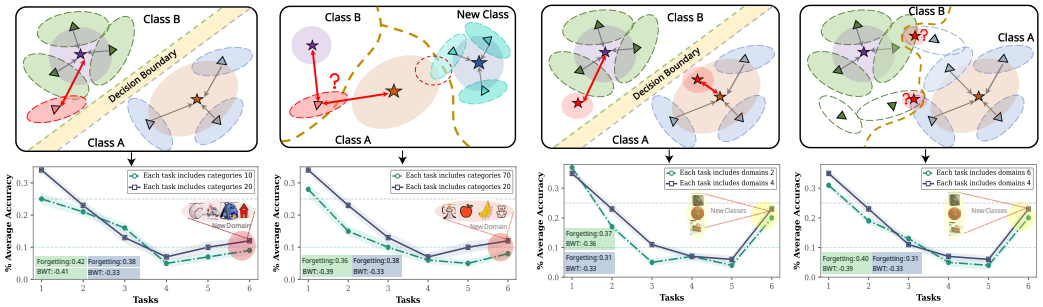
CDC-OIL: CLASS-DOMAIN COUPLED ONLINE INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world data streams often exhibit simultaneous class increment and domain shift, while existing continual learning settings typically study these two sources of non-stationarity in isolation. So we introduce Class-Domain Coupled Online Incremental Learning (CDC-OIL), where class and domain spaces evolve jointly or separately under single-pass data access and ambiguous task boundaries. We further identify a bidirectional interaction between class increment and domain shift. Class increment encourages model to refine semantic features but compress the feature space, increasing sensitivity to domain shift. Conversely, domain shift can enrich cross-domain semantics, but excessive shifts distort class distributions and hinder new-class discrimination. To address this challenge, we propose Class-Domain Decoupled Extensible Gaussian Mixture Models (CDD-EGMMs), which factorize feature distributions into class-specific semantic mixtures and domain shifts shared across classes. This compositional memory supports online updates and generative replay by recombining semantic and domain components without storing raw data. A Gaussian Representation Constraint further promotes class separability as the label space expands. Experiments on four benchmarks demonstrate that CDD-EGMMs consistently improves accuracy and reduces forgetting compared with representative continual learning baselines.



(a) The influence of class increment on domain shift (b) The influence of domain shift on class increment

Figure 1: Bidirectional coupling between class expansion and domain shift in continual learning. Class increment compresses the feature space and makes decision boundaries more fragile under domain shift, while domain shift further distorts class distributions and aggravates feature overlap.

1 INTRODUCTION

Real-world intelligent systems (Wasilewski et al., 2024; Hubbertz et al., 2025) are often required to process open and dynamic data streams, where both the label space and the input distribution continuously evolve. Continual learning (Csaba et al., 2024; Seo et al., 2025) provides a fundamental paradigm for adapting to such non-stationary data, and existing studies typically address this challenge through two settings: class-incremental learning (CIL) (Liu et al., 2025; Szatkowski et al., 2025) and domain-incremental learning (DIL) (Lamers et al., 2023; Wang et al., 2024a). Specifically, CIL focuses on learning new classes from an expanding label space while preserving the recognition ability for previously learned ones, thereby mitigating catastrophic forgetting. In contrast, DIL aims to enable models to adapt to distributional changes across domains under a relatively fixed label space. However, in practical scenarios, these two sources of non-stationarity rarely exist in isolation:

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.

the emergence of new classes is often accompanied by domain shift, and domain shift may further reshape the decision boundaries between classes. Such coupled evolution makes isolated CIL or DIL settings insufficient for faithfully modeling real-world online data streams (Shi & Wang, 2023), and studies of such combined settings remain limited.

Prior works such as CIDA and related cross-domain continual learning methods explore settings that combine class increment and domain shift (Kundu et al., 2020; Simon et al., 2022b; Xie et al., 2022), emphasizing the model's ability to adapt and generalize across both known and unknown domains. DisCo (Chen & Zhou, 2025) investigates how to leverage different domains to enhance CIL. However, these methods mainly examine how domain variation can benefit CIL, rather than how class increment and domain shift interfere with each other. Building on this, XEED (Choi et al., 2026) further investigates how models handle class-incremental data across different domains. Overall, existing studies primarily focus on within-class cross-domain knowledge transfer, overlooking how to share domain knowledge across classes while reducing interference with class semantics. This limitation is critical in real-world applications such as autonomous driving, medical imaging, and remote sensing, where new categories and changing domains often emerge simultaneously (Toldo et al., 2024). Therefore, a more realistic continual learning framework should explicitly model the bidirectional interaction between class increment and domain shift, rather than considering only how domain variation benefits CIL.

We argue that class increment and domain shift interact bidirectionally, potentially facilitating or hindering each other. Class increment promotes the model to learn finer-grained semantic features, which can enhance discrimination and improve robustness to partial domain shift. However, it also compresses the feature space, degrading decision boundaries, feature overlap and amplifying the negative impact of domain shift. Conversely, domain shift enriches cross-domain semantics, and helps learn domain-invariant features for CIL, but excessive shifts may perturb intra-class distribution, blur inter-class boundaries, and intensify forgetting, as illustrated in Fig. 1. Moreover, most existing studies are based on offline learning setting with clearly defined task boundaries, which differ significantly from continual learning scenarios in the real-world where task boundaries are ambiguous and data arrive as a continuous stream.

Table 1: Comparison between CDC-OIL and existing continual learning settings.

Setting	Class Increment	Domain Shift	Online Single-pass	Ambiguous Boundary	Class-Domain Coupling
CIL	✓	×	Optional	×	×
DIL	×	✓	Optional	×	×
CIDA/XIL	✓	✓	×	×	Partially
Online CL	✓	Optional	✓	✓	Limited
CDC-OIL	✓	✓	✓	✓	✓

These observations motivate a central question: **How can a model exploit the mutual benefits of evolving classes and domains while suppressing their interference in boundary-ambiguous data streams?** To study this question, we introduce Class-Domain Coupled Online Incremental Learning (CDC-OIL), where classes and domains evolve jointly or separately under single-pass data access without explicit task boundaries. The learner must adapt to these changes while retaining previously acquired knowledge. Table 1 compares CDC-OIL with related settings. We propose Class-Domain Decoupled Extensible Gaussian Mixture Models (CDD-EGMMs), a compositional distributional memory that separates class-specific semantic mixtures from domain shifts shared across classes (Fig. 2). This factorization enables class and domain knowledge to be maintained separately and reused through composition. Our Dynamic Decoupled Adaptation and Generative Joint Replay strategy updates these components online and recombines them to generate replay features without storing raw historical data. Finally, a Gaussian Representation Constraint promotes geometric separation among class distributions, reducing feature overlap as the label space expands.

Our contributions are summarized as follows: (1) We introduce CDC-OIL, a new online continual learning setting that captures the coupled evolution of class expansion and domain shift under single-pass data access and ambiguous task boundaries. (2) We propose CDD-EGMMs, which factorizes feature distributions into semantic mixtures and domain shifts, enabling efficient class–domain updates and reducing interference in evolving streams. (3) We conduct extensive experiments and analyze the limitations of existing CIL baselines under the CDC-OIL setting.

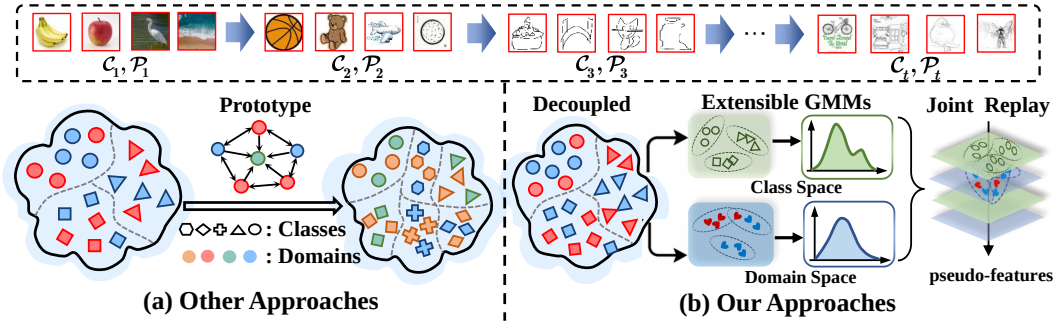


Figure 2: (a) Existing methods model class and domain prototypes in a shared feature space, (b) We decouple class space (domain-invariant semantics) and domain space (domain-specific variations).

2 Related Works

Class-Incremental Learning. CIL primarily mitigates catastrophic forgetting through three paradigms: memory replay, regularization, and parameter isolation (Wang et al., 2024b). Representative methods based on memory replay include rehearsal-based methods, which use a memory buffer to store exemplars (Rebuffi et al., 2017; Caccia et al., 2020; Zhou et al., 2022). Regularization-based approaches implement weight regularization on important parameters (Kirkpatrick et al., 2017; Lin et al., 2022) or knowledge distillation to preserve crucial outputs (Wang et al., 2022a). Parameter isolation include: L2P (Wang et al., 2022b) guide pretrained Transformers with task-specific prompts to balance shared and task-specific knowledge.

Domain-Incremental Learning. Multiple studies focus on addressing catastrophic forgetting in DIL (Csaba et al., 2024). Some methods restrict model plasticity to balance adaptation to new domains and retention of old ones, which may lead to a zero-sum trade-off between old and new domains (Kirkpatrick et al., 2017; Akyürek et al., 2021). Recent works further mitigate forgetting by preserving general pretrained features (Zhou et al., 2025), rebalancing minority classes via simplex-based memory (Li et al., 2025) or decomposing complex domain adaptation into hierarchical prompt aggregation (Xu et al., 2024). Although these methods improve generalization across domains, they mainly focus on domain evolution under a relatively fixed label space, leaving the interaction between domain shift and class expansion insufficiently explored.

Domain Shift in Class-Incremental Learning. There is limited research on the joint study of CIL and DIL. CIDA (Kundu et al., 2020) integrates class-incremental learning with source and target domain adaptation and is specifically designed for open-set environments. For general incremental learning (GIL), Xie et al. (2022) model intra-class domain structure with von Mises–Fisher mixtures to handle evolving class and domain distributions. A cross-domain continual learning framework based on Mahalanobis similarity (MSL) (Simon et al., 2022a) and exponential moving average distillation addresses catastrophic forgetting under both class-incremental and domain shift settings. XEED (Choi et al., 2026) achieves knowledge transfer under joint class and domain expansion through domain-specific prompt tuning and generative prototype expansion, without requiring cross-domain data sharing. Different from DisCo and XEED, which primarily exploit domain variation to improve CIL, CDC-OIL treats class expansion and domain shift as two coupled and evolving sources of non-stationarity. Accordingly, our method explicitly factorizes class semantics and domain shifts instead of modeling them as entangled prototypes.

