Class-Domain Incremental Learning with Extensible Multi-Center Modeling

Xuetong Yang, Yuxiang Yan, Zhiyuan Zhou, Xin Gao, Guanghao Li, Jian Pu
Fudan University, Shanghai, China
xtyang24@m.fudan.edu.cn jianpu@fudan.edu.cn

Abstract

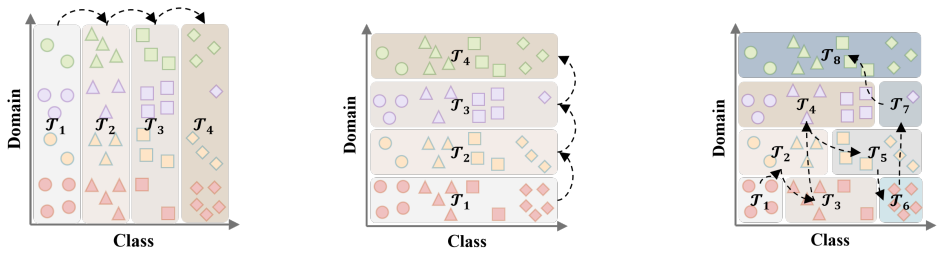
Incremental learning aims to continuously learn from dynamic data streams without suffering from catastrophic forgetting. However, existing methods may fail in challenging scenarios where class and domain spaces expand simultaneously, struggling to learn continually and stably from fragmented data streams. To address this challenge, we propose EMCift (**E**xtensible **M**ulti-**C**enter Modeling with Dual-Path Dr**ift** Harmonization), a novel model that treats heterogeneous data across domains and semantic spaces as co-existing multi-view observations, enabling the construction of global knowledge from local observations. Our approach organizes extensible multi-center prototypes as a tree hierarchy, allowing the global class representation across different domains to grow organically from fragmented local data. To sustain this dynamic topology, we introduce a dual-path drift harmonization strategy that balances structural stability with adaptive plasticity. Specifically, one path captures essential updates to enable prototype evolution, while the other path leverages a conditional adversarial mechanism to robustly initialize domains, effectively preventing catastrophic forgetting caused by class confusion. Extensive experiments on multiple datasets demonstrate the state-of-the-art performance of our proposed model in the class-domain incremental learning scenario. Codes are released at https://github.com/YanngXT/EMCift.

1 Introduction

Incremental learning [1, 2] enables neural networks to adapt continuously to dynamic environments without catastrophic forgetting [3]. In real-world data, streams usually present a complex scenario where new classes and domains emerge concurrently as fragmented observations. Consequently, it becomes critical to enable the model to learn continually and stably from fragmented and unstable data streams, allowing it to accumulate knowledge from partial views without catastrophic forgetting.

Conventional research tends to artificially partition the trajectory of data evolution into two orthogonal axes: *Class Incremental Learning* (CIL) and *Domain Incremental Learning* (DIL). As illustrated in Fig. 1a, CIL methods [4–7] assume a consistent domain to focus on semantic expansion, while DIL approaches [8–11] handle domain shifts under a fixed semantic space as Fig. 1b. Even recent unified attempts often rely on idealized assumptions. ICON [12] requires fixed class increments, DisCo [13] presumes that domain shifts strictly synchronize with class task boundaries, and the earlier GIL [14] confines diminishing class increments to initial phases. In contrast, we address a more realistic setting, termed *Class-Domain Incremental Learning* (CDIL), which is characterized by fragmented observations as shown in Fig. 1c. Unlike traditional protocols that ensure complete observations of new classes or domains, CDIL involves learning from partial views with incomplete distributions and irregular category increments. Mastering these fragmented streams is essential for real-world reliability, for instance, when an autonomous vehicle encounters novel obstacles across varying weather. Such incomplete observations inevitably trigger severe feature drift and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Class Incremental Learning (b) Domain Incremental Learning (c) Class-Domain Incremental Learning

Figure 1: Incremental learning scenarios. Colors denote domains and shapes denote classes; dotted arrows indicate the transition across increments. (a) CIL assumes a fixed domain; (b) DIL assumes a fixed semantic space; (c) Our CDIL relaxes these constraints, where new domains and new classes arrive irregularly at any phase, presenting as fragmented increments with varying sizes.

representation imbalance [15]. Consequently, establishing robust decision boundaries becomes exceptionally difficult.

To address fragmented data streams in CDIL, we draw inspiration from the human cognitive ability to synthesize knowledge from varied perspectives [16] and propose EMCift, a model integrating *Extensible Multi-Center Modeling* with *Dual-Path Drift Harmonization*. Under *extensible multi-center modeling*, we treat each class-domain combination as a distinct observational view, allowing the model to synthesize global knowledge from fragmented perspectives without enforcing rigid alignment constraints. To structurally instantiate this multi-view perspective, we utilize a hierarchical tree to organize prototypes. The tree architecture allows the model to incorporate novel domain observations as leaf extensions under class nodes, preserving historical knowledge without disruption. Therefore, our design embraces the information gap inherent in partial observations rather than attempting to eliminate it. This evolving structure, nonetheless, faces the challenge of feature drift, which induces class-wise confusion [17] due to prototype overlap in the compressed feature space. To maintain discriminability within the tree, we propose a *dual-path drift harmonization* strategy that tames feature drift through a decoupled mechanism, mitigating class-wise confusion.

Architecturally, EMCift addresses fragmented data streams through two synergistic components. First, the *Extensible Multi-Center Modeling* establishes a dynamic hierarchy to organize knowledge from scattered views. Within this hierarchy, an *extensible multi-center prototype tree* allows class nodes to spawn domain-specific leaves, elastically accommodating fragmented observations. This multi-center design effectively circumvents the catastrophic forgetting typically caused by forcing diverse domain styles into a rigid, single centroid. To further refine the resulting tree hierarchy, we integrate *inter-view modeling*, which enforces robust separation between classes and counteracts the representation bias derived from inherent data imbalance. Second, to sustain the structural resilience of this tree within CDIL streams, we employ the *Dual-Path Drift Harmonization* strategy. This strategy coordinates two synergistic paths: the *prototype drift path* passively realigns historical leaf nodes to track the evolving feature space, while the *model drift path* actively enforces a robust domain initialization to prevent class-wise confusion. Together, they ensure that leaf-level prototypes remain spatially distinct, preserving the integrity of the evolving hierarchy against catastrophic forgetting. EMCift outperforms state-of-the-art approaches on iDigits, DomainNet, and CORe50 benchmarks, with accuracy gains of **11.52**, **5.54**, and **5.05** percentage points. Our contributions include:

- We explore a more realistic scenario termed CDIL, where models must keep stable learning from fragmented, partial observations, addressing the limitations of rigid completeness assumptions in prior settings.

- We propose EMCift, which synergizes *Extensible Multi-Center Modeling* and *Dual-Path Drift Harmonization*. While the former embraces information gaps via hierarchical knowledge modeling, the latter tames feature drift by decoupling passive geometric realignment from active adversarial regularization, ensuring structural resilience against catastrophic forgetting in CDIL.

- Benchmark evaluations show our EMCift outperforms SOTA methods with significant accuracy gains. Qualitative and ablation analyses further validate the rationality of our multi-center structure with the synergistic strategy in fragmented data streams.

2 Related Work

2.1 Incremental Learning

Incremental learning, often termed continual learning, aims to learn continuously from data streams without catastrophic forgetting [18]. It is often divided into *Class Incremental Learning* (CIL) and *Domain Incremental Learning* (DIL). CIL targets semantic expansion within fixed domains [4–7]. Recent advances like SimpleCIL [19] exploit frozen pre-trained representations for robustness. Conversely, DIL tackles distribution shifts under fixed semantics [20, 21]. For instance, UDIL [10] bridges disparate strategies by adaptive coefficients. Recent works make significant strides to unify these axes. DisCo [13] proposes a prototype pool with contrastive learning to handle domain shifts that strictly align with class task boundaries. ICON [12] establishes cluster-based shift controlled prompts, advancing versatility with fixed class additions. GIL [14] employs vMF models to maintain domain integrity, where class increments only appear in early phases. CIDA [22] utilizes Gaussian prototypes for class-incremental domain adaptation in a static source-to-target setting.

Building on these foundations, our work focuses on a more realistic class-domain incremental learning scenario, where new domains and classes may emerge irregularly with arbitrary class compositions. This setting differs from existing formulations that often rely on fixed class additions, aligned task boundaries, or static source-to-target adaptation, as the model must learn from fragmented observations across both semantic and domain spaces. Accordingly, our multi-center modeling preserves diversity rather than enforcing strict invariance, allowing global patterns to be captured from local observations. Moreover, our dual-path harmonization strategy is designed to balance adaptive plasticity with structural stability.

2.2 Prototype Learning

Prototype-based learning [23] has emerged as a prominent and powerful paradigm, classifying instances via similarity to latent class-representative centroids. This concept has been successfully adapted to few-shot learning [24, 25], zero-shot learning [26–28], interpretability for deep learning [29–31] and tasks in other fields [32, 33].

In incremental learning, prototypes offer a memory-efficient approach to represent historical knowledge. Various representative strategies have been explored. Liu et al. [34] recently established a principled two-stage framework integrating self-supervised training with prototype learning. To address feature drift, pioneering efforts like SDC [35] introduce global compensation. Focusing on domain diversity, MPM [36] constructs a prototype-based graph to minimize per-domain error, and DualCP [37] establishes a rehearsal-free paradigm by leveraging concept generators. Furthermore, ACMap [38] effectively mitigates forgetting through centroid mapping calibration.

Inspired by these advances, we extend prototype learning to a more complex class-domain incremental scenario. Unlike prior work that often focuses on either semantic expansion or domain variation, our multi-center tree accommodates irregular evolution along both dimensions. Moreover, rather than relying on unified updates or static features, we perform leaf-specific geometric realignment to maintain representation precision while preserving the tree topology as the feature space evolves.

3 Methodology

We study the realistic CDIL scenario, where classes and domains evolve irregularly, yielding fragmented data streams. To address this challenge, we organize fragmented observations into a hierarchical multi-center prototype structure. As shown in Fig. 2, EMCift integrates *Extensible Multi-Center Modeling* and *Dual-Path Drift Harmonization*. The former builds a prototype tree in which class nodes branch into domain-specific leaves, with *inter-view modeling* further refining class-wise separation under data imbalance. The latter sustains this dynamic tree by decoupling drift handling into a *prototype drift path* for leaf-wise geometric realignment and a *model drift path* for conditional adversarial regularization.

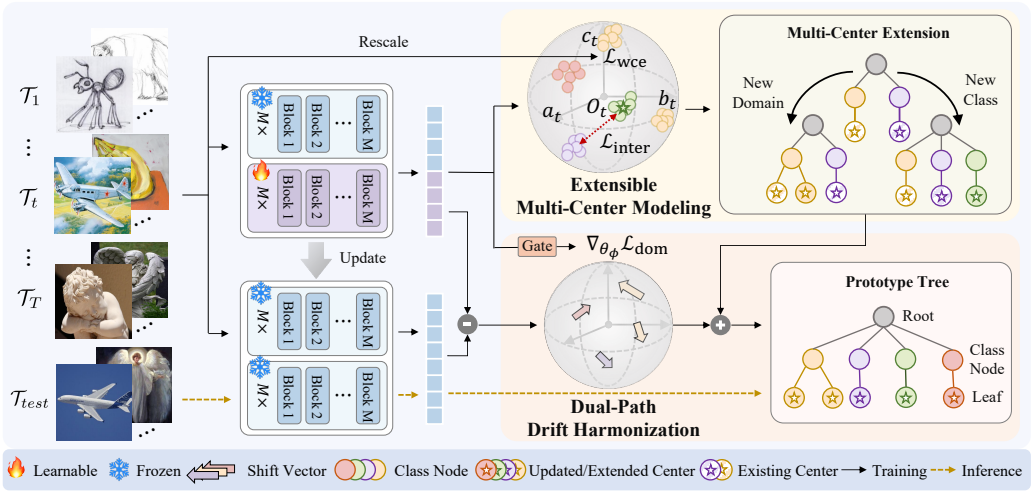


Figure 2: Overview of EMCift, comprising Extensible Multi-Center Modeling and Dual-Path Drift Harmonization. *Training*: Samples from the current phase and memory are mapped into a unified latent space. New cluster centroids form new leaf nodes in the prototype tree, while existing nodes are updated via shift vectors to effectively mitigate catastrophic drift. *Inference*: Input samples are classified using the learned prototype tree without requiring task or domain identity at inference. Extension: New classes trigger new class nodes and leaf prototypes; known classes in new domains trigger additional leaf prototypes under existing nodes to capture evolving data.

