Cognitive Bifurcation: Dual-Progressive Causal Diffusion with Hippocampal Memory for Continual Graph Learning

Jiahao Liang
School of Computer Science and Engineering,
South China University of Technology
Guangzhou, China
csjiahliang6@mail.scut.edu.cn

Carl Yang
Department of Computer Science,
Emory University
Atlanta, USA
j.carlyang@emory.edu

Haoran Yang
School of Computer Science and Engineering,
Central South University
Changsha, China
yhr.cse@csu.edu.cn

Zhiwen Yu
School of Computer Science and Engineering,
South China University of Technology
Guangzhou, China
zhwyu@scut.edu.cn

Mengzhu Wang
School of Artificial Intelligence,
Hebei University of Technology
Tianjin, China
dreamkily@gmail.com

Kaixiang Yang*
School of Computer Science and Engineering,
South China University of Technology
Guangzhou, China
yangkx@scut.edu.cn

Abstract

Continual Graph Learning (CGL) on non-stationary streams faces the fundamental challenge of adapting to complex distribution shifts, where the entanglement of invariant causal structures and transient environmental noise inevitably leads to catastrophic forgetting. Under such non-stationary conditions, existing methods relying on incremental updates or raw replay are vulnerable to *recursive error accumulation*: a minor misinterpretation of the shift at an early stage propagates over time, causing a collapse in structural understanding. To tackle this, we draw inspiration from the **cognitive bifurcation** in the human brain and propose **DCDHippo** (**D**ual-Progressive **C**ausal **D**iffusion with **Hippo**campal Memory). This framework treats adaptation as a closed-loop interplay between two systems. First, to handle real-time shifts, a Progressive Causal Masking mechanism (Fast System) dynamically prunes shift-induced noise to extract invariant causal skeletons. Simultaneously, to rectify local drifts, a Causal-Anisotropic Diffusion module (Slow System) internalizes these skeletons into a global invariant representation space via generative reconstruction. Crucially, we introduce the Evolving Hippocampal Memory that re-activates this global knowledge to distill wisdom back into the fast adapter, ensuring robust adaptation to continuous distribution shifts. Extensive experiments demonstrate that DCDHippo significantly outperforms state-of-the-art methods in both adaptability and knowledge retention.

CCS Concepts

• **Information systems** → **Information retrieval**.

Keywords

Data Mining, Continual Graph Learning, Distribution shift

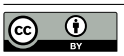
ACM Reference Format:
Jiahao Liang, Carl Yang, Haoran Yang, Zhiwen Yu, Mengzhu Wang, and Kaixiang Yang. 2026. Cognitive Bifurcation: Dual-Progressive Causal Diffusion with Hippocampal Memory for Continual Graph Learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3818129

1 Introduction

As graph data scales up and evolves, continual learning on dynamic graphs faces new and acute challenges. The primary goal of Continual Graph Learning (CGL) [2, 7, 9, 15, 17, 19, 37, 43, 52, 60, 66] is to enable a model to continuously learn from a stream of graph data without catastrophically forgetting previously acquired knowledge. To address this, existing research has primarily operated under three adaptation paradigms [34, 40, 44, 55, 64, 69], each defining a distinct mechanism for carrying historical knowledge forward.

The first paradigm, which can be termed recursive parameter adaptation (Regularization-based), treats learning as a continuous fine-tuning process constrained by regularization. Exemplified by TWP [26], which explicitly minimizes the sensitivity of topological aggregation weights to new tasks, and GraphSAIL [60], which employs knowledge distillation to preserve local and global structural patterns, these approaches recursively update parameters while penalizing deviations from the prior state. Other works, such as RieGrace [43], further extend this by adapting to geometric curvature shifts via Lorentzian distillation. Parallel to this is the paradigm of incremental architecture expansion (Parameter-isolation), where methods dynamically increase model capacity to accommodate new distributions. Instead of overwriting weights, approaches like HPNs [66] isolate new knowledge by instantiating hierarchical prototypes for emerging atomic features , while PI-GNN [64] physically segregates parameters into fixed stable parameter blocks and trainable newly introduced parameter blocks to handle graph evolution.