3 Model

3.1 Problem Setup and Model Overview

In this section, we formalize the Class-Domain Coupled Online Incremental Learning (CDC-OIL) task within data streaming and establish its mathematical formulation. In CDC-OIL, data arrives as a stream and can be observed by the model only once, with only a small number of samples allowed to be stored. Meanwhile, the label space and domain distribution continuously expand over time. For stream construction, the data may be generated from a sequence of latent tasks $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T$. However, during training, the learner only observes a continuous stream of mini-batches $\mathcal{D}_t = \{(x_i^t, y_i^t, d_i^t)\}_{i=1}^{N_t}$, where no task identity or task boundary is provided. And y_i^t belongs

to an evolving label space $\mathcal{C}_t$ and x_i^t represents the input sample from a shifting domain distribution $\mathcal{P}_t$, d_i^t denotes the corresponding domain label. Note that the total label and domain space observed up to time is represented by:

$$\mathcal{C}_t = \mathcal{C}_{1:t-1} \cup \mathcal{C}^{(\text{new})}, \qquad \mathcal{P}_t = \mathcal{P}_{1:t-1} \cup \mathcal{P}^{(\text{new})} \tag{1}$$

and $\mathcal{C}^{(\text{new})} \cap \mathcal{C}_{1:t-1} = \emptyset, \mathcal{P}^{(\text{new})} \cap \mathcal{P}_{1:t-1} = \emptyset$, the $\mathcal{T}_t = \langle \mathcal{C}_t, \mathcal{P}_t \rangle$ includes four cases: new class/old domain, old class/new domain, old class/old domain, and new class/new domain. Furthermore, within task $\mathcal{T}_t$, the samples are unordered.

The proposed CDD-EGMMs method aims to jointly address three key challenges in CDC-OIL. (1) how to enable effective sharing of semantic and domain knowledge under class expansion and domain shift, (2) how to prevent feature overlap and decision boundary degradation caused by unbounded class–domain expansion, (3) how to retain knowledge in an online setting with ambiguous task boundaries and limited storage to mitigate catastrophic forgetting. The CDD-EGMMs, which explicitly models semantic invariance and domain specificity. This design promotes knowledge sharing while reducing overlap from feature space compression and suppressing interference from new domains. Furthermore, we design a Dynamic Decoupled Adaptation and Generative Joint Replay strategy for adaptive updates under evolving data distributions. We also introduce a Gaussian Representation Constraint that enforces geometric separability between new and old class centers using Mahalanobis distance and center separation. The framework improves generalization to unseen classes and domains without storing raw data.

3.2 Class–Domain Decoupled Extensible GMMs

Existing methods mainly use prototypes to represent class centers for balancing class expansion and domain shift. However, the feature distribution of a class often exhibits complex multimodal patterns, especially in multi-domain settings, rather than a single distribution. The single prototype is insufficient to capture such complex distributions. In contrast, Gaussian Mixture Models (GMM) better model multimodal distributions and provide stronger representation ability. Therefore, we propose to assign a GMM for each learned foreground class (Zhu et al.).

To alleviate feature overlap caused by feature space compression under domain shift and class expansion, we design the CDD-EGMMs. By separating the domain-specific shift from the semantic core, the model can stably capture class semantics across domains even in a constrained feature space, thereby mitigating catastrophic forgetting. Specifically, at time t, the feature extractor $f_t(\cdot)$ extracts foreground features $\boldsymbol{z}_t = f_t(x_t) \in \mathbb{R}^m$ from $\mathcal{D}_t$. We then use CDD-EGMMs to estimate the feature distribution in feature space. And the feature distribution of each class $c \in \mathcal{C}_t$ is represented as a linear combination of K Gaussian components:

$$p_t(\boldsymbol{z}_t | y_t = c, d_t; \phi_{c,t}) = \sum_{k=1}^{K} \boldsymbol{\pi}_{c,k} \mathcal{N}\left(\boldsymbol{z}_t; \boldsymbol{\mu}_{c,k} + A\boldsymbol{u}_d, \boldsymbol{\Sigma}_{c,k}\right), \sum_{k=1}^{K} \boldsymbol{\pi}_{c,k} \quad = 1, \quad \boldsymbol{\pi}_{c,k} \geq 0 \tag{2}$$

where $\phi_{c,t} = \{\boldsymbol{\pi}_{c,k}, \boldsymbol{\mu}_{c,k}, A\boldsymbol{u}_d, \boldsymbol{\Sigma}_{c,k}\}$ is the parameters of CDD-EGMMs at time t. It is worth noting that $\boldsymbol{\mu}_{c,k} \in \mathbb{R}^m$ denotes the domain-invariant semantic feature within the same class, and $A\boldsymbol{u}_d$ represents the cross-class domain shift term that models domain-specific variations, $A \in \mathbb{R}^{m \times r}$ denotes a shared low-rank domain basis and $u_d \in \mathbb{R}^r$ denotes the domain-specific coefficient. To ensure stable online updates, we initialize A using feature residuals from the initial stream segment and keep it fixed during online learning. Decoupling them into independent representation spaces facilitates efficient sharing of class and domain knowledge, improving model robustness. It also mitigates feature overlap caused by continuous class and domain expansion, thereby preventing catastrophic forgetting. Moreover, after removing the effect of domain shift, the canonical distribution of class c is defined as:

$$p_t(\tilde{\boldsymbol{z}}_t | y_t = c, d_t) = \sum_{k=1}^{K_c} \boldsymbol{\pi}_{c,k} \mathcal{N}(\tilde{\boldsymbol{z}}_t; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k}), \tilde{\boldsymbol{z}}_t \quad = \boldsymbol{z}_t - A\boldsymbol{u}_d \tag{3}$$

3.3 Dynamic Decoupled Adaptation and Generative Joint Replay

In data streams, updates should be handled differently for four typical cases. For samples from known classes and known domains, the model should maintain stable representations. For samples

from known classes but new domains, the model needs to adjust domain shifts to align the semantic space. For samples from new classes, the model must expand its structure to capture unseen semantic patterns. We propose a Dynamic Decoupled Adaptation and Generative Joint Replay to retain prior knowledge without storing historical samples. For the dataset $\mathcal{D}_t$ at time t, if sample's class label $y_t \in \mathcal{C}_{t-1}$ and $d_t \in \mathcal{P}_{t-1}$ the sample is classified as an old-class old-domain sample, the model updates its parameters to preserve previously learned knowledge. Since the class label y_t and domain label d_t are known, the model does not need to perform joint search over class and domain sets, but only computes responsibilities within the Gaussian components of the true class y_t. Specifically, given the domain state u_d^{t-1} from the previous time step, the domain-corrected representation is defined as: $\tilde{\boldsymbol{z}}_t = \boldsymbol{z}_t - A\boldsymbol{u}_d^{(t-1)}$, The posterior responsibility of the sample belonging to the k-th Gaussian component of class y_t is given by:

$$\boldsymbol{r}_{t,k} = p_t(k \mid \tilde{\boldsymbol{z}}_t, y_t) = \frac{\boldsymbol{\pi}_{y_t,k}\mathcal{N}(\tilde{\boldsymbol{z}}_t; \boldsymbol{\mu}_{y_t,k}, \boldsymbol{\Sigma}_{y_t,k})}{\sum_j \boldsymbol{\pi}_{y_t,j}\mathcal{N}(\tilde{\boldsymbol{z}}_t; \boldsymbol{\mu}_{y_t,j}, \boldsymbol{\Sigma}_{y_t,j})} \tag{4}$$

We use online updates of the domain shift term, for the current domain d_t the objective function is defined as:

$$\xi(\boldsymbol{u}_d^{(t)}) = \sum_{k=1}^{K_{y_t}} \boldsymbol{r}_{t,k}\|\boldsymbol{z}_t - (\boldsymbol{\mu}_{y_t,k} + A\boldsymbol{u}_d^{(t)})\|^2_{\boldsymbol{\Sigma}^{-1}_{y_t,k}} + \lambda_u\|\boldsymbol{u}_d^{(t)}\|_2^2 \tag{5}$$

where λ_u is a weight. We use one-step gradient descent, and $\boldsymbol{u}_d^{(t)}$ is updated incrementally as:

$$\boldsymbol{u}_d^{(t)} = \boldsymbol{u}_d^{(t-1)} + \eta_t \left[\sum_{k=1}^{K_{y_t}} r_{t,k}A^\top\boldsymbol{\Sigma}^{-1}_{y_t,k}(\boldsymbol{z}_t - \boldsymbol{\mu}_{y_t,k} - A\boldsymbol{u}_d^{(t-1)}) - \lambda_u\boldsymbol{u}_d^{(t-1)}\right] \tag{6}$$

where η_t is the learning rate, the derivations of Eq. 5–6 are provided in Appendix A.1. Furthermore, we update the CDD-EGMMs' parameters of the true class, enabling joint online adaptation of class distributions and domain states. Unlike traditional EM, which requires multiple passes over data for batch optimization, we adopt an online update strategy based on sufficient statistics, enabling the model to gradually approach the optimum under a single-pass setting. For the $\mathcal{D}_t$, we first correct the feature using the updated domain state: $\tilde{z}_t = z_t - Au_{d_t}^{(t)}$, meanwhile, the responsibility of each Gaussian component is computed using Eq. 4. Subsequently, we perform incremental updates for each Gaussian component k of each class c:

$$N_{c,k}^{(t)} = \rho N_{c,k}^{(t-1)} + r_{t,k}\delta_{t,c}, \quad S_{c,k}^{(t)} = \rho S_{c,k}^{(t-1)} + r_{t,k}\tilde{z}_t\delta_{t,c}, \quad Q_{c,k}^{(t)} = \rho Q_{c,k}^{(t-1)} + r_{t,k}\tilde{z}_t\tilde{z}_t^\top\delta_{t,c} \tag{7}$$

where $N_{c,k}^{(t)}$ is responsibility-weighted sample count, and $k = 1, \ldots, K$, $S_{c,k}^{(t)}$ is first-order moment and $Q_{c,k}^{(t)}$ is second-order moment. $\rho \in (0, 1]$ is a forgetting factor used to gradually decay the influence of historical information in non-stationary data streams, $\delta_{t,c} = \mathbb{I}(y_t = c)$ is indicator function. Based on the above statistics, closed-form updates for the parameters of each Gaussian component within a class can be obtained:

$$\boldsymbol{\mu}_{c,k}^{(t)} = \frac{S_{c,k}^{(t)}}{N_{c,k}^{(t)}}, \quad \boldsymbol{\Sigma}_{c,k}^{(t)} = \frac{Q_{c,k}^{(t)}}{N_{c,k}^{(t)}} - \boldsymbol{\mu}_{c,k}^{(t)}\boldsymbol{\mu}_{c,k}^{(t)\top}, \quad \boldsymbol{\pi}_{c,k}^{(t)} = \frac{N_{c,k}^{(t)}}{\sum_{j=1}^{K_c} N_{c,j}^{(t)}} \tag{8}$$

Through these updates, the model can continuously approximate class distributions without storing historical samples. If sample's class label $y_t \in \mathcal{C}_{t-1}$ and $d_t \notin \mathcal{P}_{t-1}$, the sample is classified as an old-class/new-domain sample. In this case, we initialize a new domain shift term to expand the domain set by solving a regularized weighted least squares problem based on the current sample.