3.1 Problem Formulation

Let $\{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$ denote a sequence of data arriving in T incremental learning phases. At phase t, we observe $\mathcal{T}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ drawn from domain $\mathcal{V}_t$ with label set $\mathcal{C}_t$. Our goal is to learn a model over all observed class-domain pairs. During training, domain identity v_t is obtained from acquisition metadata (e.g., camera, hospital, or data-source IDs) and used to organize view-specific prototypes and identify novel domains. At inference, the learned model requires neither task nor domain identity.

Conventional paradigms simplify this evolution by restricting increments to one axis. Specifically, in CIL, the domain remains static ($\mathcal{V}_t = \mathcal{V}_1$) while categories expand; conversely, DIL handles shifts under a fixed semantic space ($\mathcal{C}_t = \mathcal{C}_1$).

In contrast, the proposed CDIL reflects a more realistic scenario where both spaces evolve concurrently and asynchronously. Each task $\mathcal{T}_t$ is associated with a single domain $\mathcal{V}_t$, yet categories may recur across different domains over time. This fragmented and irregular arrival of data, compounded by variations in task scale $|\mathcal{C}_t|$ and sample volume N_t, prevents the model from observing complete distributions at once. Consequently, the learner must synthesize a global knowledge structure from these partial views while extending a non-stationary feature space, ensuring that incremental updates do not lead to the overlap of historical representations.

3.2 Extensible Multi-Center Modeling

Most existing incremental learning approaches [19, 34, 35] rely on a single prototype per class, which compresses intra-class variability under domain shifts in CDIL. To resolve this, we model global knowledge as an *extensible multi-center prototype tree*. The tree root originates a categorical layer of class nodes, which further branch into an observational layer of domain-specific leaves. This tree hierarchy allows the model to extend its knowledge base elastically as new views arrive, as illustrated in the right part of Fig. 2.

Multi-Center Extension. For each class $c = 1, 2, \ldots, C$, we denote its multi-center prototype set as $\mathcal{P}_c = \{\, \boldsymbol{p}_c^v \mid v \in \mathcal{V}_c \,\}$. As the leaf node, each prototype center $\boldsymbol{p}_c^v$ is computed from the feature vectors $\boldsymbol{f}$. Specifically, we denote:

$$\boldsymbol{p}_c^v = \frac{\sum_{i=1}^{|\mathcal{S}^v|} \mathbb{I}(y_i = c)\, \boldsymbol{f}_i}{\sum_{i=1}^{|\mathcal{S}^v|} \mathbb{I}(y_i = c)}, \tag{1}$$

where $\mathcal{S}^v$ represents the sample set in domain view v, and $\mathbb{I}(\cdot) \in \{0, 1\}$ denotes the indicator function. The feature $\boldsymbol{f}_i = [\phi^*(\boldsymbol{x_i}), \phi(\boldsymbol{x_i})]$ is constructed by concatenating outputs from a frozen pre-trained model $\phi^*(\cdot)$ and a trainable adapter $\phi(\cdot)$.

For any sample $\boldsymbol{x}$, our model enables class recognition without task or domain information. Leveraging the concatenated feature $\boldsymbol{f} = [\phi^*(\boldsymbol{x}), \phi(\boldsymbol{x})]$, the prototype tree performs a bottom-up inference. At its core is a multi-center similarity metric derived from the tree's multi-branch structure. The categorical score (CS) is determined by the maximum similarity between the sample feature and any prototype center of the class, which can be formulated as an inner product:

$$CS(c, \boldsymbol{x}) = \max_{p_c^v \in \mathcal{P}_c} \left\{ \langle \hat{\boldsymbol{p}}_c^v, \hat{\boldsymbol{f}} \rangle \right\} = \max_{p_c^v \in \mathcal{P}_c} \left\{ \left(\frac{\boldsymbol{p}_c^v}{\|\boldsymbol{p}_c^v\|_2} \right)^\top \left(\frac{\boldsymbol{f}}{\|\boldsymbol{f}\|_2} \right) \right\} , \tag{2}$$

where $\hat{\boldsymbol{p}}_c^v$ and $\hat{\boldsymbol{f}}$ denote the normalized prototype center and sample feature, respectively. With the categorical score, we obtain the final prediction:

$$P(y = c \mid \boldsymbol{x}) = \frac{\exp\left(\max_{p_c^v \in \mathcal{P}_c} \left\{ \hat{\boldsymbol{p}}_c^{v\top} \hat{\boldsymbol{f}} \right\} \right)}{\sum_{c' \in \mathcal{C}} \exp\left(\max_{p_{c'}^v \in \mathcal{P}_{c'}} \left\{ \hat{\boldsymbol{p}}_{c'}^{v\top} \hat{\boldsymbol{f}} \right\} \right)} , \tag{3}$$

where $\mathcal{C}$ is the set of classes encountered so far.

In fact, the multi-center prototype is a generalized extension of the traditional single-center prototype. When class c degenerates to a single center, its multi-center prototype branch $\mathcal{P}_c = \{ \boldsymbol{p}_c^v \mid v \in \mathcal{V}_c \}$ degenerates to an undifferentiated single leaf $\boldsymbol{p}_c$, and the categorical score reduces to $\langle \hat{\boldsymbol{p}}_c, \hat{\boldsymbol{f}} \rangle$. Multi-center inference scales linearly with the number of observed class-domain pairs, $M_t = |\{(c, v) : (c, v) \text{ observed by phase } t\}|$, with theoretical and instance analysis detailed in Appendix E.

Inter-View Modeling. In the CDIL scenario, the semantic space grows under our multi-center modeling as new views are introduced. Enforcing intra-class alignment across domains inhibits semantic diversity. We therefore focus on inter-class separation to construct global knowledge. To this end, we introduce an inter-view constraint $\mathcal{L}_{\text{inter}}$, formulated as a multi-center contrastive loss. It is computed only when inter-class pairs are present in the batch as follows:

$$\mathcal{L}_{\text{inter}} = \frac{\|\mathbf{M} \odot [\gamma \cdot \mathbf{I} - \mathbf{B}]_+\|_1}{\|\mathbf{M}\|_1} , \tag{4}$$

where γ is a margin, $\mathbf{I}$ is an all-ones matrix with the same shape as $\mathbf{B}$, $\mathbf{M}_{ij} = \mathbb{I}(y_i \neq y_j)$ selects inter-class pairs, and $\mathbf{B}_{ij} = \|\hat{\boldsymbol{f}}_i - \hat{\boldsymbol{f}}_j\|_2$ measures feature dissimilarity.

Unlike objectives that attract all samples of a class to a common center, $\mathcal{L}_{\text{inter}}$ penalizes only pairs with different semantic labels. Therefore, it promotes category separation without explicitly compressing the cross-domain observations of the same category. This is complementary to the hierarchical structure, where the centers of each domain retain independent observation perspectives, while the marginal terms suppress the overlap between different categories.

Furthermore, to mitigate frequency imbalance, we employ a weighted cross-entropy loss $\mathcal{L}_{\text{wce}}$. We assign a calibration weight to each class-domain pair, set inversely proportional to its frequency. Specifically, we define $w_c^v = (|\mathcal{S}_c^v| \cdot \bar{w})^{-1}$, where $\bar{w}$ is a normalization factor over all seen pairs. Denoting the predicted logit as $z_{i,c}$, the loss is formulated as:

$$\mathcal{L}_{\text{wce}} = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i}^{v_i} \log \left(\frac{\exp(z_{i,y_i})}{\sum_{c=1}^{C} \exp(z_{i,c})} \right) , \tag{5}$$

where $w_{y_i}^{v_i}$ adjusts the contribution of the i-th sample based on its specific class-domain pair, preventing the optimization from being skewed by frequency imbalance.

3.3 Dual-Path Drift Harmonization

The dynamic evolution of our multi-center tree confronts temporal misalignment of historical leaves and structural collapse induced by domain overfitting. We propose *dual-path drift harmonization* to selectively accommodate necessary adaptation while suppressing detrimental drift. Specifically, the *prototype drift path* passively realigns the past via geometric updates, while the *model drift path* actively regularizes the future to prevent manifold compression.

Prototype Drift Path. The efficacy of our extensible multi-center modeling hinges on the precision of its leaf nodes. As shown in Fig. 3, updating the backbone from ϕ_{t-1} to ϕ_t causes historical prototypes $\{\boldsymbol{p}^v_{c,t-1}\}_{v\in\mathcal{V}_c}$ to lag behind the evolving feature space $\phi_t(\mathcal{X})$, leading to temporal misalignment.

Crucially, a uniform update per class cannot capture view-dependent drift. Therefore, we propose a prototype drift strategy. By estimating unique displacement vectors per leaf, we preserve the multi-center tree topology within the new feature space.

Specifically, for each leaf node corresponding to class c in domain v, we utilize the memory buffer to capture its specific centroid shift. Let $\mathcal{D}^v_c$ denote the data distribution for this leaf. The leaf-wise displacement vector $\boldsymbol{\delta}^v_{c,t}$ is defined as the expected feature difference:

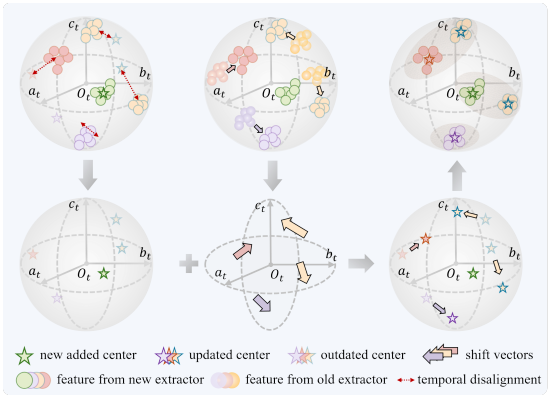


Figure 3: Feature cluster and prototype shift. Colors denote categories. *Left*: Extract prototypes from new data as new leaf nodes. *Middle*: Estimate shift vectors with memory buffer. *Second row*: Update old prototype centers by shift vectors. *Upper-right*: Class and domain distributions in the feature space.

$$\boldsymbol{\delta}^v_{c,t} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}^v_c}\left[\phi_t(\boldsymbol{x}) - \phi_{t-1}(\boldsymbol{x})\right] . \tag{6}$$

In practice, we estimate these vectors using the corresponding memory subset $\mathcal{M}^v_c$ and independently update each old leaf node in the prototype tree:

$$\hat{\boldsymbol{\delta}}^v_{c,t} = \frac{1}{|\mathcal{M}^v_c|}\sum_{\boldsymbol{x}\in\mathcal{M}^v_c}\left[\phi_t(\boldsymbol{x}) - \phi_{t-1}(\boldsymbol{x})\right] , \quad \boldsymbol{p}^v_{c,t} = \boldsymbol{p}^v_{c,t-1} + [\mathbf{0}, \hat{\boldsymbol{\delta}}^v_{c,t}] , \tag{7}$$

where $\mathbf{0}$ represents feature dimensions of frozen components. This compensation corrects misalignment without distorting the topological relationships between domain views. Further analysis of this compensation is provided in Appendix B.

Model Drift Path. Complementing the prototype update, the model needs to adapt to new domain views without inducing excessive feature compression that leads to class-wise confusion. To rectify this catastrophic forgetting, we propose a conditional adversarial drift regularization grounded in the generalization error bound of domain adaptation.