*indicates the corresponding authors.



KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3818129

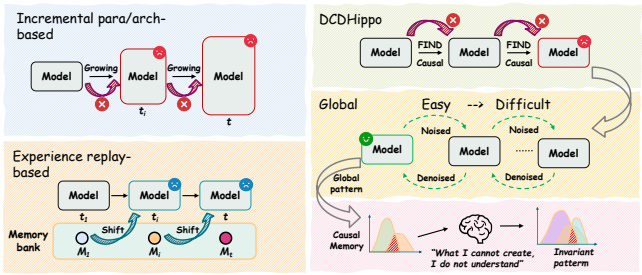


Figure 1: Motivation and paradigm comparison. The three paradigms each have complementary strengths but remain vulnerable to long-horizon drift under non-stationarity; DCDHippo integrates fast causal masking with slow diffusion-based internalization and hippocampal generative feedback to break the "one-step-error, all-step-wrong" cascade.

Similarly, RLC-CN [27] utilizes reinforcement learning to automatically search for optimal layer expansions. The third paradigm relies on raw experience replay. Approaches such as ER-GNN [70] maintain a memory buffer of individual node attributes to circumvent the memory explosion problem , whereas SSM [67] selectively stores sparsified computation subgraphs to retain essential connectivity. More recent methods like CaT and PUMA [29, 30] further advance this by condensing graphs into balanced, information-rich snapshots for efficient replay.

However, as illustrated in Fig. 1, these paradigms remain brittle under non-stationary graph shifts. Regularization/distillation methods preserve past knowledge by constraining updates, but their step-wise adaptation can propagate early mistakes and amplify drift. Architecture-expansion methods alleviate interference via parameter isolation, yet they incur growing complexity and still lack a global invariant anchor. Experience replay revisits historical data, but the buffer often mixes causal signals with context-dependent noise, which can be repeatedly reinforced. Parallel adaptive graph construction and ensemble methods [20, 21] harden representations against structural perturbations, yet they target static settings and offer little protection against long-horizon CGL drift.

Ultimately, non-stationarity exposes the curse of "one-step-error, all-step-wrong": a small misinterpretation at time t can cascade into long-horizon failure. This gap motivates a fundamental question: how can a model break this curse and achieve deep understanding? We find inspiration in Feynman's aphorism, *What I cannot create, I do not understand*. In biological systems, this is addressed by the cognitive bifurcation between the neocortex and the hippocampus [35]. As conceptually summarized in Fig. 1, our design instantiates a fast System 1 (green) for online adaptation and a slow System 2 (yellow, global) that gradually captures global patterns into a generative schema. Crucially, resilience arises from *generative feedback*: the slow system periodically re-activates its memory to rectify and evolve the fast intuition, preventing local errors from collapsing global knowledge [14, 51, 54].

Guided by this intuition, we propose a framework that treats graph continual adaptation as a collaborative journey between fast intuition and slow reflection, namely **DCDHippo** (**D**ual-Progressive **C**ausal **D**iffusion with **Hippo**campal Memory). To the best of our knowledge, this is the first work to bridge causal pruning and generative diffusion within the CGL landscape. Our framework introduces two intertwined innovations: First, to mitigate recursive drift, we design a collaborative architecture consisting of **Progressive Causal Masking (Fast Intuition)** and **Causal-Anisotropic Diffusion (Slow Internalization)**. While the Fast System performs real-time edge pruning to adapt to current shifts, the Slow System operates in the background, modeling the global distribution of causal fragments via a Diffusion process. By reconstructing the underlying invariant structure of all historical causal skeletons, the Slow System internalizes a global invariant pattern that serves as an anchor, effectively rectifying the local errors made by the Fast System during streaming adaptation. Second, to achieve true cognitive evolution, we introduce a standalone feedback module that completes the learning loop. Instead of treating memory as a static repository, **DCDHippo**'s Hippocampal memory actively generates synthetic causal experiences from its internalized diffusion-distilled distribution. These generated samples are used to refine the Fast System through knowledge distillation. This generative feedback mechanism ensures that the model not only remembers the past but also utilizes global insights to sharpen its intuitive responses to future, unseen distribution shifts.