$$u_d^{new} = \left(\sum_{k=1}^{K} r_{t,k}A^\top\boldsymbol{\Sigma}^{-1}_{y_t,k}A + \lambda_u I\right)^{-1}\left(\sum_{k=1}^{K} r_{t,k}A^\top\boldsymbol{\Sigma}^{-1}_{y_t,k}(\boldsymbol{z}_t - \boldsymbol{\mu}_{y_t,k})\right) \tag{9}$$

where I is identity matrix. The new domain is then added into the domain set $\mathcal{P}_t = \mathcal{P}_{t-1} \cup d_t$. This mechanism enables the model to dynamically expand its domain representation capacity without affecting the stability of class semantics. As the number of domain prototypes increases during online continual learning, we introduce a KL divergence-based prototype merging mechanism to

control model complexity. For any two domain prototypes μ_d and $\mu_{d'}$, their similarity is defined as: $D_{KL} = \frac{1}{2}\left(KL(\mu_d||\mu_{d'}) + KL(\mu_{d'}||\mu_d)\right)$. If $D_{KL} < \tau_{\text{merge}}$, where τ_{merge} is a merging threshold, the two components are considered highly similar. We then merge them via moment matching to keep the model size manageable. If sample's class label $y_t \notin \mathcal{C}_{t-1}$ and $d_t \in \mathcal{P}_{t-1}$, this indicates the emergence of a new-class and old-domain, we first remove the corresponding domain shift and initialize the semantic Gaussian components in the canonical space and initialize $\phi_{c,t}$ for class y_t:

$$\boldsymbol{\mu}_{y_t,k}^{(new)} = \tilde{\boldsymbol{z}}_t + \epsilon_k, \quad \boldsymbol{\Sigma}_{c,k}^{(new)} = \sigma^2 I, \quad \boldsymbol{\pi}_{c,k}^{(new)} = \frac{1}{K}, \quad k = 1, \ldots, K \tag{10}$$

where $\sigma^2 I$ is the covariance matrix, $\sigma^2 \in [0.05, 0.1]$. And the sufficient statistics are initialized as:

$$N_{c,k}^{(new)} = \frac{1}{K}, S_{c,k}^{(new)} = N_{c,k}^{(new)} \boldsymbol{\mu}_{c,k}^{(new)}, Q_{c,k}^{(new)} = N_{c,k}^{(new)} \left(\boldsymbol{\Sigma}_{c,k}^{(new)} + \boldsymbol{\mu}_{c,k}^{(new)} \boldsymbol{\mu}_{c,k}^{(new)\top}\right) \tag{11}$$

When samples of the same class arrive, we incrementally update them using Eq. 7 and 8. When a new-class and a new-domain appear simultaneously, we set $u_d^{(0)} = \mathbf{0}$ and initializing the new-class Gaussian components from feature $z_i = f_t(x_i)$. The new domain offset is then estimated from the residuals between current features and their assigned semantic components.

$$u_d^{(0)} = (A^\top A + \lambda I)^{-1} A^\top \left(\frac{1}{|\mathcal{B}_d|} \sum_{i \in \mathcal{B}_d} (z_i - \mu_{y_i,k_i^*}^{(0)}) \right) \tag{12}$$

where $k_i^* = \arg\min_k \|z_i - \mu_{y_i,k}^{(0)}\|_2^2$. We then refine the new-class Gaussians with corrected representations $\tilde{z}_i = z_i - A u_d^{(0)}$, avoiding any prior assumption about the new class or domain.

Finally, we introduce a lightweight classifier $h(\cdot)$ for fast prediction on new samples. In continual learning, since historical data is inaccessible, we resample from the stored CDD-EGMMs to construct the training set for the classifier. For replay, we first sample a semantic component k according to $\pi_{c,k}$, draw a canonical semantic feature $\tilde{z} \sim \mathcal{N}(\boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k})$, and then combine it with a sampled domain shift $A\boldsymbol{u}_d$ to obtain $\boldsymbol{z} = \tilde{\boldsymbol{z}} + A\boldsymbol{u}_d$. Since pseudo-features are generated in the latent space, they are used only to update the $h(\cdot)$. The $f_t(\cdot)$ is updated using real incoming samples together with the GRC loss. For each historical class $c \in \mathcal{C}_t$, we sample pseudo-features from its learned class-conditional distribution: $\boldsymbol{z}_i \sim p(\boldsymbol{z}|y = c, d; \phi_c)$, we construct the training set using resampled features from all existing classes and real features from new samples.

$$\mathcal{D}_{cls} = \bigcup_{c \in \mathcal{C}_t} \{(\boldsymbol{z}_i, c)\}_{i=1}^{M} \cup \{(\boldsymbol{z}_j, y_j)\}_{j=1}^{N_{new}} \tag{13}$$

After predicting the incoming samples and revealing their labels, we train on real and replay features using

$$\mathcal{L}_{cls} = -\frac{1}{|\mathcal{D}_{cls}^{t+1}|} \sum_{(\boldsymbol{z},y) \in \mathcal{D}_{cls}^{t+1}} \log h_y(\boldsymbol{z}).$$

Replay features update only h, while real features also update f.

3.4 Gaussian Representation Constraint

Previous methods (Zhu et al.) use Gaussian statistics as compact distributional memory for old classes. In CDC-OIL, new class representations may invade existing class regions under class-domain coupled evolution, causing decision boundary degradation. To address this, we propose a Gaussian Representation Constraint (GRC), where class-domain Gaussian components serve as geometric anchors to promote compact new representations and maintain margins from existing class distributions.

At time $t+1$, each incoming sample x_t is first mapped into the feature space by the trainable feature extractor: $z_{t+1} = f(x_{t+1})$. To remove the domain-specific variation, we obtain the domain-corrected representation: $\tilde{z}_{t+1} = z_{t+1} - A u_d$. Given the stored Gaussian memory of class c, i.e., (μ_c, Σ_c), the distance between the current representation and the class distribution is measured by:

$$d_M(\tilde{z}_{t+1}, \boldsymbol{\mu}_c) = \min_{k \in \{1,\ldots,K\}} (\tilde{z}_{t+1} - \mu_{c,k})^\top \Sigma_{c,k}^{-1} (\tilde{z}_{t+1} - \mu_{c,k}) \tag{14}$$

For classes first observed in the current mini-batch, GRC pulls representations toward their own Gaussian prototypes and separates them from previously observed classes:

$$\mathcal{L}_{GRC} = \frac{1}{|\mathcal{D}_{t+1}^{new}|} \sum_{(x_i,y_i,d_i)\in\mathcal{D}_{t+1}^{new}} \max\left(0, m + d_M(\tilde{z}_i, \mu_{y_i}) - \min_{c'\in\mathcal{C}_t, c'\neq y_i} d_M(\tilde{z}_i, \mu_{c'})\right) \tag{15}$$

where $m > 0$ is a predefined margin. The final objective is:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{GRC} \tag{16}$$

3.5 Analysis of CDD-EGMMs

To explain why class–domain decoupling reduces interference, consider a feature decomposition $z = \mu_c + Au_d + \epsilon$, where μ_c is the domain-invariant class semantics and Au_d is the domain-specific shift. If features are modeled in an entangled space, the estimated class center becomes $\bar{\mu}_c = \mathbb{E}[z|y = c] = \mu_c + A\mathbb{E}[u_d|y = c]$, which is biased by the domain distribution. Thus, when new domains arrive, changes in $\mathbb{E}[u_d|y = c]$ directly perturb old class centers and cause semantic drift. In contrast, CDD-EGMMs constructs the canonical representation $\tilde{z} = z - Au_d$, yielding $\mathbb{E}[\tilde{z}|y = c] = \mu_c$. Therefore, the class semantic estimator is unbiased with respect to domain shifts, and new-domain adaptation mainly updates u_d instead of overwriting μ_c, reducing interference between domain adaptation and class retention. Detailed proofs are provided in Appendix A.2.

4 Experiments

4.1 Experiments Setup

Datasets. To enable proper evaluation under the proposed CDC-OIL setting, we assume that class and domain labels become available after testing (i.e., an online incremental learning protocol). We use widely adopted datasets for domain generalization. DomainNet (Dayal et al., 2025; Peng et al., 2019) exhibits large domain style differences and severe label shift. We combine CIFAR-10 with CIFAR-10-C (Krizhevsky et al., 2009; Hendrycks & Dietterich, 2019) to simulate domain shift of varying severity. In addition, we use the CORe50 (Lomonaco & Maltoni, 2017) dataset, which contains sequential visual object categories to evaluate continual learning in object recognition. Detailed experimental settings are provided in Appendix B.2.

Evaluation Metrics. We evaluate using continual learning accuracy metrics: (1) Average Accuracy A_T over all tasks after training; (2) the Forgetting Measure F_T following (Zhang et al., 2026); (3) we also introduce Relative Forgetting (Wang et al., 2024c) RF to evaluate model stability, and the formulas for calculating A_T, F_T, RF and BWT are detailed in Appendix B.4.

Baselines. Since our proposed setting, CDC-OIL, is novel and lacks directly comparable methods under identical conditions, we adopt several classic continual learning methods and include representative recent works addressing the joint challenges of class- and domain-incremental learning. These include traditional methods such as ER (Rolnick et al., 2019), DER++ (Buzzega et al., 2020), ER-ACE (Caccia et al., 2021), OCM (Guo et al., 2022), GSA (Guo et al., 2023), OnePro (Wei et al., 2023), DisCo (Chen & Zhou, 2025) and XEED (Choi et al., 2026). Details are provided in Appendix B.1 and Appendix B.3.

Implementation Details. All compared methods use ResNet-18 as the backbone, which is randomly initialized and trained only on the online training stream. For replay-based methods such as ER, the memory buffer size is set to 500. We employ an AdamW optimizer with an initial learning rate of 0.005. For CIFAR-10 and CORe50, we set the batch size to 10, while for DomainNet it is set to 64, and all experiments use a single epoch. The number of Gaussian components K is set to 6, with hyperparameters $\alpha = 0.85$ and $\tau_{\text{merge}} = 0.2$. Detailed ablation results are provided in the appendix. All experiments are conducted in Python 3.8 using a single NVIDIA A100 GPU. Results in Table 2 are reported as mean $\pm$ standard deviation over five random seeds. The task-wise accuracy matrices in Figs. 3, 5, and 6 show results from a single randomly selected run.

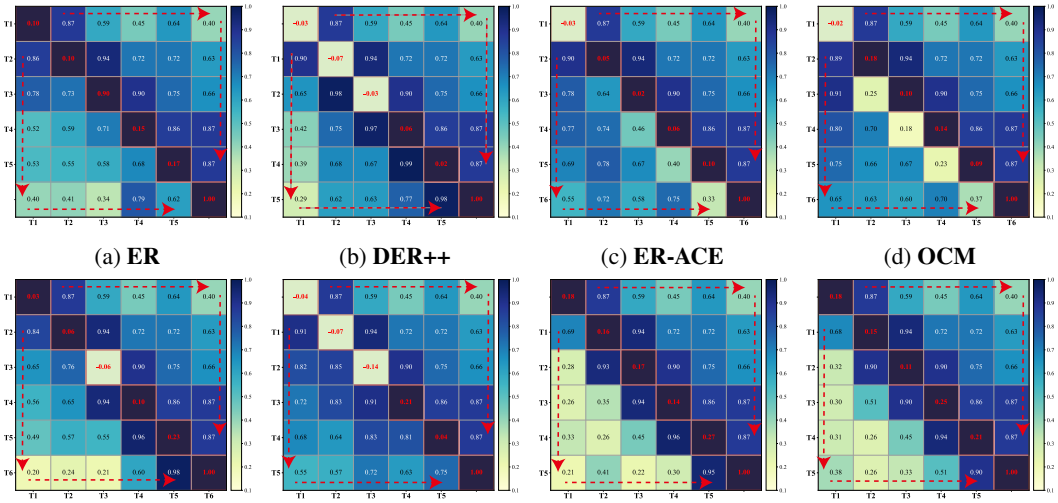
4.2 EXPERIMENTS RESULTS

Table 2 compares CDD-EGMMs with representative continual learning baselines on the four benchmarks, where the symbol "-" denotes an out-of-memory error. CDD-EGMMs achieves the highest average accuracy on all datasets and the lowest forgetting on most of them, demonstrating a better stability–plasticity balance. These results indicate that decoupling class-discriminative semantics from domain-specific shifts enables selective parameter updates and effectively mitigates catastrophic forgetting under joint class-domain evolution.