Inspired by the domain-adaptation theory [39], we instantiate the source-target relation in the irregular CDIL progression. At phase t, the incoming domain distribution $\mathcal{D}^{v_t}$ serves as the current source of supervision, while the historical distribution $\mathcal{D}^{t-1}_H$ summarizes previously observed domains $\mathcal{V}^{t-1}_H$. After incorporating the incoming view, the accumulated distribution becomes $\mathcal{D}^t_H = \mathrm{Mix}(\mathcal{D}^{t-1}_H, \mathcal{D}^{v_t})$, forming a dynamic mixture over the updated domain set $\mathcal{V}^t_H = \mathcal{V}^{t-1}_H \cup \{v_t\}$. For a hypothesis $h \in \mathcal{H}$, the error on this accumulated mixture can be bounded as:

$$\epsilon_{\mathcal{D}^t_H}(h) \le \epsilon_{\mathcal{D}^{v_t}}(h) + \tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{v_t}, \mathcal{D}^t_H) + \lambda_t, \tag{8}$$

where $\epsilon_{\mathcal{D}^{v_t}}(h)$ is the population risk on the incoming distribution, and $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{v_t}, \mathcal{D}^t_H)$ measures its divergence from the accumulated mixture. The joint optimality term is $\lambda_t = \epsilon_{\mathcal{D}^{v_t}}(h^*_t) + \epsilon_{\mathcal{D}^t_H}(h^*_t)$, where $h^*_t = \arg\min_{h\in\mathcal{H}}(\epsilon_{\mathcal{D}^{v_t}}(h) + \epsilon_{\mathcal{D}^t_H}(h))$ denotes the best shared hypothesis between the incoming distribution and the accumulated mixture. This term captures the intrinsic adaptability limit under heterogeneous class-domain observations, which can be alleviated by our multi-center strategy.

Beyond minimizing the population risk $\epsilon_{\mathcal{D}^{v_t}}(h)$, Eq. (8) hinges on balancing $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{v_t}, \mathcal{D}^t_H)$ and λ_t. When $v_t \notin \mathcal{V}^{t-1}_H$, the novel distribution $\mathcal{D}^{v_t}$ may cause a divergence spike, and adversarial alignment helps incorporate this new view. In contrast, once $v_t \in \mathcal{V}^{t-1}_H$, the historical mixture has already covered this domain view. Continuing to minimize $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{v_t}, \mathcal{D}^t_H)$ may suppress the feature space and hurt the attainable joint term. Therefore, we gate the adversarial drift regularization by domain novelty:

$$g_t = \mathbb{I}(v_t \notin \mathcal{V}^{t-1}_H) \in \{0, 1\}. \tag{9}$$

Table 1: Quantitative comparison of EMCift and state-of-the-art methods on iDigits ($T = 19$), DomainNet ($T = 167$), and CORe50 ($T = 56$). $\bar{A}$, A_T, and AF denote Average Incremental Accuracy, Last-Phase Accuracy, and Average Forgetting. Arrows ($\uparrow, \downarrow$) indicate better performance. Results are averaged over 3 runs, reporting the mean and standard deviation. Bold text indicates the best performance.

Methods	iDigits			DomainNet			CORe50		
	$\bar{A}$ (%) $\uparrow$	A_T (%) $\uparrow$	AF (%) $\downarrow$	$\bar{A}$ (%) $\uparrow$	A_T (%) $\uparrow$	AF (%) $\downarrow$	$\bar{A}$ (%) $\uparrow$	A_T (%) $\uparrow$	AF (%) $\downarrow$
LwF [6]	40.07±0.29	29.35±2.54	69.02±3.56	5.65±0.14	3.57±0.17	67.81±0.52	35.61±0.52	42.93±1.85	54.00±1.71
iCaRL [7]	61.99±2.07	47.39±4.71	54.73±4.38	42.43±0.51	30.48±0.65	51.22±0.23	84.05±0.56	87.93±0.47	5.53±0.34
SDC [35]	48.77±0.62	30.07±3.24	75.45±8.24	35.44±0.81	22.91±0.22	56.78±0.10	82.49±0.39	84.49±0.64	8.30±0.66
AFC [45]	67.46±0.90	74.63±2.53	24.76±3.70	38.06±1.40	36.29±0.88	36.54±0.63	81.99±0.93	86.53±0.49	8.33±0.70
L2P [4]	66.35±1.23	57.73±1.41	22.27±0.54	57.71±0.24	50.03±0.23	12.46±2.02	81.96±1.05	85.97±0.96	2.98±0.82
DualPrompt [5]	72.97±1.52	67.27±1.69	19.47±0.52	59.93±0.36	51.37±0.70	14.13±0.27	84.67±0.29	86.27±1.03	3.87±0.92
ICON [12]	76.13±0.99	74.00±1.17	14.05±1.05	61.00±0.21	53.66±0.14	**11.98±0.23**	85.64±0.24	87.89±0.26	3.72±0.44
SimpleCIL [19]	44.01±0.00	14.16±0.00	69.27±0.00	39.78±0.00	25.13±0.00	43.00±0.00	59.79±0.00	53.91±0.00	18.72±0.00
APER Adapter [19]	51.28±0.64	33.58±2.45	51.95±1.53	35.13±2.86	22.38±1.85	34.10±4.25	69.06±2.09	66.35±1.94	10.59±0.19
EMCift (Ours)	**87.65±0.76**	**85.85±0.56**	**10.45±0.52**	**66.54±0.21**	**58.70±0.29**	17.54±0.15	**90.69±0.27**	**92.19±0.37**	**2.27±0.28**

With the conditional adversarial gate, we employ a domain discriminator ψ to regularize the learnable adapter ϕ via a gated adversarial objective. The optimization problem and the corresponding update rule for ϕ are formulated as:

$$\min_{\theta_\phi} \max_{\theta_\psi} \mathcal{J}_t = \mathcal{L}_{task}(\theta_\phi) - \beta\, g_t\, \mathcal{L}_{dom}(\theta_\phi, \theta_\psi), \tag{10}$$

where $\mathcal{L}_{task} = \mathcal{L}_{\text{wce}} + \alpha\, \mathcal{L}_{\text{inter}}$ represents primary task classification loss, $\mathcal{L}_{dom}$ denotes the domain adversarial loss. This design matches the bound in Eq. (8). When $v_t \notin \mathcal{V}_H^{t-1}$, reducing divergence is beneficial for incorporating a novel domain view. Once $v_t \in \mathcal{V}_H^{t-1}$, enforcing additional invariance unnecessarily restricts the hypothesis space and thus cannot improve the joint optimality term. Detailed analyses are in Appendix C.

4 Experiments

4.1 Experimental Settings

Datasets and Protocols. To evaluate the proposed realistic setting, we conduct experiments on three benchmarks with increasing complexity: **iDigits** [40], **CORe50** [41], **DomainNet** [42]. These datasets feature strong domain shifts and diverse categories, mirroring real-world streams. Maintaining official splits, we reconstruct CDIL task sequences. To reflect irregularity, $|\mathcal{C}_t|$ varies dynamically: 1–3 for iDigits, 5–10 for CORe50, 10–15 for DomainNet. Details are provided in Appendix A.

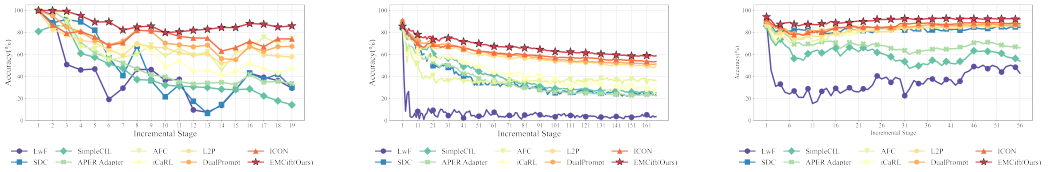
Evaluation Metrics. Following standard protocols [43, 44], we report: (1) **Average Accuracy** ($\bar{A}$): $\bar{A} = \frac{1}{T}\sum_{t=1}^{T} A_t$, where A_t is the accuracy on all seen tasks at phase t; (2) **Last-Phase Accuracy** (A_T): the final accuracy after the entire sequence; (3) **Average Forgetting** (AF): measuring the stability of historical knowledge, defined as $AF = \frac{1}{T-1}\sum_{j=1}^{T-1} \max_{l\in\{1,\dots,T-1\}}(a_{l,j} - a_{T,j})$, where $a_{l,j}$ denotes the accuracy on task j after learning task l.

Implementation Details. We employ a unified ViT-B/16 backbone (ImageNet-21K pre-trained). Training utilizes AdamW on GPU v100. We use only 1 exemplar per class-domain pair via herding [7].

4.2 Comparison to the State-of-the-Art Methods

We compare EMCift against nine representative incremental learning approaches: LwF [6], iCaRL [7], AFC [45]; SDC [35], L2P [4], DualPrompt [5], ICON [12], SimpleCIL [19], APER Adapter [19]. To ensure a fair comparison, all methods employ a unified ViT-B/16 backbone (ImageNet-21K pre-trained), with non-ViT baselines adapted accordingly. For replay-based methods (iCaRL and AFC), we use the same size of memory buffer as ours.

Results on iDigits Dataset. As shown in Tab. 1, on iDigits ($T = 19$), characterized by diverse digit styles and sharp style transitions, EMCift outperforms ICON by 11.52% ($\bar{A}$). Moreover, in Fig. 4a, the jagged curves of baselines suggest distinct failure modes. Single-center methods (iCaRL) and global drift compensation (SDC) struggle with sharp covariate shifts, while ICON's weight-space orthogonality regularization restricts the plasticity needed for data streams. In contrast, EMCift



(a) iDigits (19 tasks). EMCift maintains consistent superiority despite domain shifts.

(b) DomainNet (167 tasks). EMCift demonstrates robust long-term retention.

(c) CORe50 (56 tasks). EMCift achieves consistently high accuracy and low forgetting.

Figure 4: Incremental accuracy across three benchmarks. Each subfigure shows per-task accuracy over the full incremental sequence.

circumvents these limitations via domain-specific leaf nodes within its multi-center tree, where the dual-path harmonization prevents class confusion by leaf-level compensation vectors and conditional regularization. This synergy ensures a stable accuracy curve over static baselines like SimpleCIL.

Results on DomainNet Dataset. On DomainNet ($T = 167$), featuring extensive categories and complex shifts, EMCift leads with 66.54% $\bar{A}$ and 58.70% A_T. Prompt-based methods (L2P, DualPrompt, ICON) show resilience but face plasticity limits across diverse domains. For instance, ICON achieves low forgetting but suboptimal overall accuracy, suggesting a trade-off where plasticity is sacrificed for interference mitigation. Consistent with iDigits, single-center methods suffer from forgetting as the lack of domain-awareness leads to class confusion. These results demonstrate that EMCift tames long-term drift via domain-specific leaves while capturing complex variations.

Results on CORe50 Dataset. Tab. 1 and Fig. 4c present a comprehensive evaluation on the CORe50 dataset ($T = 56$), which is characterized by its object-centric nature and significant environmental variations. As shown in Fig. 4c, EMCift consistently maintains the highest accuracy throughout all stages, exhibiting remarkable robustness against catastrophic forgetting. Quantitatively, EMCift achieves a leading $\bar{A}$ of 90.69% and A_T of 92.19%, surpassing the strongest baseline for each metric by 5.05 and 4.26 percentage points, respectively.

Pair-as-Class Adaptation. To examine a simpler alternative for handling class-domain variations, we adapt nine standard continual learning baselines using a Pair-as-Class strategy, assigning each observed class-domain pair an independent training label and mapping pair-level predictions back to semantic classes for evaluation. As reported in Appendix I.1, EMCift achieves higher average incremental accuracy than the strongest Pair-as-Class baseline by 14.45, 1.24, and 9.13 percentage points on iDigits, DomainNet, and CORe50, respectively. The Pair-as-Class baselines also use domain metadata when constructing training labels for observed pairs, helping distinguish the benefits of our modeling and drift handling from metadata access alone.

Training-Time Domain Information. To assess performance without training-time domain metadata, we use phase indices as views and disable adversarial alignment. On iDigits, this Task-as-View variant achieves $83.67 \pm 0.93\%$ average accuracy, versus $87.65 \pm 0.76\%$ for standard EMCift, as detailed in Appendix I.2. This comparison changes both view grouping and alignment, while neither setting requires task or domain identity at inference.

Further experiments in Appendix I cover PACS [46], randomized task orders, standard CIL/DIL settings, and backbone/pretraining robustness.

4.3 Ablation Studies

To validate the efficacy of EMCift, we conduct comprehensive ablation studies on iDigits shown in Tab. 2. More detailed results are provided in Appendix G.