Our contributions are summarized as follows:

- We propose **DCDHippo**, a cognitive-inspired dual-system framework that integrates causal masking and generative diffusion for robust continual graph learning.
- We introduce a progressive causal masking mechanism that explicitly identifies invariant structures, providing a distilled memory buffer that isolates causal regularities from transient noise.
- We design a causality-guided curriculum diffusion strategy that internalizes global knowledge through a generative reconstruction process, effectively balancing stability and plasticity.
- Extensive experiments on diverse benchmarks demonstrate that **DCDHippo**achieves state-of-the-art performance and offers enhanced interpretability through its explicit causal subgraphs.

2 Problem Statement

2.1 The Recursive Drift Phenomenon

Standard forgetting refers to the erasure of weights. However, in graph streaming, we identify a more pernicious failure mode: *Recursive Drift*. Since GNNs rely on message passing, structural noise in G_t propagates through layers. If the model $f_{\theta_{t-1}}$ (trained on past data) has "forgotten" the invariant structure, it may misinterpret the new graph G_t, generating biased embeddings. These biased embeddings then serve as the supervision for θ_t, creating a feedback loop where errors compound over time:

$$\text{Drift}: \quad \mathbb{E}_{G_t}\left[\nabla_\theta \mathcal{L}(f_{\theta_t}(G_t))\right] \neq \mathbb{E}_{G_{0:t}}\left[\nabla_\theta \mathcal{L}_{\text{true}}\right]. \tag{1}$$

To prevent this, the model requires a stable reference mechanism (Phase I) and a robust generative replay (Phase II). We study continual graph learning where a GNN predictor f_θ is trained on a

stream of tasks $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$. At time step t, the learner observes only the current labeled graph dataset $\mathcal{D}_t = (G_t, \mathcal{Y}_t)$, where $G_t = (\mathcal{V}_t, \mathbf{A}_t, \mathbf{X}_t)$ denotes the node set, adjacency matrix, and node features, respectively.

2.2 Continual Graph Learning Setting

We consider a continual learning scenario where a graph neural network (GNN) model f_θ is trained on a sequence of graph tasks $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$. At each time step t, the model receives a new dataset $\mathcal{D}_t = (G_t, \mathcal{Y}_t)$, where $G_t = (\mathcal{V}_t, \mathbf{A}_t, \mathbf{X}_t)$ denotes the graph structure (adjacency matrix $\mathbf{A}_t$ and node features $\mathbf{X}_t$), and $\mathcal{Y}_t$ represents the corresponding labels.

Following the strict continual learning protocol, once step t starts, the learner cannot access raw data from previous tasks $\{\mathcal{D}_0, \ldots, \mathcal{D}_{t-1}\}$. After updating to parameters θ_t, the model should (i) adapt to the current task by minimizing the empirical risk on $\mathcal{D}_t$ and (ii) retain competence on past tasks. Concretely, we seek to minimize the current loss while maintaining performance on all previously seen tasks:

$$\min_{\theta_t} \mathcal{L}(\mathcal{D}_t; \theta_t) \quad \text{s.t.} \quad \mathcal{L}(\mathcal{D}_i; \theta_t) \text{ does not degrade for } i < t. \tag{2}$$

This stability–plasticity trade-off is the core challenge addressed by our method.

3 Methodology

We propose **DCDHippo**, a bio-inspired framework engineered to mitigate the *Recursive Drift Phenomenon* in continual graph learning. DCDHippo orchestrates a synergy between a *Fast Intuitive System* (System 1) namely Progressive Causal Masking for real-time incremental adaptation and a *Slow Internalization Process* (System 2) namely Causal-Anisotropic Diffusion that utilizes diffusion to distill invariant causal laws, which are subsequently stored in the *Evolving Hippocampal Memory* for generative replay.

3.1 Phase I: Progressive Causal Masking (System 1)

To meet the stringent latency demands of online streaming, we forgo heavy generative steps during inference. Instead, we deploy a lightweight **Causal Masker**, formally defined as a parameterized GNN-based scorer $f_S(\cdot; \theta_S)$, which acts as a structural gatekeeper. This module performs incremental learning to identify the causal skeleton of the current graph snapshot G_t, aiming to maintain consistency with the recently acquired knowledge while filtering transient noise.