Table 2: Comparison results on CIFAR-10/CIFAR-10-C, CIFAR-100/CIFAR-100-C, DomainNet, and CORe50. Best results are highlighted in bold, and second-best results are underlined. We report mean ± standard deviation over five random seeds.

Method	CIFAR-10/CIFAR-10-C			CIFAR-100/CIFAR-100-C			DomainNet			CORe50		
	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$
ER	$51.82_{\pm 0.47}$	$0.287_{\pm 0.13}$	$0.296_{\pm 0.04}$	$\underline{36.27}_{\pm 0.35}$	$0.461_{\pm 0.16}$	$0.533_{\pm 0.27}$	$14.46_{\pm 1.39}$	$0.595_{\pm 0.21}$	$0.774_{\pm 0.04}$	$56.25_{\pm 1.16}$	$\underline{0.189}_{\pm 0.15}$	$\underline{0.215}_{\pm 0.11}$
DER++	$\underline{66.23}_{\pm 1.05}$	$0.387_{\pm 0.15}$	$0.321_{\pm 0.12}$	$27.98_{\pm 0.42}$	$0.386_{\pm 0.10}$	$0.553_{\pm 0.09}$	$\underline{21.35}_{\pm 1.70}$	$0.652_{\pm 0.11}$	$0.721_{\pm 0.05}$	$58.97_{\pm 1.45}$	$0.251_{\pm 0.03}$	$0.2333_{\pm 0.21}$
ER-ACE	$60.12_{\pm 1.11}$	$\underline{0.121}_{\pm 0.11}$	$\underline{0.110}_{\pm 0.27}$	$30.10_{\pm 0.55}$	$\mathbf{0.205}_{\pm 0.07}$	$\underline{0.252}_{\pm 0.15}$	$16.44_{\pm 2.08}$	$\underline{0.345}_{\pm 0.05}$	$\underline{0.599}_{\pm 0.14}$	$\underline{59.09}_{\pm 1.34}$	$0.293_{\pm 0.23}$	$0.281_{\pm 0.06}$
OCM	$59.76_{\pm 1.06}$	$0.503_{\pm 0.03}$	$0.543_{\pm 0.14}$	$30.13_{\pm 0.62}$	$0.247_{\pm 0.12}$	$0.275_{\pm 0.09}$	$13.27_{\pm 3.22}$	$\mathbf{0.316}_{\pm 0.28}$	$\mathbf{0.343}_{\pm 0.07}$	$54.93_{\pm 0.89}$	$0.230_{\pm 0.01}$	$0.239_{\pm 0.02}$
GSA	$45.98_{\pm 1.26}$	$0.304_{\pm 0.02}$	$0.299_{\pm 0.01}$	$29.31_{\pm 0.71}$	$0.465_{\pm 0.06}$	$0.595_{\pm 0.12}$	$15.77_{\pm 0.69}$	$0.667_{\pm 0.1}$	$0.779_{\pm 0.03}$	-	-	-
OnPro	$64.32_{\pm 2.55}$	$0.171_{\pm 0.11}$	$0.192_{\pm 0.07}$	$25.23_{\pm 0.32}$	$0.277_{\pm 0.05}$	$0.331_{\pm 0.08}$	-	-	-	$57.44_{\pm 1.88}$	$0.321_{\pm 0.11}$	$0.286_{\pm 0.03}$
DisCo	$59.69_{\pm 0.85}$	$0.504_{\pm 0.01}$	$0.887_{\pm 0.00}$	$29.06_{\pm 1.88}$	$0.280_{\pm 0.23}$	$0.713_{\pm 0.09}$	$20.15_{\pm 2.13}$	$0.611_{\pm 0.01}$	$0.818_{\pm 0.04}$	$57.33_{\pm 3.96}$	$0.412_{\pm 0.15}$	$0.418_{\pm 0.17}$
XEED	$42.89_{\pm 3.35}$	$0.714_{\pm 0.07}$	$0.800_{\pm 0.02}$	$31.60_{\pm 0.54}$	$0.296_{\pm 0.20}$	$0.473_{\pm 0.13}$	$19.41_{\pm 1.74}$	$0.579_{\pm 0.07}$	$0.793_{\pm 0.07}$	$56.87_{\pm 5.31}$	$0.275_{\pm 0.16}$	$0.433_{\pm 0.14}$
Ours	$\mathbf{70.17}_{\pm 1.42}$	$\mathbf{0.116}_{\pm 0.12}$	$\mathbf{0.109}_{\pm 0.11}$	$\mathbf{37.45}_{\pm 0.38}$	$\underline{0.230}_{\pm 0.03}$	$\mathbf{0.232}_{\pm 0.11}$	$\mathbf{24.49}_{\pm 2.31}$	$0.565_{\pm 0.05}$	$0.725_{\pm 0.06}$	$\mathbf{60.57}_{\pm 0.99}$	$\mathbf{0.14}_{\pm 0.11}$	$\mathbf{0.17}_{\pm 0.08}$

Fig. 3 compares task-wise accuracy changes on CIFAR-10/CIFAR-10-C. The lower-left and upper-right matrices report the compared methods and CDD-EGMMs, respectively, with darker colors indicating higher accuracy. CDD-EGMMs exhibits smaller color variation and maintains strong performance across tasks, demonstrating better knowledge retention and reduced catastrophic forgetting. Results on the remaining datasets are provided in Appendix B.5.



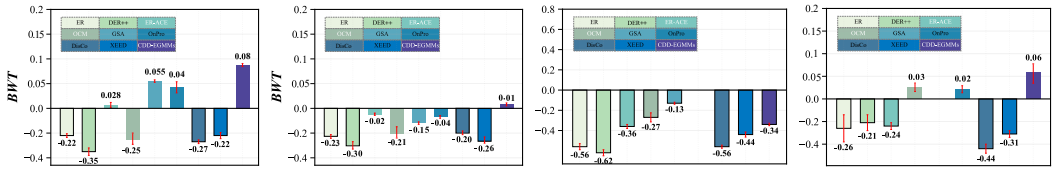
(a) **ER** (b) **DER++** (c) **ER-ACE** (d) **OCM**

(e) **GSA** (f) **OnPro** (g) **DisCo** (h) **XEED**

Figure 3: Task-wise Accuracy Matrices across Continual Learning Methods on CIFAR-10

Fig. 4 compares the backward transfer of different methods. Most baselines exhibit negative BWT, indicating forgetting of earlier tasks. CDD-EGMMs achieves positive BWT on CIFAR-10 and CORe50, indicating beneficial transfer to previously learned tasks. On DomainNet, its BWT of -0.34 reflects less degradation than several baselines, although GSA and OCM achieve higher BWT.

Ablation study on proposed components of CDD-EGMMs. Table 3 examines the contribution of each component. Introducing CDD-EGMMs substantially improves accuracy and alleviates forgetting, and further adding GRC yields the best overall performance, confirming that the compact distributional memory and the representation constraint are complementary.



(a) **CIFAR-10** (b) **CIFAR-100** (c) **DomainNet** (d) **CORe50**

Figure 4: Comparison of BWT across different methods

4.3 Ablation Study and Parameter Analysis

Table 3: Ablation study of components in the proposed method on ResNet18.

Baseline	CDD-EGMMs	GRC	CIFAR-10			CIFAR-100			DomainNet			CORe50		
			$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$
✓			43.81	0.702	0.797	25.14	0.475	0.369	15.34	0.893	0.910	41.21	0.342	0.474
✓	✓		66.21	0.364	0.212	33.77	0.260	0.296	**25.45**	0.661	**0.665**	56.71	0.215	**0.114**
✓	✓	✓	**70.17**	**0.116**	**0.109**	**37.45**	**0.230**	**0.232**	24.49	**0.565**	0.725	**60.57**	**0.140**	0.170

Select the hyperparameter K for CDD-EGMMs. Table 4 studies the effect of the number of Gaussian components K. Too few components are insufficient to capture multimodal class distributions, while too many introduce redundant components and unstable estimation. A moderate K balances distributional expressiveness and model stability, so we adopt $K = 6$ throughout. Additional ablations in Appendix B.6 show that CDD-EGMMs > Entangled GMM Replay > Prototype Replay, confirming that explicit class-domain decoupling brings gains beyond generic distributional replay.

Analysis of the domain basis. We further examine how the rank r, the initialization strategy, and the update strategy of the shared domain basis A affect performance, with full results reported in Appendix B.7. Accuracy improves consistently as r grows from 16 to 64 and saturates at $r = 128$, indicating that a moderate rank suffices to span domain variations. Among the initialization strategies, coordinate initialization performs the best, followed by PCA and random initialization, while zero initialization degrades accuracy the most. Moreover, keeping A fixed during online learning achieves the lowest forgetting on nearly all benchmarks, whereas updating A with Oja's rule or gradients brings only negligible accuracy gains at the cost of clearly aggravated forgetting. These results justify the default design of CDD-EGMMs, i.e., a rank-64, coordinate-initialized, and fixed domain basis.

Efficiency analysis. We further evaluate the GPU memory usage and runtime of all methods, as reported in Appendix B.8. CDD-EGMMs requires the least memory on three of the four benchmarks and achieves the shortest runtime on CIFAR-100, while remaining substantially faster than complex baselines such as OCM, GSA, and XEED, making it well suited to resource-constrained online learning.

Table 4: Analysis of the parameter K.

	CIFAR-10			CIFAR-100			DomainNet			CORe50		
	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$
$K = 2$	68.01	0.176	0.191	24.31	0.75	0.62	23.95	0.613	0.771	**60.58**	0.149	0.177
$K = 4$	69.19	0.144	0.123	35.76	0.40	0.38	24.47	0.590	0.762	60.32	0.142	0.179
$K = 6$	**70.17**	0.116	**0.109**	**37.45**	0.23	**0.23**	**24.49**	0.565	**0.725**	60.57	**0.140**	**0.170**
$K = 8$	68.93	**0.111**	0.110	36.48	**0.17**	0.26	24.31	**0.559**	0.779	60.11	0.148	0.178

5 Conclusion

This study proposes the CDC-OIL framework to simulate scenarios where classes and domains evolve simultaneously in data streams with ambiguous task boundaries. Based on this setting, we design the CDD-EGMMs, which prevents catastrophic forgetting by separating the class space from the domain space and introduces a dynamic adaptation and joint generation mechanism. To prevent decision boundary degradation during continual learning, we design a Gaussian Representation Constraint. Experimental results show that the CDD-EGMMs method performs well on multiple datasets, but still has limitations in streaming environments with scarce labels. We hope this work provides new insights for continual learning research.

REFERENCES

Afra Feyza Akyürek, Ekin Akyürek, Derry Tanti Wijaya, and Jacob Andreas. Subspace regularizers for few-shot class incremental learning. *arXiv preprint arXiv:2110.07059*, 2021.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. volume 33, pp. 15920–15930, 2020.

Lucas Caccia, Eugene Belilovsky, Massimo Caccia, and Joelle Pineau. Online learned continual compression with adaptive quantization modules. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1240–1250, 2020.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021.

Wei Chen and Yi Zhou. Make domain shift a catastrophic forgetting alleviator in class-incremental learning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 15866–15874, 2025.