Impact of Extensible Multi-Center Modeling. The structural advantage is evident in the transition from the single-center baseline (Row 1) to multi-center modeling (Row 2), which yields an 8.9% gain. This confirms that a hierarchical prototype tree is essential for disentangling domain diversity in our model. Integrating inter-view modeling (Row 3) further boosts performance by structuring the semantic space. Crucially, replacing our inter-view loss with traditional triplet loss (Row 4) causes a

Table 2: Ablation study of proposed components on the iDigits dataset. **Multi**: Multi-Center Prototype Tree. **IVM**: Inter-View Modeling, including class balancing and inter-view loss. **P-Drift / M-Drift**: Prototype and Model Drift Paths. †: Replaces our inter-view loss $\mathcal{L}_{\text{inter}}$ with a traditional triplet loss. ‡: Replaces our drift path with continuous adversarial training. Results are selected based on the optimal $\bar{A}$.

Row	Multi	IVM	P-Drift	M-Drift	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
1					67.08	77.77	21.79
2	✓				75.98	70.37	29.84
3	✓	✓			82.77	83.30	13.28
4	✓	†	✓	✓	67.63	74.50	25.96
5	✓	✓	✓		83.76	80.18	14.09
6	✓	✓		✓	84.21	83.14	15.23
7	✓	✓	✓	‡	81.33	81.26	15.77
8	✓	✓	✓	✓	**88.51**	**86.47**	**10.00**

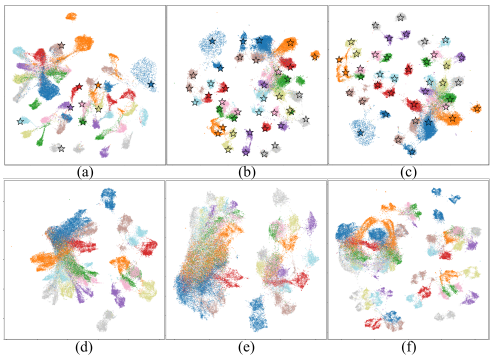


Figure 5: Feature visualization on iDigits, ☆: prototype centers. (**a-c**) Component ablation: (**a**) single-center baseline, (**b**) w/o P-Drift, (**c**) EMCift. (**d-f**) Baselines ICON, L2P, and DualPrompt show overlapping clusters.

drastic drop to 67.63%. This empirically proves that simple contrastive learning is insufficient for preserving global semantic coherence in evolving streams.

Efficacy of Dual-Path Harmonization. The full model achieves a peak performance of 88.51% by harmonizing stability and plasticity (Row 8). Compared with Row 5, incorporating the conditional model drift path further improves $\bar{A}$ by 4.75 percentage points and reduces forgetting by 4.09 percentage points. The prototype drift path serves as the stability anchor and removing it leads to severe forgetting with AF reaching 15.23% (Row 6) due to the temporal misalignment of leaves. On the model drift path, continuous adversarial training yields an accuracy of only 81.33% (Row 7), falling behind the 83.76% achieved by omitting the path entirely (Row 5). This degradation implies that permanent alignment over-regularizes the feature space. Conversely, our conditional strategy limits regularization strictly, successfully preventing structural collapse. In contrast, our conditional strategy applies alignment selectively, avoiding unnecessary constraints on the evolving representations. Together, the two paths complement each other by preserving historical prototype consistency while allowing the feature space to adapt to evolving domain distributions.

Qualitative Analysis. Fig. 5(a-c) illustrates how the proposed components affect feature distributions and prototype positioning, where stars denote prototype centers. In Fig. 5a, the single-center baseline struggles to accommodate domain-specific variations, forcing fragmented feature clusters into an inadequate shared representation. This leads to dispersed distributions and ambiguous cluster boundaries. In Fig. 5b, although multi-center modeling separates different domain-specific clusters, removing P-Drift causes historical prototypes to gradually deviate from their corresponding feature centers. Some prototypes even become misaligned with the evolving class distributions, weakening their representativeness. In contrast, Fig. 5c shows that EMCift maintains better alignment between prototypes and their corresponding feature clusters through geometric drift compensation. This demonstrates how multi-center modeling and prototype drift compensation jointly preserve the structure and representativeness of evolving class-domain distributions.

4.4 Visualization and Analysis

Fig. 5(c-f) visualizes and compares the feature spaces of EMCift and three strong baselines. Notably, fragmented sub-clusters within single classes are observed across all methods, reflecting the intrinsic multi-modal nature of domain-shifting streams in the CDIL scenario. As shown in Fig. 5d-e, clusters in the baselines are inherently loose, failing to maintain clear boundaries and leading to significant overlap and categorical confusion. This can be attributed to their single-center assumptions, which cannot effectively structure fragmented streams. In contrast, as seen in Fig. 5c, EMCift utilizes the extensible multi-center prototype tree to model each fragment as a domain-specific leaf node. This multi-center modeling ensures that each category maintains a compact, well-separated cluster, empirically validating EMCift's ability to effectively construct a coherent global knowledge base. More visuals are provided in Appendix H.

4.5 Empirical Efficiency Analysis

To assess the computational overhead of our multi-center tree, we evaluate inference latency for our model and strong baselines using CUDA events averaged over 100 iterations. On iDigits, EMCift takes 19.72 ms, compared with ICON's 5.82 ms (approximately $3.39\times$ the latency). Its average incremental accuracy is 87.65%, exceeding ICON by 11.52 percentage points. Table 3 uses the three-run mean accuracies from Table 1. These results quantify the accuracy–latency trade-off under the evaluated setup; the prototype-scoring cost scales linearly with the number of observed class-domain pairs, as discussed in Appendix E.

Table 3: Accuracy–efficiency comparison on iDigits. Accuracies are three-run means from Table 1.

Method	Inference Time ↓	$\bar{A}$ (%) ↑
ICON [12]	**5.82 ms**	76.13
AFC [45]	6.15 ms	67.46
iCaRL [7]	12.78 ms	61.99
DualPrompt [5]	14.99 ms	72.97
EMCift (Ours)	19.72 ms	**87.65**

4.6 Real-World Stream Analysis

To better reflect real-world streams, we add a metadata-derived stream from a subset of iWildCam 2022 [47]. Using the provided species annotations as semantic classes and camera locations as domains, we group images into camera-month fragments ordered by recorded acquisition time. For this added experiment, the stream is fixed using only dataset-provided timestamps and camera IDs, with no model output or performance involved, and the identical stream is used for every method. It contains 7 increments and 6 recurring camera domains, with classes recurring across domains and naturally uneven class compositions across increments. Across three runs, EMCift achieves the best $\bar{A}$ and A_T while remaining comparable in AF. These results show that EMCift's advantage persists in a naturally ordered, metadata-defined stream and is not limited to the controlled fragmentation protocol.

Table 4: Comparison with standard baselines on the iWildCam-TinyPlus stream. Results are mean±std over three runs.

Method	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
LwF	45.19±3.55	39.55±0.00	18.16±2.10
SDC	48.79±1.49	38.81±3.25	29.10±5.24
AFC	47.88±0.97	31.84±1.73	35.99±2.64
iCaRL	52.73±2.08	46.52±1.14	22.34±1.48
APER	49.32±0.98	36.07±1.55	36.27±1.26
SimpleCIL	49.98±0.00	34.33±0.00	41.17±0.00
L2P	50.90±0.72	46.27±1.97	**13.79±3.89**
DualPrompt	53.99±2.02	48.51±0.75	15.06±2.06
ICON	57.41±2.35	49.25±1.97	15.58±4.92
EMCift (Ours)	**66.46±0.09**	**53.73±2.24**	13.81±0.38

5 Conclusion

We propose EMCift to learn a global knowledge representation from fragmented incremental tasks within the CDIL scenario. By organizing data within an extensible multi-center prototype tree, we address the intrinsic diversity where the traditional single-center assumption fails. Theoretical analysis and qualitative visualizations further confirm the rationality of this multi-center structure in capturing class-domain distributions. To sustain the tree topology, our dual-path drift harmonization synergizes geometric compensation with conditional adversarial regularization, ensuring stable feature space expansion. Superior performance on iDigits, DomainNet and CORe50 benchmarks validates the efficacy of our model. Ultimately, our work offers a comprehensive solution for incremental learning in evolving environments, demonstrating that embracing information gaps and taming feature drift facilitates the organic growth of global knowledge from fragmented real-world observations.

Broader Impact. EMCift facilitates continual knowledge acquisition from evolving classes and domains, offering a solution for long-term adaptation to heterogeneous open-world environments.

Limitations. The multi-center tree requires storage that grows linearly with the number of observed class-domain pairs. Dynamic leaf-pruning could improve scalability in long-term open-world streams. Although leaf-wise displacement estimates have theoretically bounded variance, more robust estimation under limited memory budgets could improve compensation in the presence of outliers.

Acknowledgments and Disclosure of Funding

This research received no external funding. The authors declare no competing interests.

References

[1] Bing Liu. Lifelong machine learning: a paradigm for continuous learning. *Frontiers of Computer Science*, 11(3):359–361, 2017.

[2] Sheng Luo, Wei Chen, Wanxin Tian, Rui Liu, Luanxuan Hou, Xiubao Zhang, Haifeng Shen, Ruiqi Wu, Shuyi Geng, Yi Zhou, et al. Delving into multi-modal multi-task foundation models for road scene understanding: From learning paradigm perspectives. *IEEE Transactions on Intelligent Vehicles*, 2024.

[3] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[4] 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*, pages 139–149, 2022.

[5] 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*, pages 631–648. Springer, 2022.

[6] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[7] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017.

[8] M Jehanzeb Mirza, Marc Masana, Horst Possegger, and Horst Bischof. An efficient domain-incremental learning approach to drive in all weather conditions. In *CVPR*, pages 3001–3011, 2022.

[9] Christiaan Lamers, René Vidal, Nabil Belbachir, Niki van Stein, Thomas Bäeck, and Paris Giampouras. Clustering-based domain-incremental learning. In *ICCV*, pages 3384–3392, 2023.

[10] Haizhou Shi and Hao Wang. A unified approach to domain incremental learning with memory: Theory and algorithm. *NeurIPS*, 36:15027–15059, 2023.

[11] Gen Luo, Jiamu Sun, Lei Jin, Yiyi Zhou, Qiang Xu, Rongrong Fu, Xiaoshuai Sun, and Rongrong Ji. Domain incremental learning for object detection. *Pattern Recognition*, 170:111882, 2026.

[12] Min-Yeong Park, Jae-Ho Lee, and Gyeong-Moon Park. Versatile incremental learning: Towards class and domain-agnostic incremental learning. In *ECCV*, pages 271–288. Springer, 2024.

[13] Wei Chen and Yi Zhou. Make domain shift a catastrophic forgetting alleviator in class-incremental learning. In *AAAI*, volume 39, pages 15866–15874, 2025.

[14] Jiangwei Xie, Shipeng Yan, and Xuming He. General incremental learning with domain-aware categorical representations. In *CVPR*, pages 14351–14360, 2022.

[15] Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

[16] Na Luo, Weiyang Shi, Zhengyi Yang, Ming Song, and Tianzi Jiang. Multimodal fusion of brain imaging data: Methods and applications. *Machine Intelligence Research*, 21(1):136–152, 2024.

[17] Huitong Chen, Yu Wang, Yan Fan, Guosong Jiang, and Qinghua Hu. Reducing class-wise confusion for incremental learning with disentangled manifolds. In *CVPR*, pages 10121–10130, 2025.

[18] Neil Ashtekar, Jingxi Zhu, and Vasant G Honavar. Class incremental learning from first principles: A review. *Transactions on Machine Learning Research*, 2025.

[19] 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. *International Journal of Computer Vision*, 133(3):1012–1032, 2025.

[20] Qiang Wang, Yuhang He, Songlin Dong, Xinyuan Gao, Shaokun Wang, and Yihong Gong. Non-exemplar domain incremental learning via cross-domain concept integration. In *ECCV*, pages 144–162. Springer, 2024.

[21] 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 *CVPR*, pages 20547–20557, 2025.

[22] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-incremental domain adaptation. In *ECCV*, pages 53–69. Springer, 2020.

[23] Michael Biehl, Barbara Hammer, and Thomas Villmann. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(2):92–111, 2016.

[24] Baoquan Zhang, Xutao Li, Shanshan Feng, Yunming Ye, and Rui Ye. Metanode: Prototype optimization as a neural ode for few-shot learning. In *AAAI*, volume 36, pages 9014–9021, 2022.

[25] Hao Zhu and Piotr Koniusz. Transductive few-shot learning with prototype-based label propagation by iterative graph refinement. In *CVPR*, pages 23996–24006, 2023.

[26] Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. In *ICLR*, 2025.