Reference Retrieval. To constrain the masking process, we require a structural reference adjacency matrix $\mathbf{A}_{\text{ref}}$. At $t = 0$, we use an uninformative prior. For $t > 0$, we retrieve the reference from the historical context to guide the current adaptation:

$$\mathbf{A}_{\text{ref}} = \begin{cases} \mathbf{I}_{|\mathcal{V}|} & \text{if } t = 0, \\ \hat{G}_{t-1} & \text{if } t > 0, \end{cases} \tag{3}$$

where $\mathbf{I}_{|\mathcal{V}|}$ is the identity matrix representing a neutral bias, and $\hat{G}_{t-1}$ denotes the explicit graph structure consolidated in the previous stage. With this structural prior $\mathbf{A}_{\text{ref}}$ establishing a baseline, the immediate challenge for f_S is to distinguish which edges in the new observation G_t deviate from this reference due to noise versus genuine causal shifts.

Associative Masking. The Causal Masker employs a parameterized scorer $f_S(\cdot; \theta_S)$ to evaluate the causal importance of the graph structure. We formally define this scorer using a Graph Neural Network (GNN) that processes the current node features X_t and the historical reference adjacency $\mathbf{A}_{\text{ref}}$:

$$f_S(X_t, \mathbf{A}_{\text{ref}}; \theta_S) = \text{GNN}(X_t, \mathbf{A}_{\text{ref}}), \tag{4}$$

which generates a raw score matrix $\mathbf{S}$. Conditioned on these scores, we formulate the invariant subgraph extraction as a differentiable soft masking operation:

$$\mathbf{M}_{uv} = \sigma(\mathbf{S}_{uv}), \quad \mathbf{A}_t^{\text{inv}} = \mathbf{A}_t \odot \mathbf{M}, \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, $\odot$ denotes the Hadamard product, and $\mathbf{M} \in [0, 1]^{|\mathcal{V}| \times |\mathcal{V}|}$ is the learned causal mask. This formulation allows the GNN to learn an associative mapping between temporal structural changes and causal stability. This differentiable operation preserves gradient flow.

Incremental Consistency Regularization. Since ground-truth causal masks are unavailable in streaming scenarios, we cannot optimize θ_S via direct supervision. Instead, we impose a predictive consistency constraint via the Kullback-Leibler divergence ($\mathbb{D}_{\text{KL}}$). To obtain a global graph representation for classification, we employ a Readout function $\mathcal{R}(\cdot)$ that aggregates the node embeddings. We force the prediction of the classifier P_θ on the current masked graph representation to align with the prediction derived from the reference context:

$$\mathcal{L}_{\text{cons}} = \mathbb{D}_{\text{KL}} \left(P_\theta(Y \mid \mathcal{R}(\mathcal{G}(\mathbf{A}_t^{\text{inv}}))) \parallel P_\theta(Y \mid \mathcal{R}(\mathcal{G}(\mathbf{A}_{\text{ref}}))) \right). \tag{6}$$

where $\mathcal{G}(\cdot)$ constructs the graph from the adjacency matrix and node features. This objective ensures smooth transitions between time steps by penalizing the masker if it preserves edges that cause abrupt shifts in the model's decision logic.

While $\mathcal{L}_{\text{cons}}$ handles short-term consistency, relying solely on this incremental constraint leads to recursive drift, where minor errors in $\mathbf{A}_t^{\text{inv}}$ accumulate over time. To arrest this drift, we must ground the model in a global, invariant distribution—a role fulfilled by the Slow System described next.

3.2 Phase II: Causal-Anisotropic Diffusion (System2)

To rectify the recursive drift inherent in Phase I, we introduce a **Causal-Anisotropic Diffusion** process as the Slow System. This module does not participate in online inference but works in the background to distill a *Global Causal Invariant*.