Heayoun Choi, Hyundong Jin, and Eunwoo Kim. XIL: Cross-expanding incremental learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Botos Csaba, Wenxuan Zhang, Matthias Müller, Ser-Nam Lim, Mohamed Elhoseiny, Philip Torr, and Adel Bibi. Label delay in online continual learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 119976–120012, 2024.

Aveen Dayal, S Shrusti, Linga Reddy Cenkeramaddi, C Krishna Mohan, and Abhinav Kumar. Leveraging mixture alignment for multi-source domain adaptation. *IEEE Transactions on Image Processing*, 34:885–898, 2025.

Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 8109–8126, 2022.

Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11878–11887, 2023.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Michael Hubbertz, Pascal Colling, Qi Han, and Tobias Meisen. Inferring driving maps by deep learning-based trail map extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2425–2434, 2025.

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. *National Academy of Sciences*, 114(13):3521–3526, 2017.

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

Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R. Venkatesh Babu. Class-incremental domain adaptation. In *Proceedings of the Computer Vision (ECCV)*, pp. 53–69, 2020.

Christiaan Lamers, René Vidal, Nabil Belbachir, Niki van Stein, Thomas Bäck, and Paris Giampouras. Clustering-based domain-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3376–3384, 2023.

Lan Li, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Addressing imbalanced domain-incremental learning through dual-balance collaborative experts. *arXiv preprint arXiv:2507.07100*, 2025.

Guoliang Lin, Hanlu Chu, and Hanjiang Lai. Towards better plasticity-stability trade-off in incremental learning: A simple linear connector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 89–98, 2022.

Sihao Liu, Yibo Yang, Xiaojie Li, David A. Clifton, and Bernard Ghanem. Enhancing online continual learning with plug-and-play state space model and class-conditional mixture of discretization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20502–20511, 2025.

Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. pp. 17–26, 2017.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. Icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. volume 32, 2019.

Minhyuk Seo, Hyunseo Koh, and Jonghyun Choi. Budgeted online continual learning by adaptive layer freezing and frequency-based sampling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Haizhou Shi and Hao Wang. A unified approach to domain incremental learning with memory: Theory and algorithm. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 36:15027–15059, 2023.

Christian Simon, Masoud Faraki, Yi-Hsuan Tsai, Xiang Yu, Samuel Schulter, Yumin Suh, Mehrtash Harandi, and Manmohan Chandraker. On generalizing beyond domains in cross-domain continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9255–9264, 2022a.

Christian Simon, Masoud Faraki, Yi-Hsuan Tsai, Xiang Yu, Samuel Schulter, Yumin Suh, Mehrtash Harandi, and Manmohan Chandraker. On generalizing beyond domains in cross-domain continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9265–9274, 2022b.

Filip Szatkowski, Yaoyue Zheng, Fei Yang, Tomasz Trzcinski, Bartlomiej Twardowski, and Joost van de Weijer. Improving continual learning performance and efficiency with auxiliary classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Marco Toldo, Umberto Michieli, and Pietro Zanuttigh. Learning with style: Continual semantic segmentation across tasks and domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(11):7434–7450, 2024.

Keyao Wang, Guosheng Zhang, Haixiao Yue, Ajian Liu, Gang Zhang, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Multi-domain incremental learning for face presentation attack detection. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 5499–5507, 2024a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024b.

Maorong Wang, Nicolas Michel, Ling Xiao, and Toshihiko Yamasaki. Improving plasticity in online continual learning via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23460–23469, 2024c.

Zhen Wang, Liu Liu, Yiqun Duan, and Dacheng Tao. Continual learning through retrieval and imagination. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 8594–8602, 2022a.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149, 2022b.

Jan Wasilewski, Michał Woźniak, and Bartosz Krawczyk. Combining active learning and self-labeling for deep learning from data streams. In *Proceedings of the 2024 IEEE International Conference on Data Mining (ICDM)*, pp. 842–849, 2024.

Yujie Wei, Jiaxin Ye, Zhizhong Huang, Junping Zhang, and Hongming Shan. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18764–18774, 2023.

Jiangwei Xie, Shipeng Yan, and Xuming He. General incremental learning with domain-aware categorical representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14351–14360, 2022.

Yicheng Xu, Yuxin Chen, Jiahao Nie, Yusong Wang, Huiping Zhuang, and Manabu Okumura. Advancing cross-domain discriminability in continual learning of vision-language models. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 37:51552–51576, 2024.

Xi Zhang, Yanyi Li, Yisi Luo, Qi Xie, and Deyu Meng. Online functional tensor decomposition via continual learning for streaming data completion. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2026.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. *arXiv preprint arXiv:2205.13218*, 2022.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, Lijun Zhang, and De-Chuan Zhan. Dual consolidation for pre-trained model-based domain-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20547–20557, 2025.

Guilin Zhu, Runmin Wang, Yuanjie Shao, Wei dong Yang, Nong Sang, and Changxin Gao. Continual gaussian mixture distribution modeling for class incremental semantic segmentation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

A METHOD

Algorithm 1 summarizes mini-batch online learning. For memory updates, samples are processed in stream order using detached features and the current class/domain sets. For GRC in Eq. 15, $\mathcal{D}_{t+1}^{new} = \{(x_i, y_i, d_i) \in \mathcal{D}_{t+1} : y_i \notin \mathcal{C}_t\}$, where $\mathcal{C}_t$ is fixed before the mini-batch arrives. Thus, all samples of its newly observed classes are included. We set $\mathcal{L}_{GRC} = 0$ when this sample set or the previously observed class set is empty. Gaussian statistics and domain states are held fixed during backpropagation. The sequential new-class/new-domain branch uses Eq. 12 with $\mathcal{B}_d = \{i\}$. Replay domains are drawn from a sampling distribution q_t over the observed domains.

A.1 DERIVATION OF EQ. 5

In Eq. 5, the Mahalanobis norm is defined as:

$$\left\| z_t - (\mu_{y,k}^{(t)} + Au_d^{(t)}) \right\|_{\Sigma_{y,k}^{-1}}^2 = \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t)} \right)^\top \Sigma_{y,k}^{-1} \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t)} \right) \tag{17}$$

To simplify the notation, we set:

$$e_{t,k} = z_t - \mu_{y,k}^{(t)} - Au_d^{(t)} \tag{18}$$

and Eq. 5 can be expressed as:

$$\xi(\boldsymbol{u}_d^{(t)}) = \sum_{k=1}^{K} r_{t,k} e_{t,k}^\top \Sigma_{y,k}^{-1} e_{t,k} + \lambda_u u_d^{(t)\top} u_d^{(t)}. \tag{19}$$

Since $e_{t,k}$ is an affine function of $u_d^{(t)}$, we directly compute the gradient of each term.

(1) By taking the gradient of the quadratic form of the first item, we can obtain:

$$\frac{\partial}{\partial u_d^{(t)}} \left(e_{t,k}^\top \Sigma_{y,k}^{-1} e_{t,k} \right) = -2A^\top \Sigma_{y,k}^{-1} \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t)} \right) \tag{20}$$

Therefore, after adding the responsibility degree $r_{t,k}$, we have:

$$\frac{\partial}{\partial u_d^{(t)}} \sum_{k=1}^{K} r_{t,k} e_{t,k}^\top \Sigma_{y,k}^{-1} e_{t,k} = -2 \sum_{k=1}^{K} r_{t,k} A^\top \Sigma_{y,k}^{-1} \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t)} \right) \tag{21}$$

(2) Similarly, we take the derivative of the regularization term:

$$\lambda_u \|u_d^{(t)}\|_2^2 = \lambda_u u_d^{(t)\top} u_d^{(t)}, \quad \frac{\partial}{\partial u_d^{(t)}} \lambda_u u_d^{(t)\top} u_d^{(t)} = 2\lambda_u u_d^{(t)} \tag{22}$$

In conclusion, the overall gradient is:

$$\nabla_{u_d^{(t)}} \xi(\boldsymbol{u}_d^{(t)}) = -2 \sum_{k=1}^{K} r_{t,k} A^\top \Sigma_{y,k}^{-1} \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t)} \right) + 2\lambda_u u_d^{(t)} \tag{23}$$

We update the domain offset term using a one-step gradient descent method:

$$u_d^{(t)} = u_d^{(t-1)} - \eta_t \nabla_{u_d^{(t)}} \xi(\boldsymbol{u}_d^{(t-1)}) \tag{24}$$

Substituting Eq. 23 gives:

$$\begin{aligned} u_d^{(t)} &= u_d^{(t-1)} + 2\eta_t \left[\sum_{k=1}^{K} r_{t,k} A^\top \Sigma_{y,k}^{-1} \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t-1)} \right) - \lambda_u u_d^{(t-1)} \right] \\ &= u_d^{(t-1)} + \eta_t \left[\sum_{k=1}^{K} r_{t,k} A^\top \Sigma_{y,k}^{-1} \left(z_t - \mu_{y,k}^{(t)} - Au_d^{(t-1)} \right) - \lambda_u u_d^{(t-1)} \right] \end{aligned} \tag{25}$$

which is exactly Eq. 6. This completes the proof.

Algorithm 1: CDD-EGMMs for Class-Domain Coupled Online Incremental Learning

Input: Mini-batch stream $\{\mathcal{B}_t\}_{t=1}^{T}$ with labels revealed after prediction; feature extractor f_θ; classifier h_ω; fixed basis A; K, ρ, η_t, λ_u, λ, σ^2, margin m, α; replay count M per class; domain sampler q_t; optional merge threshold τ_{merge}

Output: f_θ, h_ω, Gaussian memory Φ, and domain states $\{u_d\}$

Initialize $\mathcal{C} = \emptyset$, $\mathcal{P} = \emptyset$, $\Phi = \emptyset$, and $\{u_d\} = \emptyset$;

foreach *incoming mini-batch* $\mathcal{B}_t$ **do**

  Save the previously observed class set $\mathcal{C}^- \leftarrow \mathcal{C}$;

  Compute $z_i = f_\theta(x_i)$; predict with h_ω before revealing labels (abstain if $\mathcal{C}^- = \emptyset$);

  Reveal $\{y_i, d_i\}$; set $I_{\text{new}} = \{i \in \mathcal{B}_t : y_i \notin \mathcal{C}^-\}$;

  foreach $i \in \mathcal{B}_t$ *in stream order* (memory updates use detached z_i) **do**

    Set $b_c = \mathbb{I}[y_i \notin \mathcal{C}]$ and $b_d = \mathbb{I}[d_i \notin \mathcal{P}]$;

    if $b_c = 0$ *and* $b_d = 0$ **then**

      `// Old class / old domain`

      Compute responsibilities using $z_i - Au_{d_i}$ by Eq. 4;

      Update u_{d_i} by Eq. 6;

    else if $b_c = 0$ *and* $b_d = 1$ **then**

      `// Old class / new domain`

      Set $u_{d_i} = 0$ and compute initial responsibilities by Eq. 4;

      Estimate u_{d_i} by Eq. 9;

    else if $b_c = 1$ *and* $b_d = 0$ **then**

      `// New class / old domain`

      Initialize the class GMM from $z_i - Au_{d_i}$ and its statistics by Eqs. 10–11;

    else

      `// New class / new domain`

      Set $u_{d_i} = 0$ and initialize provisional class components from z_i by Eq. 10;

      Estimate u_{d_i} by Eq. 12 with $\mathcal{B}_{d_i} = \{i\}$;

      Shift the provisional means by $-Au_{d_i}$ and initialize statistics by Eq. 11;

    if $b_c = 0$ **then**

      Recompute responsibilities using $z_i - Au_{d_i}$; update statistics and GMM parameters by Eqs. 7–8;