[27] Wenjin Hou, Shiming Chen, Shuhuang Chen, Ziming Hong, Yan Wang, Xuetao Feng, Salman Khan, Fahad Shahbaz Khan, and Xinge You. Visual-augmented dynamic semantic prototype for generative zero-shot learning. In *CVPR*, pages 23627–23637, 2024.

[28] Jiamin Wu, Tianzhu Zhang, Zheng-Jun Zha, Jiebo Luo, Yongdong Zhang, and Feng Wu. Prototype-augmented self-supervised generative network for generalized zero-shot learning. *IEEE Transactions on Image Processing*, 33:1938–1951, 2024.

[29] Aaron J Li, Robin Netzorg, Zhihan Cheng, Zhuoqin Zhang, and Bin Yu. Improving prototypical visual explanations with reward reweighing, reselection, and retraining. In *ICML*, pages 28466–28479, 2024.

[30] Chiyu Ma, Jon Donnelly, Wenjun Liu, Soroush Vosoughi, Cynthia Rudin, and Chaofan Chen. Interpretable image classification with adaptive prototype-based vision transformers. *NeurIPS*, 37:41447–41493, 2024.

[31] Mikołaj Sacha, Bartosz Jura, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Interpretability benchmark for evaluating spatial misalignment of prototypical parts explanations. In *AAAI*, volume 38, pages 21563–21573, 2024.

[32] Yanjie Wang, Liqun Chen, Tianming Zhao, Tao Zhang, Guodong Wang, Luxin Yan, Sheng Zhong, Jiahuan Zhou, and Xu Zou. High-dimension prototype is a better incremental object detection learner. In *ICLR*, 2025.

[33] Yuchun Fang, Chen Chen, Wei Zhang, Jiahua Wu, Zhaoxiang Zhang, and Shaorong Xie. Prototype learning for adversarial domain adaptation. *Pattern Recognition*, 155:110653, 2024.

[34] Wenzhuo Liu, Xin-Jian Wu, Fei Zhu, Ming-Ming Yu, Chuang Wang, and Cheng-Lin Liu. Class incremental learning with self-supervised pre-training and prototype learning. *Pattern Recognition*, 157:110943, 2025.

[35] 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*, pages 6982–6991, 2020.

[36] Haopeng Sun, Yingwei Zhang, Lumin Xu, Sheng Jin, Ping Luo, Chen Qian, Wentao Liu, and Yiqiang Chen. Unsupervised continual domain shift learning with multi-prototype modeling. In *CVPR*, pages 10131–10141, 2025.

[37] Qiang Wang, Yuhang He, Songlin Dong, Xiang Song, Jizhou Han, Haoyu Luo, and Yihong Gong. Dualcp: Rehearsal-free domain-incremental learning via dual-level concept prototype. In *AAAI*, volume 39, pages 21198–21206, 2025.

[38] Takuma Fukuda, Hiroshi Kera, and Kazuhiko Kawamoto. Adapter merging with centroid prototype mapping for scalable class-incremental learning. In *CVPR*, pages 4884–4893, 2025.

[39] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1): 151–175, 2010.

[40] Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *CVPR*, pages 4443–4453, 2021.

[41] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *CoRL*, pages 17–26. PMLR, 2017.

[42] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019.

[43] 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*, volume 38, pages 17237–17244, 2024.

[44] Arslan Chaudhry, Ranzato Marc'Aurelio, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019.

[45] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *CVPR*, pages 16071–16080, 2022.

[46] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[47] JohnBeuving, Phil Culliton, sbeery, and Ștefan Istrate. iwildcam 2022 - fgvc9. `https://www.kaggle.com/competitions/iwildcam2022-fgvc9`, 2022. Kaggle.

[48] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[49] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

A Formal Definition of CDIL Protocol Construction

To rigorously define the *Class-Domain Incremental Learning* (CDIL) setting, we formulate the protocol that transforms a static dataset $\mathbb{D} = \{(x_i, y_i, v_i)\}$ with ground-truth domain labels $v_i \in \mathcal{V}_{orig}$ into a sequential stream $\mathcal{S} = \{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$ via a fragmentation-and-permutation process.

We initiate the fragmentation by separating the dataset according to its natural domains. We maintain the original train-test split and utilize the dataset-provided domain attributes as the invariant domain set $\mathcal{V}_{orig}$. For each domain $v \in \mathcal{V}_{orig}$, we retrieve the domain-specific subset $\mathbb{D}^v = \{(x, y) \mid v_i = v\}$. To simulate the fragmented observations described in the problem formulation, each domain subset $\mathbb{D}^v$ is randomly partitioned into k_v disjoint fragments based on category, denoted as $\mathbb{F}^v = \{S_{v,1}, S_{v,2}, \ldots, S_{v,k_v}\}$. The parameter k_v determines the task granularity. Specifically, each fragment $S_{v,i}$ samples a variable number of classes within the domain, consistent with the experimental settings: 1–3 for iDigits, 5–10 for CORe50, and 10–15 for DomainNet.

After fragmentation, we apply a random permutation to all fragments for the final CDIL data stream. The final CDIL stream is constructed by pooling all fragments across all domains and applying a random permutation π:

$$\mathcal{S} = \text{Permute}\left(\bigcup_{v \in \mathcal{V}_{orig}} \mathbb{F}^v\right) = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T\}\ , \tag{11}$$

where each task $\mathcal{T}_t$ corresponds to a specific fragment $S_{\pi(t)}$. This stochastic assembly ensures three key properties:

- Unlike CIL's uniform increments and consistent domain, our stream features varying class counts and alternating domains. This irregularity creates structural fragmentation, presenting tasks as heterogeneous shards rather than consistent blocks.
- Unlike DIL's fixed semantic space, classes recur across disparate domains over time. This compels the model to synthesize global concepts from partial views.
- Unlike the staged domain progression adopted in CIDA, our stream stochastically alternates between new classes and domains. As a result, the adaptation trajectory becomes temporally incoherent.

B Theoretical Analysis of Prototype Drift

This section provides a theoretical analysis of the proposed *Prototype Drift Path*. We first model the feature space transformation and derive the error bound for the drift estimation. Furthermore, we demonstrate the theoretical superiority of our multi-center compensation strategy over single-center baselines. Finally, we analyze the robustness of our method under varying memory buffer sizes.

B.1 Mathematical Formulation of Feature Space Transformation

When the feature extractor evolves from ϕ_{t-1} to ϕ_t, the feature space undergoes a transformation $\mathcal{T} : \phi_{t-1}(\mathcal{X}) \to \phi_t(\mathcal{X})$. For any sample $\boldsymbol{x}$, this transformation can be modeled as a displacement field:

$$\phi_t(\boldsymbol{x}) = \phi_{t-1}(\boldsymbol{x}) + \boldsymbol{\delta}(\boldsymbol{x})\ , \tag{12}$$

where $\boldsymbol{\delta}(\boldsymbol{x})$ represents the point-wise drift in the trainable feature branch. Since the frozen branch remains unchanged, the ideal update for a historical concatenated prototype $\boldsymbol{p}^v_{c,t-1}$ is

$$\boldsymbol{p}^v_{c,t} = \boldsymbol{p}^v_{c,t-1} + \left[\mathbf{0},\, \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}^v_c}[\boldsymbol{\delta}(\boldsymbol{x})]\right]\ . \tag{13}$$

This population update describes how the corresponding centroid changes as the trainable feature branch evolves. In practice, the expected drift is approximated using stored exemplars.

B.2 Error Bound via Lipschitz Continuity

We analyze the reliability of estimating the population drift $\mathbb{E}[\boldsymbol{\delta}(\boldsymbol{x})]$ using a finite memory buffer $\mathcal{M}^v_c$ of size n.

Assumption 1 (Lipschitz Continuity of Drift)**.** *The drift function $\boldsymbol{\delta}(\cdot)$ is L-Lipschitz continuous, i.e., any $\boldsymbol{x}_1, \boldsymbol{x}_2$:*

$$\|\boldsymbol{\delta}(\boldsymbol{x}_1) - \boldsymbol{\delta}(\boldsymbol{x}_2)\|_2 \leq L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2\ . \tag{14}$$

Justification: In incremental learning, ϕ_t differs from ϕ_{t-1} by a small weight update $\Delta\theta$. For neural networks with bounded activation derivatives (e.g., ReLU, Tanh), the output change is Lipschitz continuous with respect to the input [48, 49].

Definition 1 (View-Specific Variance)**.** *Let $\boldsymbol{\mu}^v_c$ be the centroid of class c in domain view v. The view-specific variance is bounded by $\sigma^2_{c,v}$:*

$$\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}^v_c}[\|\boldsymbol{x} - \boldsymbol{\mu}^v_c\|^2_2] \leq \sigma^2_{c,v}\ . \tag{15}$$

Proposition 1 (Drift Estimation Error)**.** *Assume that the drift function is fixed and L-Lipschitz continuous, and that n samples are drawn independently from $\mathcal{D}_c^v$ and independently of the drift function. Then the sample-mean drift estimator satisfies:*

$$\mathbb{E}\left[\|\hat{\boldsymbol{\delta}}_{c,t}^v - \boldsymbol{\delta}_{c,t}^v\|_2^2\right] \leq \frac{L^2\sigma_{c,v}^2}{n} . \tag{16}$$

Proof. Let $\boldsymbol{Z} = \boldsymbol{\delta}(\boldsymbol{x})$ denote the random variable representing the drift. We seek to bound the variance of the sample mean estimator, given by $\text{Var}(\hat{\boldsymbol{\delta}}_{c,t}^v) = \frac{1}{n}\text{Var}(\boldsymbol{Z})$. Recall that the variance of a random variable is bounded by its second moment centered at any constant vector $\boldsymbol{c}$, i.e., $\text{Var}(\boldsymbol{Z}) \leq \mathbb{E}[\|\boldsymbol{Z} - \boldsymbol{c}\|_2^2]$. By choosing $\boldsymbol{c} = \boldsymbol{\delta}(\boldsymbol{\mu}_c^v)$ and invoking the L-Lipschitz continuity as Assumption 1, we obtain:

$$\|\boldsymbol{\delta}(\boldsymbol{x}) - \boldsymbol{\delta}(\boldsymbol{\mu}_c^v)\|_2^2 \leq L^2\|\boldsymbol{x} - \boldsymbol{\mu}_c^v\|_2^2 . \tag{17}$$

Taking the expectation over $\boldsymbol{x} \sim \mathcal{D}_c^v$ and applying the View-Specific Variance in Definition 1 yields an upper bound on the variance:

$$\text{Var}(\boldsymbol{Z}) \leq L^2\mathbb{E}[\|\boldsymbol{x} - \boldsymbol{\mu}_c^v\|_2^2] \leq L^2\sigma_{c,v}^2 . \tag{18}$$

Consequently, the estimation error is bounded by $\frac{1}{n}\text{Var}(\boldsymbol{Z}) \leq \frac{L^2\sigma_{c,v}^2}{n}$, which completes the proof. □

This result characterizes an idealized independent-sampling setting. In EMCift, exemplars are selected by herding and are also used in model training.

B.3 Theoretical Implications

The derived bound in Proposition 1 reveals two critical factors governing the stability of incremental learning: the variance structure (σ^2) and the sample size (n). In this section, we will analyze the superiority of our multi-center design and the robustness of our method. To facilitate comparison, we denote the expected intra-domain variance as $\sigma_{intra}^2 := \mathbb{E}_{v\in\mathcal{V}_c}[\sigma_{c,v}^2]$.

B.3.1 Variance Decomposition

This theoretical bound explicitly justifies our extensible multi-center design over traditional single-center methods. Consider a *Single-Center* baseline where each class c aggregates data from all seen domains $v \in \mathcal{V}_c$ into a single representation. The total variance σ_{global}^2 of this merged distribution can be decomposed as:

$$\sigma_{global}^2 \approx \underbrace{\mathbb{E}_{v\in\mathcal{V}_c}[\sigma_{c,v}^2]}_{\sigma_{intra}^2} + \underbrace{\text{Var}_{v\in\mathcal{V}_c}(\boldsymbol{\mu}_c^v)}_{\sigma_{inter}^2} , \tag{19}$$

where σ_{inter}^2 represents the variance between domain-specific centroids. In Class-Domain Incremental Learning (CDIL), significant domain shifts imply $\sigma_{inter}^2 \gg \sigma_{intra}^2$. This structural decomposition highlights that single-center methods inherently suffer from inflated variance due to domain heterogeneity.