Causal-Adaptive Noise Injection. We model the global internalization as learning to reverse a diffusion process. Implicitly, this acts as a **Causality-Guided Curriculum**: the model prioritizes learning robust causal structures (which are preserved) before modeling noisy variations (which are destroyed). The noise variance schedule β_k at diffusion step k is modulated by the causal scores $\mathbf{M}_{uv}$ derived in Eq. (5):

$$\beta_k^{(u,v)} = \bar{\beta}_k \cdot (1 - \lambda \cdot \mathbf{M}_{uv}), \tag{7}$$

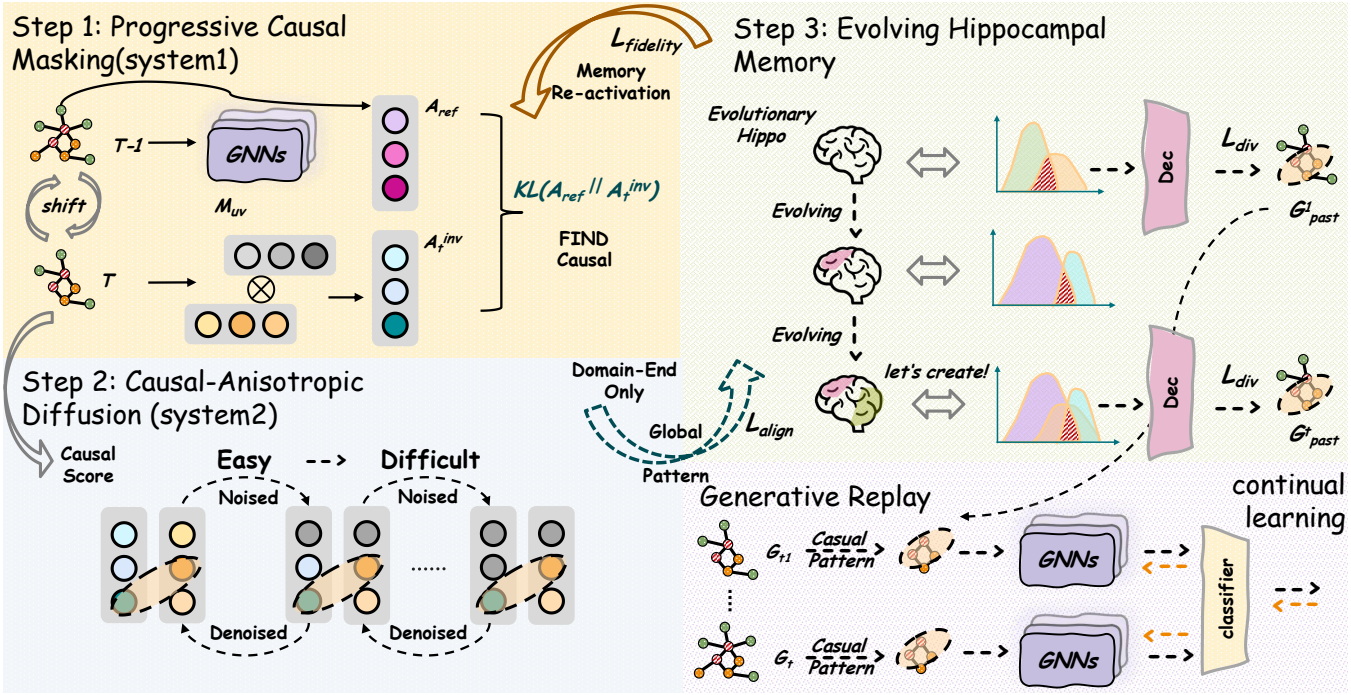


Figure 2: Framework of DCDHippo. A fast system performs Progressive Causal Masking to extract an invariant subgraph for online adaptation, while a slow Causal-Anisotropic Diffusion module internalizes historical causal skeletons into global invariant knowledge. An Evolving Hippocampal Memory provides generative feedback to distill stable knowledge back to the fast system, mitigating recursive drift. The three modules run under a decoupled two-stage schedule: System 1 operates at every streaming step t, whereas System 2 and the Hippocampus are activated once per domain during a consolidation phase.

where $\bar{\beta}_k$ is the standard base noise schedule, and $\lambda \in [0, 1]$ is a hyperparameter controlling the strength of causal protection. The latent adjacency structure $\mathbf{Z}^{(k)}$ transitions via:

$$\mathbf{Z}_{uv}^{(k)} = \sqrt{1-\beta_k^{(u,v)}}\mathbf{Z}_{uv}^{(k-1)} + \sqrt{\beta_k^{(u,v)}}\epsilon, \quad \epsilon \sim \mathcal{N}(0,1). \tag{8}$$

Insight: High-causal structures result in low noise variance and are preserved as anchors. This anisotropic forward process defines the learning curriculum. The subsequent objective is to train a decoder that can reverse this specific, causality-weighted corruption to recover the invariant core.