    $\mathcal{C} \leftarrow \mathcal{C} \cup \{y_i\}$; $\mathcal{P} \leftarrow \mathcal{P} \cup \{d_i\}$;

  Expand the output space of h_ω to include all classes in $\mathcal{C}$;

  `// Generative joint replay`

  For each $c \in \mathcal{C}$, draw M features: $d \sim q_t$, $k \sim \pi_c$, $\tilde{z} \sim \mathcal{N}(\mu_{c,k}, \Sigma_{c,k})$, $z^{\text{rep}} = \tilde{z} + Au_d$;

  Form $\mathcal{R}_t = \{(z^{\text{rep}}, c)\}$ and detach all replay features;

  $$\mathcal{L}_{\text{cls}} = -\frac{\sum_{i \in \mathcal{B}_t} \log h_\omega(z_i)_{y_i} + \sum_{(z,c) \in \mathcal{R}_t} \log h_\omega(z)_c}{|\mathcal{B}_t| + |\mathcal{R}_t|};$$

  `// GRC on all newly observed classes in the mini-batch`

  Set $\mathcal{L}_{\text{GRC}} = 0$;

  if $I_{\text{new}} \neq \emptyset$ *and* $\mathcal{C}^- \neq \emptyset$ **then**

    Compute Eq. 15 over I_{new}, with $\tilde{z}_i = z_i - Au_{d_i}$ and negative classes in $\mathcal{C}^-$;

  Hold A, Φ, and $\{u_d\}$ fixed; update (θ, ω) with $\mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{GRC}}$;

  `// Real features and GRC update` f_θ`; real and replay features update` h_ω

  If enabled, merge similar domain prototypes as in Section 3.3, preserving the domain-label-to-state mapping;

A.2 Theoretical Analysis of Class–Domain Decoupling

In this section, we provide a theoretical explanation of why class–domain decoupling can reduce the interference between class semantics and domain-specific variations. The key intuition is that, in a mixed class-domain feature space, the estimated class representation is inevitably biased by domain shifts. In contrast, by explicitly decomposing each feature into a domain-invariant semantic component and a domain-specific shift component, the proposed CDD-EGMMs can obtain a less biased estimation of class semantics, thereby reducing the interference caused by continuously emerging domains.

Feature generation model. For a sample belonging to class c and domain d, we assume that its latent feature $z \in \mathbb{R}^m$ is generated by

$$z = \mu_{c,k} + Au_d + \epsilon, \tag{26}$$

where $\mu_{c,k}$ denotes the k-th domain-invariant semantic component of class c, $A \in \mathbb{R}^{m\times r}$ is a shared domain subspace basis, $u_d \in \mathbb{R}^r$ is the domain-specific shift coefficient of domain d, and ϵ is a zero-mean noise term satisfying $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}\|\epsilon\|_2^2 < \infty$. This assumption is consistent with our class–domain decoupled Gaussian mixture model:

$$p(z|y = c, d; \phi_c) = \sum_{k=1}^{K} \pi_{c,k}\mathcal{N}(z; \mu_{c,k} + Au_d, \Sigma_{c,k}). \tag{27}$$

For simplicity, we first analyze the single-component case and then extend the conclusion to the mixture case.

Proposition 1. Domain shifts introduce biased class estimation in the entangled feature space. Suppose that the feature of class c under domain d follows Eq. (26). If the class prototype is estimated directly in the original feature space without removing the domain shift, i.e.,

$$\bar{\mu}_c = \mathbb{E}[z|y = c], \tag{28}$$

then the estimated class prototype is biased by the domain distribution:

$$\bar{\mu}_c = \mu_c + A\mathbb{E}[u_d|y = c]. \tag{29}$$

Consequently, the estimation bias is

$$\bar{\mu}_c - \mu_c = A\mathbb{E}[u_d|y = c], \tag{30}$$

and its magnitude is bounded by

$$\|\bar{\mu}_c - \mu_c\|_2 \le \|A\|_2 \cdot \|\mathbb{E}[u_d|y = c]\|_2. \tag{31}$$

Proof. According to Eq. (26), for any sample from class c, we have

$$z = \mu_c + Au_d + \epsilon. \tag{32}$$

Taking conditional expectation with respect to $y = c$ yields

$$\mathbb{E}[z|y = c] = \mathbb{E}[\mu_c + Au_d + \epsilon|y = c]. \tag{33}$$

Since μ_c is the domain-invariant semantic representation of class c and $\mathbb{E}[\epsilon] = 0$, we obtain

$$\mathbb{E}[z|y = c] = \mu_c + A\mathbb{E}[u_d|y = c]. \tag{34}$$

Therefore,

$$\bar{\mu}_c - \mu_c = A\mathbb{E}[u_d|y = c]. \tag{35}$$

By the submultiplicativity of the spectral norm,

$$\|\bar{\mu}_c - \mu_c\|_2 = \|A\mathbb{E}[u_d|y = c]\|_2 \le \|A\|_2 \cdot \|\mathbb{E}[u_d|y = c]\|_2. \tag{36}$$

This completes the proof. □

Discussion. Proposition 1 shows that when class and domain factors are modeled in a shared entangled space, the learned class prototype does not purely represent class semantics. Instead, it contains an additional bias term $A\mathbb{E}[u_d|y = c]$ induced by domain shifts. As new domains continuously arrive, $\mathbb{E}[u_d|y = c]$ changes over time, which causes the estimated class center to drift. This drift directly interferes with the stability of old class representations and may lead to catastrophic forgetting.

Proposition 2. Class–domain decoupling provides an unbiased semantic estimator. Given the feature generation model in Eq. (26), define the domain-corrected representation as

$$\tilde{z} = z - Au_d. \tag{37}$$

Then the expectation of the corrected representation is an unbiased estimator of the class semantics:

$$\mathbb{E}[\tilde{z}|y = c] = \mu_c. \tag{38}$$

Proof. Substituting Eq. (26) into Eq. (37), we obtain

$$\tilde{z} = z - Au_d = \mu_c + Au_d + \epsilon - Au_d = \mu_c + \epsilon. \tag{39}$$

Taking conditional expectation with respect to $y = c$, we have

$$\mathbb{E}[\tilde{z}|y = c] = \mathbb{E}[\mu_c + \epsilon|y = c] = \mu_c + \mathbb{E}[\epsilon]. \tag{40}$$

Since $\mathbb{E}[\epsilon] = 0$, it follows that

$$\mathbb{E}[\tilde{z}|y = c] = \mu_c. \tag{41}$$

This completes the proof. □

Implication. Proposition 2 indicates that after removing the domain-specific shift Au_d, the corrected representation $\tilde{z}$ preserves only the class semantic component and the random noise. Therefore, the class representation estimated in the corrected space is not affected by the mean shift of domains. This explains why the proposed decoupled modeling strategy can reduce the interference between domain adaptation and class knowledge retention.

Proposition 3. Decoupling reduces the semantic drift caused by new domains. Let $\bar{\mu}_c^{(t)}$ denote the class representation estimated in the entangled feature space after observing domains up to time t, and let $\tilde{\mu}_c^{(t)}$ denote the class representation estimated in the decoupled semantic space. Suppose the observed domain distribution of class c changes from time t to $t + 1$. Then the semantic drift in the entangled space is

$$\Delta_{\text{ent}}^{(t)} = \|\bar{\mu}_c^{(t+1)} - \bar{\mu}_c^{(t)}\|_2 = \|A\left(\mathbb{E}_{t+1}[u_d|y = c] - \mathbb{E}_t[u_d|y = c]\right)\|_2, \tag{42}$$

while the semantic drift in the decoupled space satisfies

$$\Delta_{\text{dec}}^{(t)} = \|\tilde{\mu}_c^{(t+1)} - \tilde{\mu}_c^{(t)}\|_2 = 0 \tag{43}$$

in expectation. Therefore,

$$\mathbb{E}[\Delta_{\text{dec}}^{(t)}] \leq \mathbb{E}[\Delta_{\text{ent}}^{(t)}]. \tag{44}$$

Proof. From Proposition 1, the class representation estimated in the entangled space at time t can be written as

$$\bar{\mu}_c^{(t)} = \mu_c + A\mathbb{E}_t[u_d|y = c], \tag{45}$$

where $\mathbb{E}_t[u_d|y = c]$ denotes the expectation over the domain distribution observed up to time t. Similarly, at time $t + 1$,

$$\bar{\mu}_c^{(t+1)} = \mu_c + A\mathbb{E}_{t+1}[u_d|y = c]. \tag{46}$$

Thus,

$$\begin{aligned}\Delta_{\text{ent}}^{(t)} &= \|\bar{\mu}_c^{(t+1)} - \bar{\mu}_c^{(t)}\|_2 \\ &= \|A\left(\mathbb{E}_{t+1}[u_d|y = c] - \mathbb{E}_t[u_d|y = c]\right)\|_2.\end{aligned} \tag{47}$$

This shows that any change in the observed domain distribution directly changes the estimated class representation in the entangled space.

In contrast, from Proposition 2, the expected class representation in the decoupled space is

$$\tilde{\mu}_c^{(t)} = \mathbb{E}_t[\tilde{z}|y=c] = \mu_c, \tag{48}$$

and

$$\tilde{\mu}_c^{(t+1)} = \mathbb{E}_{t+1}[\tilde{z}|y=c] = \mu_c. \tag{49}$$

Therefore,

$$\Delta_{\text{dec}}^{(t)} = \|\tilde{\mu}_c^{(t+1)} - \tilde{\mu}_c^{(t)}\|_2 = 0. \tag{50}$$

It follows that

$$\mathbb{E}[\Delta_{\text{dec}}^{(t)}] \leq \mathbb{E}[\Delta_{\text{ent}}^{(t)}]. \tag{51}$$

This completes the proof. □

Extension to Gaussian mixture models. The above analysis can be naturally extended to the mixture case. For class c, the class-conditional distribution is modeled as

$$p(z|y=c,d) = \sum_{k=1}^{K} \pi_{c,k} \mathcal{N}(z; \mu_{c,k} + Au_d, \Sigma_{c,k}). \tag{52}$$

The entangled class center is

$$\bar{\mu}_c = \sum_{k=1}^{K} \pi_{c,k}(\mu_{c,k} + A\mathbb{E}[u_d|y=c]) = \sum_{k=1}^{K} \pi_{c,k}\mu_{c,k} + A\mathbb{E}[u_d|y=c]. \tag{53}$$

The decoupled class center is computed from the corrected feature $\tilde{z} = z - Au_d$:

$$\tilde{\mu}_c = \mathbb{E}[\tilde{z}|y=c] = \sum_{k=1}^{K} \pi_{c,k}\mu_{c,k}. \tag{54}$$

Comparing Eq. (53) and Eq. (54), we can see that the entangled estimation contains the additional domain-dependent term $A\mathbb{E}[u_d|y=c]$, whereas the decoupled estimation removes this term. Therefore, the proposed CDD-EGMMs can reduce the interference caused by domain shifts not only in the single-prototype case but also in multimodal class distributions.