B.3.2 Superiority of Prototype Drift Strategy

Building on the variance decomposition, we demonstrate the precision advantage of our drift strategy. Applying Proposition 1, the expected drift estimation error for a single-center prototype ($\mathcal{E}_{single}$) versus our EMCift ($\mathcal{E}_{ours}$) satisfies:

$$\begin{aligned} \mathcal{E}_{single} &\leq \frac{L^2(\sigma_{intra}^2 + \sigma_{inter}^2)}{n} \quad , \\ \mathcal{E}_{ours} &\leq \frac{L^2\sigma_{intra}^2}{n} . \end{aligned} \tag{20}$$

The term σ_{inter}^2 increases the error bound for the single-center estimator due to domain shifts. In contrast, our leaf-wise update depends solely on σ_{intra}^2 by treating domains separately. For the same sample size n, this decomposition yields a smaller upper bound for view-specific drift estimation when $\sigma_{inter}^2 > 0$. This result suggests that separating domain views can reduce the effect of between-view variation on drift estimation.

B.3.3 Robustness under Memory Constraints

The bound also elucidates the impact of memory size n, particularly in extreme cases.

Effectiveness at $n = 1$. A single-item memory buffer can still provide a stable drift proxy when the exemplar is selected by herding. With $n = 1$, herding selects an exemplar close to the mean of the available candidates, providing a representative point for estimating the leaf-wise drift. By Lipschitz continuity, $\|\boldsymbol{\delta}(\boldsymbol{x}_h) - \boldsymbol{\delta}(\boldsymbol{\mu}_c^v)\|_2 \le L\|\boldsymbol{x}_h - \boldsymbol{\mu}_c^v\|_2$, so the one-sample drift deviates from a centroid-level drift by at most a bounded intra-leaf term. Therefore, even at $n = 1$, the compensation provides an approximate correction for historical prototype drift rather than leaving a persistent temporal misalignment.

Benefit of Larger n. Increasing the memory buffer size n supplies more replay points for both rehearsal and drift estimation; in our setting, the primary benefit is more accurate drift-based prototype compensation. As shown in Proposition 1, the drift estimation error bound decreases with the number of samples, motivating the use of more exemplars to capture within-leaf variation and improve prototype compensation. This is particularly important because $\boldsymbol{\delta}(\boldsymbol{x})$ can vary within the same leaf and the Lipschitz assumption only guarantees smooth within-leaf changes: with $n = 1$, $\hat{\boldsymbol{\delta}}_{c,t}^v$ may reflect a single local drift, whereas larger n averages out such heterogeneity and better approximates $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_c^v}[\boldsymbol{\delta}(\boldsymbol{x})]$. As a result, structural fidelity improves steadily, while downstream accuracy gains typically exhibit sub-linear saturation as the estimator converges toward the population mean.

C Theoretical Analysis of Model Drift Path

This section establishes the theoretical foundation for the *Model Drift Path*. Grounding our analysis in the domain adaptation generalization bound, we show that multi-center modeling relaxes the joint optimality term, while unconditional alignment may restrict the hypothesis space and worsen this term. This theoretically supports our conditional novelty gate.

C.1 Streaming View of the Domain-Adaptation Bound

At phase t, data arrive from the current view v_t with distribution $\mathcal{D}^{v_t}$. Let $\mathcal{V}_H^{t-1}$ denote the domain views observed before phase t, and let $\mathcal{V}_H^t = \mathcal{V}_H^{t-1} \cup \{v_t\}$ be the updated view set after incorporating the current phase. We define the accumulated mixture distribution as

$$\mathcal{D}_H^t = \sum_{v\in\mathcal{V}_H^t} \pi_v^t \, \mathcal{D}^v, \qquad \sum_{v\in\mathcal{V}_H^t} \pi_v^t = 1,\ \pi_v^t \ge 0, \tag{21}$$

where π_v^t denotes the mixture weight of view v.

Inspired by the domain-adaptation error bound [39], for any hypothesis class $\mathcal{H}$ and any $h \in \mathcal{H}$, the error on the accumulated mixture can be written as:

$$\epsilon_{\mathcal{D}_H^t}(h) \le \epsilon_{\mathcal{D}^{v_t}}(h) + \tfrac{1}{2}\, d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{v_t}, \mathcal{D}_H^t) + \lambda_t(\mathcal{H}), \tag{22}$$

where $\epsilon_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \ne y]$, and

$$\lambda_t(\mathcal{H}) = \inf_{h\in\mathcal{H}} \Big(\epsilon_{\mathcal{D}^{v_t}}(h) + \epsilon_{\mathcal{D}_H^t}(h)\Big) \tag{23}$$

is the joint optimality term.

C.2 Why Over-Alignment Can Hurt for Seen Views

Let $\mathcal{G}$ be a set of admissible representations $g : \mathcal{X} \to \mathbb{R}^d$ and let $\mathcal{T}$ be a fixed classifier-head family. Define the induced hypothesis class:

$$\mathcal{H}(\mathcal{G}) = \big\{\, h_{g,\tau} :\ x \mapsto \tau(g(x)) \;\big|\; g \in \mathcal{G},\ \tau \in \mathcal{T} \,\big\}. \tag{24}$$

Let $\mathcal{G}_{\text{inv}} \subseteq \mathcal{G}$ denote representations satisfying an extra alignment/invariance constraint.

Proposition 2 (Over-Alignment Impairs Joint Optimality)**.** *For any $\mathcal{G}_{\text{inv}} \subseteq \mathcal{G}$, the joint optimality term satisfies:*

$$\lambda_t\big(\mathcal{H}(\mathcal{G}_{\text{inv}})\big) \ \ge\ \lambda_t\big(\mathcal{H}(\mathcal{G})\big). \tag{25}$$

Proof. Recall that $\lambda_t(\mathcal{H}) = \inf_{h\in\mathcal{H}}(\epsilon_{\mathcal{D}^{v_t}}(h) + \epsilon_{\mathcal{D}^t_H}(h))$. Since $\mathcal{G}_{\text{inv}} \subseteq \mathcal{G}$, we have $\mathcal{H}(\mathcal{G}_{\text{inv}}) \subseteq \mathcal{H}(\mathcal{G})$. By the property that the infimum over a set is upper-bounded by the infimum over any subset, minimizing over the restricted space $\mathcal{H}(\mathcal{G}_{\text{inv}})$ yields a value no smaller than minimizing over $\mathcal{H}(\mathcal{G})$. □

Implication. This proposition highlights the risk of unconditional domain alignment. In the CDIL setting, if $v_t \in \mathcal{V}_H^{t-1}$, the incoming view has already been covered by the historical mixture. Enforcing strict invariance ($\mathcal{G}_{\text{inv}}$) then arbitrarily constrains the capacity of the feature extractor. This may inflate the approximation error λ_t, preventing the model from capturing fine-grained intra-class variations and potentially leading to semantic confusion. Our novelty gate g_t avoids this pitfall by applying alignment only when $v_t \notin \mathcal{V}_H^{t-1}$.

C.3 Multi-Center Modeling Relaxes the Joint Optimality Term

Our multi-center design has two aspects in the main paper: a multi-prototype head and a multi-center-motivated training objective (L_{wce} and L_{inter}). Here we give a safe formal statement for the head, which directly relaxes λ_t.

Let $\hat{\boldsymbol{f}}(x) = \boldsymbol{f}(x)/\|\boldsymbol{f}(x)\|_2$ with $\boldsymbol{f}(x) = [\phi^*(x), \phi(x)]$. Define the prototype-induced families:

Single-center (SC).

$$\mathcal{H}_{sc} = \Big\{h(x) = \arg\max_c \langle \hat{\boldsymbol{p}}_c, \hat{\boldsymbol{f}}(x)\rangle\Big\}. \tag{26}$$

Multi-center (MC).

$$\mathcal{H}_{mc} = \Big\{h(x) = \arg\max_c \max_{k\in[K_c]} \langle \hat{\boldsymbol{p}}_{c,k}, \hat{\boldsymbol{f}}(x)\rangle\Big\}, \tag{27}$$

where $K_c \geq 1$ and ties are broken by a fixed rule.

Lemma 1 (Containment and relaxation)**.** $\mathcal{H}_{sc} \subseteq \mathcal{H}_{mc}$. *Consequently,*

$$\lambda_t(\mathcal{H}_{mc}) \leq \lambda_t(\mathcal{H}_{sc}). \tag{28}$$

Proof. Any SC classifier is recovered by setting $\hat{\boldsymbol{p}}_{c,k} \equiv \hat{\boldsymbol{p}}_c$ for all $k \in [K_c]$. Taking an infimum over a superset cannot be larger. □

Connection to the training objective. L_{inter} encourages inter-class separation in evolving feature space, while L_{wce} mitigates class–domain imbalance. Together with the relaxed λ_t in Lemma 1, this supports robust adaptation without perpetual alignment, matching our novelty-gated *Model Drift Path*.

D Training and Tree Update.

At the first phase, the pretrained backbone ϕ^* is frozen, while the FFN adapters ϕ and classifier are optimized using the classification objective. We then construct the two-branch representation $\boldsymbol{f} = [\phi^*(\boldsymbol{x}), \phi(\boldsymbol{x})]$ and initialize the prototype tree from an empty root by computing class-domain centers according to Eq. (1). Replay exemplars are selected by herding. Since no historical representation exists, prototype-drift compensation is unnecessary at this stage.

At each subsequent phase, the classifier is expanded for newly observed classes, and the adapters and classifier are optimized on current samples and replay exemplars using $\mathcal{L}_{\text{wce}}$ in Eq. (5), $\mathcal{L}_{\text{inter}}$ in Eq. (4), and the novelty-gated adversarial objective in Eq. (10).

After optimization, we update the prototype tree using features extracted by the updated model. Specifically, historical leaves are updated by applying their estimated feature displacements to the corresponding prototype centers, compensating for representation drift while preserving the accumulated class-domain structure. For incoming observations, a new class node is created if the class has not been previously observed. Otherwise, a new class-domain leaf is added under the corresponding class node when a previously unseen domain is encountered. Each new node or leaf is initialized using current feature statistics according to Eq. (1). Finally, the exemplar buffer is refreshed by herding to retain representative samples for subsequent replay and prototype-drift

Row	Multi	Rescale	Inter	P-Drift	M-Drift	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
1						67.08	77.77	21.79
2	✓					75.98	70.37	29.84
3	✓	✓				77.54	76.69	20.64
4	✓		✓			76.13	74.18	23.00
5	✓			✓		81.37	81.67	15.70
6	✓				✓	76.52	79.06	21.38
7	✓	✓	✓			82.77	83.30	13.28
8	✓			✓	✓	86.39	84.50	12.71
9	✓		✓	✓	✓	84.59	83.56	13.41
10	✓	✓		✓	✓	81.97	80.50	15.46
11	✓	✓	†	✓	✓	67.63	74.50	25.96
12	✓	✓	✓		✓	84.21	83.14	15.23
13	✓	✓	✓	✓		83.76	80.18	14.09
14	✓	✓	✓	✓	†	81.33	81.26	15.77
15	✓	✓	✓	✓	✓	**88.51**	**86.47**	**10.00**

Table 5: Comprehensive ablation study on iDigits ($T = 19$). **Multi**: Multi-Center Extension; **Rescale**: Class Balancing; **Inter**: Inter-View Contrastive Loss; **P-Drift**: Prototype Drift Path; **M-Drift**: Model Drift Path. **Critical Comparisons**: Row 11 (†) replaces Inter-View Loss with traditional triplet loss. Row 14 (†) applies Continuous Adversarial Training instead of Conditional.

estimation. These operations update the tree structure and prototype centers directly from feature statistics without requiring gradient-based optimization or backpropagation through the prototype tree.

E Computational Complexity and Scalability Analysis

To address the concerns regarding the scalability of the extensible multi-center prototype tree, we provide a formal analysis of its computational and storage complexity.