Theorem 1. Class–domain decoupling reduces interference between old and new domains. Assume that for each class c, the latent feature follows the decoupled generation model in Eq. (26). Let $\bar{\mu}_c^{(t)}$ be the class center estimated in the original entangled feature space, and let $\tilde{\mu}_c^{(t)}$ be the class center estimated in the domain-corrected space. When a new domain d_{new} arrives, the expected perturbation imposed on the class center in the entangled space is bounded by

$$\|\bar{\mu}_c^{(t+1)} - \bar{\mu}_c^{(t)}\|_2 \leq \|A\|_2 \cdot \|\mathbb{E}_{t+1}[u_d|y=c] - \mathbb{E}_t[u_d|y=c]\|_2\,, \tag{55}$$

whereas the expected perturbation in the decoupled semantic space is zero:

$$\mathbb{E}\left[\|\tilde{\mu}_c^{(t+1)} - \tilde{\mu}_c^{(t)}\|_2\right] = 0. \tag{56}$$

Therefore, class–domain decoupling strictly reduces the expected semantic interference introduced by new domains whenever

$$A\left(\mathbb{E}_{t+1}[u_d|y=c] - \mathbb{E}_t[u_d|y=c]\right) \neq 0. \tag{57}$$

Proof. According to Proposition 1, the entangled class center at time t is

$$\bar{\mu}_c^{(t)} = \sum_{k=1}^{K} \pi_{c,k}\mu_{c,k} + A\mathbb{E}_t[u_d|y=c]. \tag{58}$$

After the new domain arrives, the observed domain distribution changes. Therefore, the entangled class center becomes

$$\bar{\mu}_c^{(t+1)} = \sum_{k=1}^{K} \pi_{c,k}\mu_{c,k} + A\mathbb{E}_{t+1}[u_d|y=c]. \tag{59}$$

Subtracting the two equations gives

$$\bar{\mu}_c^{(t+1)} - \bar{\mu}_c^{(t)} = A\left(\mathbb{E}_{t+1}[u_d|y=c] - \mathbb{E}_t[u_d|y=c]\right). \tag{60}$$

Taking the Euclidean norm and applying the spectral norm inequality, we obtain

$$\begin{aligned}\|\bar{\mu}_c^{(t+1)} - \bar{\mu}_c^{(t)}\|_2 &= \|A\left(\mathbb{E}_{t+1}[u_d|y=c] - \mathbb{E}_t[u_d|y=c]\right)\|_2 \\ &\le \|A\|_2 \cdot \|\mathbb{E}_{t+1}[u_d|y=c] - \mathbb{E}_t[u_d|y=c]\|_2 .\end{aligned} \tag{61}$$

This proves Eq. (55).

For the decoupled representation, we have

$$\tilde{z} = z - Au_d. \tag{62}$$

Substituting Eq. (26) into the above equation yields

$$\tilde{z} = \mu_{c,k} + \epsilon. \tag{63}$$

Thus, the expected class center in the decoupled space is

$$\tilde{\mu}_c^{(t)} = \sum_{k=1}^{K} \pi_{c,k}\mu_{c,k}, \tag{64}$$

which is independent of the domain distribution. Similarly,

$$\tilde{\mu}_c^{(t+1)} = \sum_{k=1}^{K} \pi_{c,k}\mu_{c,k}. \tag{65}$$

Therefore,

$$\mathbb{E}\left[\|\tilde{\mu}_c^{(t+1)} - \tilde{\mu}_c^{(t)}\|_2\right] = 0. \tag{66}$$

When

$$A\left(\mathbb{E}_{t+1}[u_d|y=c] - \mathbb{E}_t[u_d|y=c]\right) \neq 0, \tag{67}$$

the perturbation in the entangled space is strictly positive, while the expected perturbation in the decoupled space is zero. Hence, class–domain decoupling strictly reduces the expected semantic interference caused by new domains. This completes the proof. □

Connection to catastrophic forgetting. The above theorem explains why the proposed decoupled modeling strategy is beneficial for continual learning under simultaneous class expansion and domain shift. In the entangled space, new domains change the estimated class centers through the bias term $A\mathbb{E}[u_d|y=c]$. Such changes distort previously learned class semantics and may move old class distributions toward incorrect decision regions. In contrast, CDD-EGMMs assigns domain-specific variations to the domain shift term Au_d while preserving the class semantics in $\mu_{c,k}$. Therefore, the update caused by a new domain mainly affects the domain state u_d rather than overwriting the semantic components $\mu_{c,k}$. This reduces the interference between new-domain adaptation and old-class retention, thereby alleviating catastrophic forgetting.

B Experiments

B.1 Fair comparison protocol

Since some compared methods were originally designed for offline or multi-pass continual learning, we adapt them to the proposed CDC-OIL protocol for fair comparison. Specifically, all methods are trained in a strictly single-pass online manner, where each sample is observed only once and the full training set cannot be revisited. Offline retraining, multiple training epochs, and explicit task-boundary information are disabled for all baselines. All methods use the same ResNet-18 backbone, the same data stream order, the same batch size, and the same memory budget when replay buffers are required. Therefore, the comparison focuses on the ability of each method to learn under the same online class-domain evolving setting.

B.2 Experimental Settings

CDC-OIL follows a test-then-train protocol. At each time step, the model first predicts the label of an incoming sample without accessing its ground-truth class and domain labels. After prediction, the class label and domain label are revealed for online updating. The model has no access to explicit task identities or task boundaries, and each sample can be used only once. Unless otherwise specified, raw historical samples are not stored; the model only maintains compact distributional statistics. Evaluation is conducted over all classes and domains observed so far.

In CDC-OIL, each incoming sample may belong to one of four cases: old-class/old-domain, old-class/new-domain, new-class/old-domain, and new-class/new-domain. The first case requires stable adaptation, the second requires domain-space expansion, the third requires semantic-space expansion, and the fourth requires joint initialization of both semantic and domain structures.

CIFAR-10. CIFAR-10 contains 10 classes, which are randomly divided into five two-class tasks to simulate class expansion. For each task, we randomly sample five corruption types from CIFAR-10-C and apply them to the current classes to introduce domain shifts during continual learning. Each corruption type is treated as a distinct domain, and all samples are streamed once under the test-then-train protocol. The model is not given explicit task-boundary information, and evaluation is conducted over all classes and domains observed so far.

DomainNet. DomainNet contains 345 classes and six visual domains. Following prior work, we exclude the real domain and use the remaining five domains to construct domain shifts. To better match CDC-OIL, we revise the DomainNet protocol by introducing both class increments and domain increments over the stream. Specifically, the 345 classes are split into five disjoint class increments, and the five non-real domains are progressively introduced across tasks instead of being fully exposed within each task. Samples within each task are then streamed without explicit task-boundary information during training.

CORe50. We split the 10 object classes into five class-incremental tasks, with each task containing two classes. Within each task, samples are collected from different domains induced by changes in acquisition environment, illumination, background, and viewpoint. Unlike randomly shuffled datasets, CORe50 naturally follows a sequential streaming order, such as illumination changes from weak to strong and camera viewpoints varying from top to bottom. The model is not given explicit task-boundary information, and all samples are processed once under the test-then-train protocol.

B.3 Baseline Methods

This section systematically introduces the baseline methods as follows:

ER (Rolnick et al., 2019), Experience Replay mitigates catastrophic forgetting by storing a small number of historical samples and replaying them during subsequent training.

DER++ (Buzzega et al., 2020) , Dark Experience Replay++ introduces a distillation constraint on historical logits based on experience replay, enabling the model to preserve old knowledge while learning new knowledge.

ER-ACE (Caccia et al., 2021), Experience Replay with Asymmetric Cross-Entropy reduces interference between old and new classes through asymmetric cross-entropy, improving the stability of class-incremental learning.

OCM (Guo et al., 2022), Online Continual Learning further enhances knowledge retention by maximizing the mutual information between feature representations of new and old samples.

GSA (Guo et al., 2023), Gradient Self-Adaptation addresses cross-task class discrimination in online continual learning through a gradient-based adaptive loss that compensates for gradient imbalance caused by data imbalance and continual learning imbalance.

OnePro (Wei et al., 2023), by introducing online prototype equilibrium (OPE) and adaptive prototype feedback (APF), this method effectively mitigates shortcut learning and catastrophic forgetting in online continual learning from the perspectives of representative feature learning and discriminative feature enhancement.

DisCo (Chen & Zhou, 2025), DisCo builds coarse and fine-grained concept prototypes for each class and aligns image features with these prototypes to balance adaptation to new domains and retention of old knowledge.

XEED (Choi et al., 2026), XEED is designed for class-incremental and domain-expansion settings, enhancing adaptation to new classes and domains through cross-domain expansion and generative prototypes.

B.4 Evaluation Metrics

To comprehensively evaluate the performance of CDD-EGMMs across different datasets, we use three core metrics: Average Accuracy A_T, Forgetting Rate F_T, and Relative Forgetting Rate RF. The detailed definitions and calculation methods of these metrics are as follows:

Average accuracy A_T denotes the average accuracy after learning T tasks,

$$A_T = \frac{1}{T}\sum_{t=1}^{T} a_t \tag{68}$$

where a_t denotes the model accuracy on task t.

Forgetting Measure F_T, it measures the difference between the model's best performance during continual learning and its performance after incremental learning; a larger gap indicates more severe forgetting,

$$F_T = \frac{1}{T-1}\sum_{j=1}^{T-1}\left(\max_{i\in\{j,\ldots,T-1\}} a_{i,j} - a_{T,j}\right), \qquad T > 1 \tag{69}$$

where $a_{i,j}$ denotes the test accuracy on task j after the model has learned task i, $\max_i a_{i,j}$ is the best accuracy achieved on task j throughout the entire learning process, $a_{T,j}$ is the final accuracy on task j after learning the last task. The difference between them indicates the degree of forgetting on that task.

Relative Forgetting RF measures how much proportion of performance the model forgets. Specifically, RF for the j-th task after training on task t is defined as:

$$RF = \frac{1}{T}\sum_{j=1}^{T}\max_{i\in\{1,\ldots,t\}}\left(1 - \frac{a_{j,t}}{a_{i,j}}\right), \forall j \le t \tag{70}$$

RF enables quantitative analysis of the relationship between model stability and final performance, helping guide the improvement of continual learning strategies.

Backward Transfer BWT used to evaluate the impact of learning new tasks on the model's performance on previous tasks:

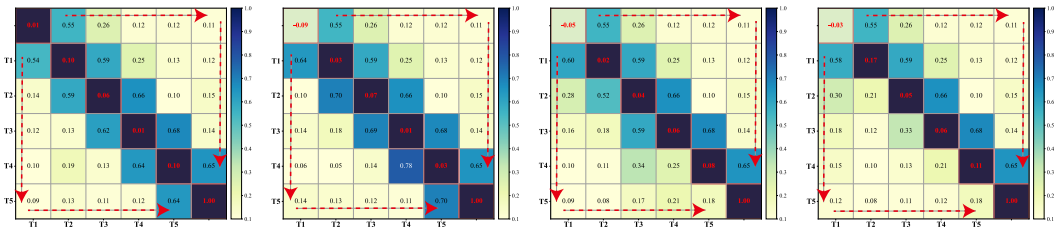
$$BWT = \frac{1}{T-1}\sum_{j=1}^{T-1}\left(a_{T,j} - a_{j,j}\right) \tag{71}$$

where $a_{j,j}$ denotes the accuracy on task j immediately after the model learns task j, while $a_{T,j}$ denotes the final accuracy on task j after the model has learned all tasks. A larger BWT is better; values closer to 0 indicate less forgetting, while more negative values indicate more severe forgetting.

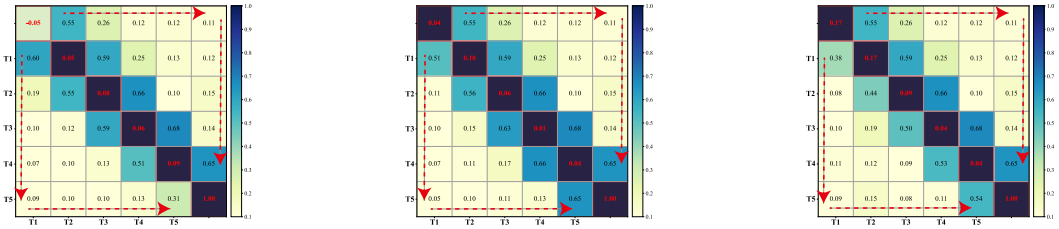
B.5 Additional Experiments Results

Fig. 5 shows the task-wise accuracy matrices of different methods on DomainNet, where darker colors denote higher accuracy. Most methods suffer clear accuracy degradation on previous tasks after learning new ones, indicating severe catastrophic forgetting. This is mainly due to the large label space and strong domain heterogeneity of DomainNet. Although DisCo and XEED achieve relatively better retention, their performance still fluctuates across tasks, highlighting the difficulty of continual learning under class-domain co-evolution.

Fig. 6 shows the task-wise accuracy matrices of different methods on CORe50 with five tasks. Most methods exhibit clear performance drops on previous tasks after learning new ones, revealing



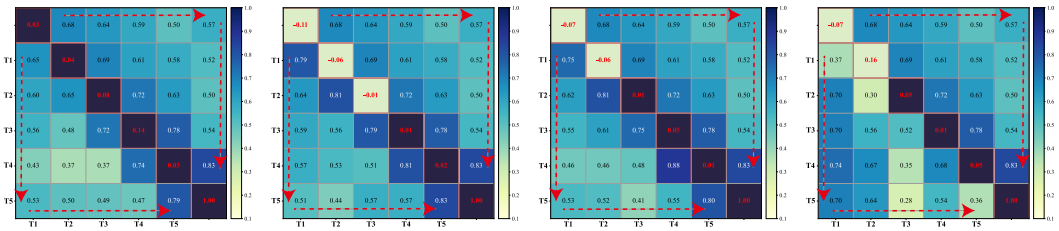
(a) **ER** (b) **DER++** (c) **ER-ACE** (d) **OCM**



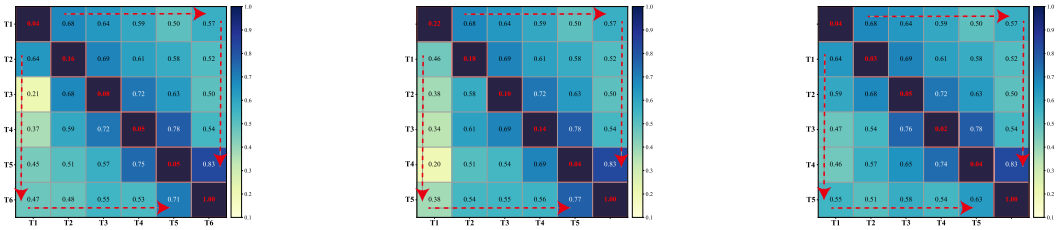
(e) **GSA** (f) **DisCo** (g) **XEED**

Figure 5: Task-wise Accuracy Matrices across Continual Learning Methods on DomainNet

severe catastrophic forgetting. The strong fluctuations across tasks further indicate the difficulty of CORe50, where large-scale class expansion and highly heterogeneous domains substantially increase class-domain ambiguity.



(a) **ER** (b) **DER++** (c) **ER-ACE** (d) **OCM**



(e) **OnePro** (f) **DisCo** (g) **XEED**

Figure 6: Task-wise Accuracy Matrices across Continual Learning Methods on CORe50

B.6 Supplementary Ablation Experiment

Table 5 reports the ablation results of the core components in CDD-EGMMs on three datasets. Compared with the baseline, both Prototype Replay and Entangled GMM Replay improve performance to some extent, showing the benefit of distributional replay. However, CDD-EGMMs w/o GRC achieves stronger results, indicating that class–domain decoupled modeling is more effective than entangled distribution modeling. The full CDD-EGMMs further obtains the best or competitive performance across most metrics, especially achieving the highest A_T and lowest F_T on CIFAR-10

and CORe50. These results demonstrate that decoupled Gaussian modeling, generative replay, and Gaussian Representation Constraint jointly contribute to better accuracy, stability, and forgetting mitigation.

Table 5: Ablation study of the core components in CDD-EGMMs on three datasets.

Method Variant	CIFAR-10/CIFAR-10-C				DomainNet				CORe50			
	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$BWT \uparrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$BWT \uparrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$BWT \uparrow$
Baseline	43.81	0.702	0.797	-0.479	15.34	0.893	0.910	-0.821	41.21	0.342	0.474	-0.13
+ Prototype Replay	60.76	0.431	0.333	-0.109	22.22	0.819	0.893	-0.441	53.86	0.301	0.463	-0.16
+ Entangled GMM Replay	60.88	0.499	0.313	-0.200	22.90	0.748	0.801	-0.36	53.74	0.256	0.411	0.019
+ CDD-EGMMs w/o GRC	66.21	0.364	0.212	**0.09**	**25.45**	0.661	**0.665**	-0.37	56.71	0.215	**0.114**	0.033
+ CDD-EGMMs Full	**70.17**	**0.116**	**0.109**	0.08	24.49	**0.565**	0.725	**-0.34**	**60.57**	**0.14**	0.17	**0.06**

B.7 Domain Basis Analysis

We further investigate how the design choices of the shared low-rank domain basis A (Section 3.2) affect the performance of CDD-EGMMs, including its rank r, its initialization strategy, and its update strategy during online learning. Table 6 reports A_T, F_T, and RF on the four benchmarks, where the configuration adopted by CDD-EGMMs ($r = 64$, coordinate initialization, and a fixed basis, marked with †) is repeated in each group for comparison.

Rank r. A small rank limits the capacity of the basis to span domain shifts: with $r = 16$, A_T drops by up to 3.66 points on CIFAR-10 and 2.65 points on CORe50 compared with $r = 64$. Performance improves consistently from $r = 16$ to $r = 64$ and saturates afterwards, as $r = 128$ brings no further gains and even slightly degrades A_T on three of the four benchmarks. A moderate rank is thus sufficient to capture the dominant directions of domain variation, and we set $r = 64$ throughout.

Initialization. Coordinate initialization performs the best or tied-best on 11 of the 12 columns. PCA initialization, which extracts principal directions from the feature statistics of the initial stream segment, ranks second on most benchmarks, whereas zero initialization is consistently the worst (e.g., 1.71 points lower than coordinate initialization on CIFAR-10 and 2.39 points lower on CORe50). These results indicate that a well-structured initial basis is important for disentangling domain shifts from the very beginning of the stream, while an uninformative basis forces the semantic components to absorb domain variations.

Update strategy. Keeping A fixed after initialization yields the lowest F_T and RF on nearly all benchmarks together with the best A_T on CIFAR-10. Although updating A online with Oja's rule or gradient descent marginally improves A_T on a few benchmarks (by at most 0.06), it consistently aggravates forgetting (e.g., F_T on CIFAR-10 rises from 0.12 to 0.21 with gradient updates). A fixed basis stabilizes the class–domain decomposition, so that the online updates of the semantic components $\boldsymbol{\mu}_{c,k}$, $\boldsymbol{\Sigma}_{c,k}$ and the domain coefficients $\boldsymbol{u}_d$ are not disturbed by a drifting subspace.

Overall, the default configuration of CDD-EGMMs, i.e., a rank-64, coordinate-initialized, and fixed domain basis, achieves the best accuracy–stability trade-off among all evaluated settings.

B.8 Efficiency Analysis

We also evaluate the GPU memory usage and runtime across all methods, as reported in Table 7. CDD-EGMMs uses the least memory on CIFAR-100, DomainNet, and CORe50, requiring only 3.41, 4.18, and 2.12 GB, respectively. On CIFAR-10, it uses 3.67 GB, second only to DER++. It also achieves the shortest runtime on CIFAR-100 (1,348 s) and remains computationally competitive on the other datasets. Although slower than some simple replay methods on CIFAR-10 and DomainNet, it is substantially faster than more complex approaches such as OCM, GSA, and XEED. Overall, CDD-EGMMs combines low memory overhead with competitive efficiency, making it well suited to resource-constrained online learning.

Table 6: Ablation study on the rank r, the initialization strategy, and the update strategy of the shared domain basis A on four benchmarks. Bold values denote the best result per column within each group. † denotes the default configuration of CDD-EGMMs.

Setting		CIFAR-10			CIFAR-100			DomainNet			CORe50		
		$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$	$A_T \uparrow$	$F_T \downarrow$	$RF \downarrow$
Rank	$r = 16$	66.51	0.23	0.20	35.99	0.31	0.33	23.51	0.68	0.89	57.92	0.18	0.23
	$r = 32$	68.46	0.17	0.15	36.23	0.33	0.33	23.87	0.68	0.78	58.37	0.15	0.18
	$r = 64^{\dagger}$	**70.17**	**0.12**	**0.11**	37.45	0.23	0.23	**24.49**	**0.56**	0.72	**60.57**	0.14	**0.17**
	$r = 128$	70.11	0.13	0.12	**37.51**	**0.22**	**0.20**	24.33	**0.56**	0.73	60.52	**0.13**	0.18
Initialization	PCA	69.72	0.20	0.19	36.74	0.24	0.24	23.72	0.58	**0.72**	59.04	0.15	0.18
	Random	69.23	0.17	**0.10**	36.65	0.29	**0.23**	23.68	0.65	0.81	58.18	0.16	**0.17**
	Coordinate†	**70.17**	**0.12**	0.11	**37.45**	**0.23**	**0.23**	**24.49**	**0.56**	**0.72**	**60.57**	**0.14**	**0.17**
	Zero	68.46	0.20	**0.10**	36.60	0.25	0.25	23.70	0.66	0.83	58.18	0.19	0.21
Update	Fixed†	**70.17**	**0.12**	**0.11**	37.45	**0.23**	**0.23**	24.49	**0.56**	**0.72**	60.57	**0.14**	**0.17**
	Oja's rule	70.15	0.16	0.14	**37.51**	0.24	0.25	24.39	0.64	0.79	**60.61**	0.16	0.19
	Gradient	70.06	0.21	0.18	37.50	0.24	0.24	**24.55**	0.62	**0.72**	60.45	0.16	0.18

Table 7: GPU memory usage (GB) and runtime (s) of different methods on four benchmarks. Best results are highlighted in bold. The symbol "-" denotes an out-of-memory error.

Method	CIFAR-10		CIFAR-100		DomainNet		CORe50	
	Memory(GB)	Time(s)	Memory(GB)	Time(s)	Memory(GB)	Time(s)	Memory(GB)	Time(s)
ER	4.59	1172	7.18	1391	8.73	10864	2.99	1533
DER++	**2.89**	2992	5.09	2735	10.48	14143	4.38	**1113**
ER-ACE	9.71	10875	6.35	1511	14.31	15076	6.47	1579
OCM	4.31	24061	6.94	7813	23.70	41991	4.75	2786
GSA	6.43	7317	8.55	14630	8.74	56724	-	-
OnPro	5.03	2896	16.98	13437	-	-	9.46	3712
DisCo	6.73	2923	7.97	1672	7.06	**5490**	4.92	2333
XEED	6.66	6316	6.57	6779	10.32	27714	4.85	2702
CDD-EGMMs	3.67	1755	**3.41**	**1348**	**4.18**	11781	**2.12**	1692