Storage Complexity. Each node in the prototype tree represents a d-dimensional feature vector. For a CDIL scenario with N classes and an average of V domains per class, the total number of leaf nodes is $M = \sum_{c=1}^{N} |\mathcal{V}_c|$. The storage requirement for the entire tree is $O(M \cdot d)$. In a large-scale setting like DomainNet ($T = 167$), even if we accumulate $M = 2,000$ leaf nodes with a ViT-B/16 backbone ($d = 768$), the memory consumption is approximately:

$$2,000 \times 768 \times 4 \text{ Bytes} \approx 6.14 \text{ MB} . \tag{29}$$

Compared to the backbone parameters ($\sim$ 300 MB) or the memory required by exemplar-based methods to store raw images, this overhead is negligible.

Computational Complexity.

- **Inference:** For a given test sample, classification involves computing the inner product between the feature vector $\hat{\boldsymbol{f}}$ and all M normalized leaf nodes. The complexity is $O(M \cdot d)$. Since inner product operations are highly optimized on modern GPUs, the increase in latency as the tree grows is marginal.
- **Update:** During each incremental phase, historical leaf nodes are updated following the *Prototype Drift Path* detailed in the main text. This process entails only simple vector additions per node, incurring $O(M \cdot d)$ complexity once per phase.

Scalability Justification. Unlike methods that grow the number of trainable parameters (e.g., adding new expert heads or adapters) which might lead to GPU memory exhaustion, EMCift only grows the prototype memory. The prototype-scoring cost grows linearly with M_t, providing predictable scaling as new class-domain pairs are observed.

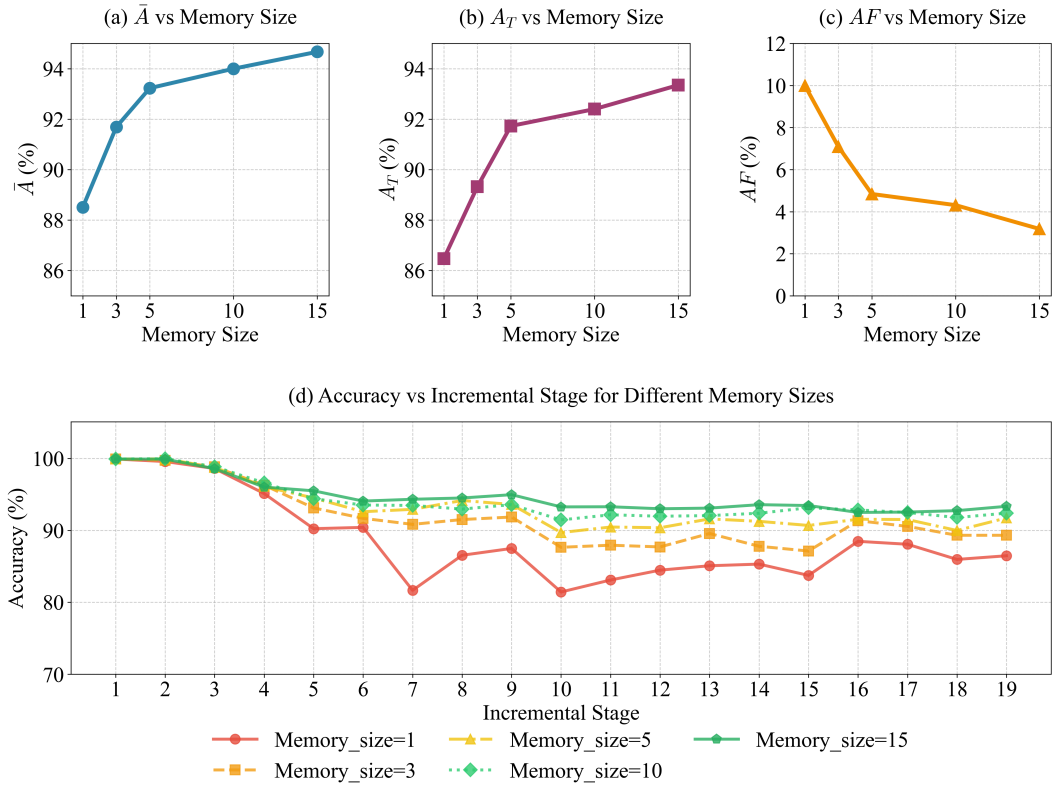


Figure 6: **Memory Analysis.** We conducted experimental verification on the iDigits dataset using multiple memory sizes. Consistent with our theoretical analysis (Sec. B.3), increasing memory size (n) reduces drift estimation error, thereby improving model performance.

F Extended Memory Analysis

To empirically substantiate the theoretical link between memory size n and drift-estimation quality derived in Sec. B, we perform a sensitivity study on iDigits by varying the per-class-domain memory budget $|\mathcal{M}_c^v|$ from 1 to 15 (Fig. 6). Notably, the memory buffer supports both replay-based training and the estimation of the leaf-wise drift $\hat{\boldsymbol{\delta}}_{c,t}^v$. Here we focus on how increasing n strengthens drift-driven prototype compensation.

Efficiency at the Limit ($n = 1$). Our main experiments report performance using the minimal setting ($n = 1$), where our method already outperforms the compared baselines. These results demonstrate the practical effectiveness of EMCift with a single exemplar per class-domain pair.

Scalability and Growth Space. Crucially, Fig. 6 reveals substantial headroom for improvement. As memory size increases, we observe a continuous rise in average accuracy ($\bar{A}$) and a sharp decline in forgetting (AF). This trend demonstrates that while our method is highly data-efficient, it is also scalable. By leveraging additional memory resources, it can further minimize estimation variance and enhance stability.

G Detailed Ablation Analysis

In this section, we provide a granular analysis of every component in EMCift using the comprehensive ablation results reported in Tab. 5.

G.1 Foundational Components Analysis

Tab. 5 first verifies that a multi-center prototype structure is the prerequisite for learning from heterogeneous streams. Compared with the single-center baseline (Row 1), the *Multi-Center* base (Row 2) improves $\bar{A}$ from 67.08% to 75.98%, indicating that explicitly modeling multiple centers is essential for capturing cross-domain variations.

We then examine the two optimization objectives on top of the Multi-Center base. Adding class balancing (Rescale, Row 3) yields a clear gain ($75.98\% \rightarrow 77.54\%$), while introducing Inter-View Loss alone (Inter, Row 4) provides a smaller but consistent improvement ($75.98\% \rightarrow 76.13\%$). Notably, combining them (Row 7) brings $\bar{A}$ to 82.77%, substantially beyond either individual addition. This suggests that *Inter-View Loss* benefits from the more balanced gradients induced by *Rescale*, leading to a better-organized global semantic space.

G.2 Decoupling the Dual-Path Mechanism

We further decouple the dual-path design by removing each drift path from the full model (Row 15, $\bar{A} = 88.51\%$). Removing the *Prototype Drift Path* (P-Drift, Row 12) leads to a degradation ($88.51\% \rightarrow 84.21\%$) and increases forgetting (AF rises to 15.23%), implying that prototype-level compensation is important for maintaining alignment between historical prototypes and the evolving feature space. In contrast, removing the *Model Drift Path* (M-Drift, Row 13) reduces accuracy from 88.51% to 83.76%, with AF increasing from 10.00% to 14.09%. This suggests that M-Drift contributes to both final performance and retention, complementing the prototype-level compensation of P-Drift.

G.3 Critical Design Validations

We finally compare EMCift against two common alternatives. First, replacing *Inter-View Loss* with the standard triplet loss (Row 11) severely harms performance ($\bar{A} = 67.63\%$), even below the *Multi-Center* base (Row 2). This indicates that instance-level triplet constraints are insufficient to preserve the global structure required by a growing prototype hierarchy, whereas *Inter-View Loss* explicitly encourages cross-domain class separation at a manifold level. Second, using continuous adversarial training (Row 14) underperforms our conditional strategy (Row 15) ($81.33\% \rightarrow 88.51\%$). A persistent alignment constraint can suppress domain-specific cues that are necessary for fine-grained classification, while conditioning the regularization avoids over-constraining representation learning.

H Additional Visualizations and Analysis

We present t-SNE visualizations for all compared methods in Fig. 7 to analyze the evolution of the feature space. None of the methods were able to cluster a category across domains into a single cluster, which once again confirmed the endogeneity of multi-centered categories in the CDIL scenario.

Semantic Interference in Regularization Methods. Methods relying on distillation such as AFC, LwF, SDC and iCaRL suffer from significant semantic overlap. Specifically, AFC exhibits a constrained topology where clusters appear radially distributed as seen in Fig. 7. This pattern stems from a strict alignment strategy where the model penalizes feature deviations to ensure stability. Such a rigid approach collapses the natural intra-class variance required to represent diverse domains. The model prioritizes a fixed geometric structure over a flexible semantic manifold, which consequently limits generalization on unseen domains.

Domain Dominance in Prompt Methods. Prompt-based approaches such as DualPrompt, L2P, and ICON demonstrate a pattern of domain dominance. Their visualizations show features split into large disjoint clusters based on domain origins instead of unified semantic categories. This distribution suggests that domain-specific styles overwhelm the underlying semantic identity. Such representations make it difficult for the model to maintain a coherent class structure across the different domains of the iDigits dataset. Additionally, due to the static feature space in whole learning space, SimpleCIL and APER Adapter show a relatively low adaptability to complex scenarios as a whole, with features in more than one domain mixed together.

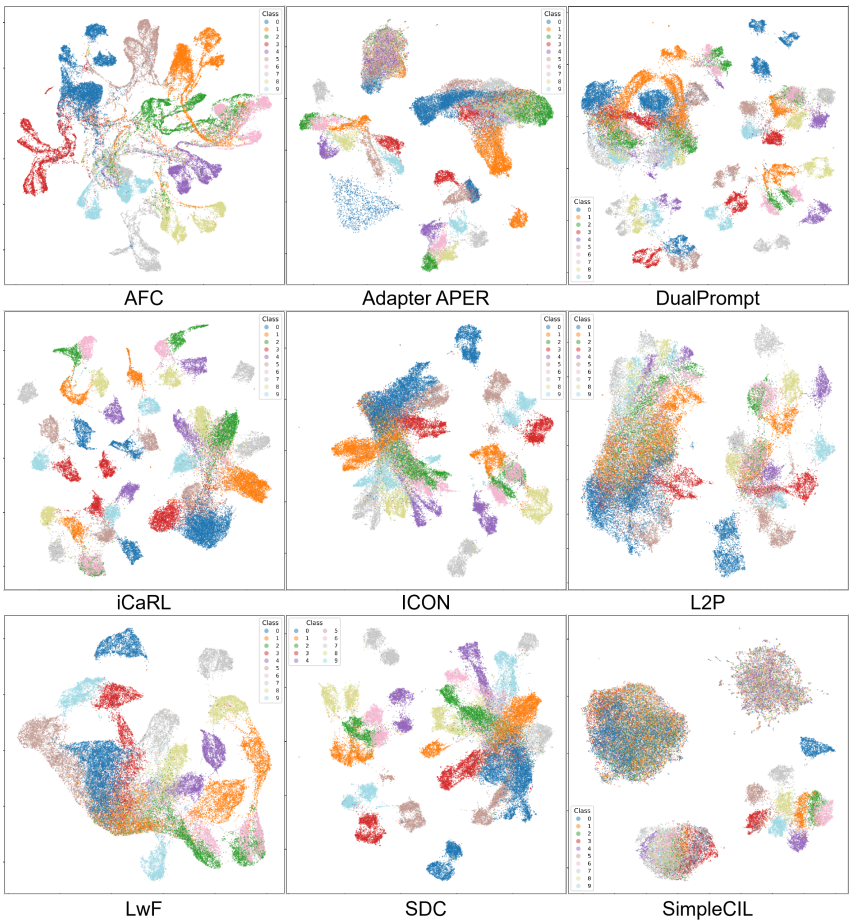


Figure 7: Full t-SNE visualizations of feature representations for all compared methods on the iDigits dataset.

I Additional Experimental Results

I.1 Pair-as-Class Adaptation of Standard Baselines

To further evaluate EMCift under CDIL, we adapt nine standard continual learning baselines using a Pair-as-Class formulation, as shown in Tab. 6. Each observed class-domain pair (c, v) is assigned an independent output label during training, and predicted pair labels are mapped back to their corresponding semantic classes for evaluation. All methods use the same incremental streams, ViT-B/16 backbone, exemplar budget, and evaluation protocol as the corresponding CDIL experiments.

EMCift achieves the highest $\bar{A}$ and A_T among all evaluated methods on all three datasets, with particularly large margins on iDigits and CORe50. Compared with the strongest Pair-as-Class baseline for each metric, EMCift improves $\bar{A}$ and A_T by 14.45 and 19.35 percentage points on iDigits, 1.24 and 0.03 points on DomainNet, and 9.13 and 8.33 points on CORe50, respectively. These results demonstrate the consistent advantage of EMCift across datasets, although the final-accuracy difference on DomainNet is marginal.

The effects of Pair-as-Class vary across methods and datasets. For methods involving continuous parameter optimization, assigning independent labels to different domain views within the same semantic category may introduce competing learning objectives and lead to fragmentation of shared category knowledge. In contrast, prototype-based methods (such as APER and SimpleCIL) gain substantial performance improvements under this formulation, indicating that separate pairing-level prototypes can capture certain domain-specific variations. However, representing class-domain pairs independently does not explicitly provide EMCift's shared semantic hierarchy, inter-view

Table 6: Pair-as-Class adaptation of nine standard continual learning baselines on iDigits, DomainNet, and CORe50, with EMCift as a reference. Results are mean±std over three runs. $\bar{A}$, A_T, and AF denote average incremental accuracy, final accuracy, and average forgetting, respectively. Bold indicates the best mean in each column.

Method	iDigits			DomainNet			CORe50		
	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
LwF	33.35 ± 1.93	24.12 ± 1.57	73.30 ± 3.26	2.68 ± 0.22	2.77 ± 0.23	48.65 ± 4.78	10.61 ± 1.34	3.33 ± 2.30	43.09 ± 7.88
SDC	41.69 ± 1.04	13.46 ± 0.76	87.86 ± 1.00	21.32 ± 0.11	8.29 ± 0.17	69.65 ± 0.18	26.95 ± 0.19	9.86 ± 0.15	58.72 ± 1.30
AFC	48.66 ± 2.18	29.01 ± 6.95	63.69 ± 3.09	17.67 ± 0.25	8.54 ± 1.03	48.48 ± 1.49	66.16 ± 1.47	69.07 ± 2.16	11.29 ± 1.10
iCaRL	64.53 ± 1.37	58.77 ± 2.52	42.55 ± 1.31	42.54 ± 0.23	31.13 ± 0.17	53.68 ± 0.86	25.37 ± 0.22	10.62 ± 0.16	47.15 ± 1.00
APER	73.20 ± 1.37	63.09 ± 1.23	10.63 ± 0.22	61.27 ± 0.27	53.98 ± 0.28	$\mathbf{6.03 \pm 0.09}$	81.00 ± 1.09	83.20 ± 0.91	1.07 ± 0.03
SimpleCIL	63.17 ± 0.00	50.29 ± 0.00	11.71 ± 0.00	64.98 ± 0.00	57.58 ± 0.00	6.41 ± 0.00	79.05 ± 0.00	83.86 ± 0.00	$\mathbf{0.74 \pm 0.00}$
L2P	62.35 ± 0.64	50.77 ± 2.69	23.46 ± 1.16	60.20 ± 0.07	53.95 ± 0.32	8.86 ± 0.08	76.43 ± 0.07	77.97 ± 0.22	4.38 ± 0.30
DualPrompt	69.40 ± 0.25	53.09 ± 4.32	24.78 ± 3.32	65.30 ± 0.19	58.67 ± 0.17	7.71 ± 0.11	80.31 ± 1.04	80.59 ± 0.47	3.84 ± 0.72
ICON	72.92 ± 0.36	66.50 ± 3.03	13.80 ± 1.60	64.10 ± 0.28	57.93 ± 0.19	8.41 ± 0.11	81.56 ± 0.27	82.31 ± 0.80	4.03 ± 0.49
EMCift (Ours)	$\mathbf{87.65 \pm 0.76}$	$\mathbf{85.85 \pm 0.56}$	$\mathbf{10.45 \pm 0.52}$	$\mathbf{66.54 \pm 0.21}$	$\mathbf{58.70 \pm 0.29}$	17.54 ± 0.15	$\mathbf{90.69 \pm 0.27}$	$\mathbf{92.19 \pm 0.37}$	2.27 ± 0.28

Table 7: Sensitivity to training-time domain information on iDigits. All results are mean±std over three runs. No task or domain identity is required at inference.

Setting	Alignment	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
True domain ID	Conditional	**87.65±0.76**	**85.85±0.56**	**10.45±0.52**
Task-as-View	Off	83.67±0.93	83.81±1.61	13.10±1.19
Task-as-View	Always on	67.26±12.40	55.08±21.40	32.33±12.49
Domain ID + 5% noise	Conditional	82.36±1.39	80.53±3.43	16.80±2.28
Domain ID + 10% noise	Conditional	83.39±0.88	83.05±1.36	13.64±1.42

coordination, or dual-path drift compensation. Overall, these results indicate that Pair-as-Class can partially capture the advantages of multi-center representations, while the structured design of EMCift provides a more effective solution for CDIL on multiple datasets evaluated.

I.2 Training-Time Domain Information

EMCift uses acquisition-domain identity only as training-time metadata for view-indexed operations, while the learned model does not require task or domain identity at inference. We first consider a *Task-as-View* variant that replaces the true domain index v with the observable phase index t, converting class-domain leaves (c, v) into class-phase leaves (c, t). We additionally evaluate robustness to corrupted training metadata by replacing the domain labels of approximately 5% and 10% of training stages with incorrect observed domains. The results are summarized in Tab. 7.

As shown in Tab. 7, replacing explicit domain metadata with the observable phase index still retains $83.67 \pm 0.93\%$ average accuracy, compared with $87.65 \pm 0.76\%$ under the standard EMCift setting. This indicates that explicit acquisition-domain metadata improves view organization but is not indispensable to the multi-center framework. In contrast, always-on alignment substantially degrades performance because temporal phases do not necessarily correspond to recurring acquisition domains. EMCift also remains effective under moderate corruption of the training-time domain metadata. The non-monotonic difference between 5% and 10% noise reflects experimental variability rather than a monotonic relationship between corruption rate and performance.

I.3 Backbone and Pretraining Robustness

To examine the dependence of EMCift on the pretrained representation, we conduct controlled evaluations on iDigits with alternative backbone initializations. The main experiments use the same ViT-B/16 pretrained on ImageNet-21K for all methods to ensure a fair comparison. Here, we additionally evaluate ViT-B/16 with ImageNet-1K supervised pretraining, DINO self-supervised pretraining, and OpenAI CLIP image-text contrastive pretraining. For these ViT variants, the CDIL stream, memory budget, model architecture, and training and evaluation protocols remain unchanged, with only the pretrained initialization varied. We further replace ViT-B/16 with an ImageNet-21K pretrained ResNet-50 to evaluate sensitivity to the backbone architecture. The results are summarized in Tab. 8.

Table 8: Backbone and pretraining robustness on iDigits. The CDIL stream, memory budget, and training/evaluation protocol are kept unchanged. Results are mean±std over three runs.

Backbone	Pretraining	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
ViT-B/16	ImageNet-21K supervised	**87.65±0.76**	**85.85±0.56**	**10.45±0.52**
ViT-B/16	ImageNet-1K supervised	83.36±1.34	81.57±2.55	14.94±1.89
ViT-B/16	ImageNet-1K DINO	80.08±1.01	77.02±3.75	20.44±2.35
ViT-B/16	OpenAI CLIP	80.13±1.07	83.95±1.49	13.67±1.29
ResNet-50	ImageNet-21K supervised	79.73±0.96	72.60±1.47	19.78±0.69

As shown in Tab. 8, EMCift remains effective across supervised, self-supervised, and image-text contrastive pretraining. With DINO initialization, it achieves $80.08 \pm 1.01\%$ average incremental accuracy and $77.02 \pm 3.75\%$ last-phase accuracy. Under CLIP initialization, it obtains $80.13 \pm 1.07\%$ average incremental accuracy and $83.95 \pm 1.49\%$ last-phase accuracy, together with lower forgetting than the DINO variant. Replacing the ViT backbone with ResNet-50 also yields $79.73 \pm 0.96\%$ average incremental accuracy. Although absolute performance varies with the pretrained representation and backbone architecture, these results show that EMCift is not restricted to the original ViT-B/16 ImageNet-21K supervised initialization.

I.4 Evaluation on PACS Dataset

To simulate real-world data streams with different scales and severe domain discrepancies, we extend our evaluation to the PACS dataset. As reported in Tab. 9, EMCift consistently maintains its superiority. The robust performance indicates that our multi-center design effectively captures unified knowledge from diverse domain shifts, significantly outperforming prior prototype-based and prompt-based methods.

Table 9: Quantitative comparison on the PACS dataset.

Method	$\bar{A}$ (%) ↑	A_T (%) ↑	Method	$\bar{A}$ (%) ↑	A_T (%) ↑
LwF [6]	26.30±0.77	16.21±0.00	AFC [45]	70.87±4.13	68.35±2.85
iCaRL [7]	75.21±1.48	69.80±1.38	SimpleCIL [19]	62.02±0.00	34.37±0.00
SDC [35]	68.33±4.36	42.44±14.08	APER Adapter [19]	61.53±0.99	39.85±0.62
L2P [4]	61.92±0.63	53.54±1.41	DualPrompt [5]	64.72±1.52	53.88±1.10
ICON [12]	59.67±1.11	52.94±0.45	**EMCift (Ours)**	**88.19±1.37**	**85.27±1.36**

I.5 Robustness to Randomized Task Orders

A core challenge of the CDIL scenario is learning from unpredictable streams with unfixed numbers of new classes. To verify that our gains do not rely on a specific task sequence, we additionally evaluate two randomized task orders on iDigits. As shown in Tab. 10, EMCift consistently achieves the highest $\bar{A}$ and A_T under entirely different progression trajectories, demonstrating exceptional robustness against the irregularity of class and domain arrivals.

Table 10: Performance evaluation under different randomized task orders on iDigits.

Method	Order 1			Order 2		
	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
ICON [12]	41.46	40.49	20.02	62.64	58.29	13.06
DualPrompt [5]	65.27	66.67	16.37	66.84	79.36	**12.18**
EMCift (Ours)	**85.46**	**82.16**	**15.29**	**81.34**	**79.40**	19.09

I.6 Compatibility with Standard CIL and DIL Protocols

While CDIL represents a more general and complex stream with simultaneous class and domain evolution, a robust classification model should also excel under established protocols where only one

axis varies. We evaluate EMCift against strong baselines under standard Class Incremental Learning (CIL) and Domain Incremental Learning (DIL) settings. As Tab. 11 illustrates, EMCift sweeps the best performance across all three metrics in DIL and delivers the strongest overall accuracy in CIL. These results confirm that our multi-center design is not exclusively tailored to CDIL, but also seamlessly scales down to provide state-of-the-art backward compatibility for standard continual learning scenarios.

Table 11: Evaluation of EMCift under standard DIL and CIL settings.

Method	DIL			CIL		
	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓	$\bar{A}$ (%) ↑	A_T (%) ↑	AF (%) ↓
DualPrompt [5]	73.99	66.46	21.96	72.71	58.56	20.83
ICON [12]	80.86	73.85	18.40	71.75	57.74	**16.29**
EMCift (Ours)	**83.54**	**83.12**	**11.39**	**77.53**	**62.37**	47.67

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately state the contributions, the proposed EMCift framework, and the scope of the class-domain incremental learning (CDIL) scenario.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our approach in the Limitations paragraph.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The full set of assumptions theoretical bounds, and proofs are formally provided in Appendix B and Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the model architecture, datasets, and experimental protocols in Section 4.1 and Appendix A. The implementation and evaluation code are publicly available at the GitHub repository linked in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publicly available benchmark datasets. The training and evaluation code, together with the CDIL stream construction protocols, is publicly available at the GitHub repository linked in the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We describe the datasets and experimental protocols in Section 4.1 and Appendix A, with implementation configurations and hyperparameter settings provided in the released code.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The quantitative results in Table 1 report both the mean and standard deviation computed over 3 independent runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are conducted on a single NVIDIA V100 GPU, as specified in Section 4.1. Inference efficiency is further evaluated in Section 4.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research fully conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on foundational incremental learning methodology; we believe there is no direct, foreseeable negative societal impact or direct path to malicious applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks, as it does not release high-risk generative models or scraped datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original creators of the benchmarks (iDigits, DomainNet, CORe50) and baseline methods are properly cited and credited in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release any new datasets or assets, it relies on public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology does not involve the use of LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.