Global Reconstruction. The diffusion decoder D_ϕ learns to denoise these states to recover the original adjacency. We employ a causal-weighted Mean Squared Error (MSE) objective:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{k,\epsilon,G_t}\left[\sum_{(u,v)} \omega(\mathbf{M}_{uv}) \cdot \left\|\mathbf{A}_t - D_\phi(\mathbf{Z}^{(k)}, k, \mathbf{M})\right\|_F^2\right], \tag{9}$$

where $\omega(\cdot)$ is a weighting function, and $\|\cdot\|_F$ is the Frobenius norm. By minimizing Eq. (9), D_ϕ captures the global distribution of causal structures. Crucially, once optimized, the decoder D_ϕ transcends its role as a mere denoiser. It effectively encapsulates the global causal law, allowing us to repurpose it as a generative engine for the Hippocampal Memory in the final phase.

3.3 Phase III: Evolving Hippocampal Memory

To enable efficient and robust replay, we introduce a dedicated **Hippocampal Memory Module**, parameterized by a lightweight generator H_ψ. Unlike System 2 (Diffusion), which performs heavy iterative denoising to distill global laws, the Hippocampus acts as a rapid-recall engine. Crucially, this memory module is not static; it co-evolves asynchronously with System 1 and System 2 across two distinct phases—a high-frequency assimilation tied to System 1 and a low-frequency consolidation tied to System 2 (detailed in Sec. 3.4).

Asynchronous Knowledge Distillation. The Hippocampus H_ψ maintains a generative memory model of the causal history. Its evolution is driven by a dual-objective: assimilating the current causal snapshot from System 1 while aligning with the global invariant distribution distilled by System 2. We synthesize the replay batch by sampling from this dedicated generator:

$$\mathcal{G}_{\text{past}} = \{G_k \mid G_k = H_\psi(z_k),\ z_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\}_{k=1}^{B}, \tag{10}$$

where H_ψ maps latent noise directly to graph structures. This explicit separation allows System 2 to focus on deep internalization (high-fidelity but slow) while the Hippocampus focuses on agile reproduction (fast and diverse) for replay. During online streaming steps, H_ψ is held frozen and used purely as a sampler producing $\mathcal{G}_{\text{past}}$ to anchor System 1; its deep alignment with D_ϕ is deferred to the periodic consolidation phase.

Hippocampal Evolutionary Objective. To ensure the memory generator H_ψ accurately reflects the evolving global truth, we optimize an evolutionary loss $\mathcal{L}_{\text{evolve}}$ that enforces both fidelity and diversity:

$$\mathcal{L}_{\text{evolve}} = \mathcal{L}_{\text{fidelity}} + \lambda_1 \mathcal{L}_{\text{align}} + \lambda_2 \mathcal{L}_{\text{div}}. \tag{11}$$

The three terms in Eq. (11) are activated at distinct frequencies: $\mathcal{L}_{\text{fidelity}}$ fires at every streaming step, whereas $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{div}}$ are only invoked during the periodic consolidation phase (Sec. 3.4).

Here, $\mathcal{L}_{\text{fidelity}}$ ensures the generator retains the most recent knowledge by minimizing the reconstruction error of the current causal snapshot $G_t \odot M_t$:

$$\mathcal{L}_{\text{fidelity}} = \left\| (G_t \odot M_t) - H_\psi(E_{\text{mem}}(G_t \odot M_t)) \right\|_F^2, \tag{12}$$

where E_{mem} is a lightweight projection head that maps the graph snapshot to the Hippocampus's latent space; it is parameterized independently of the backbone GNN so that memory dynamics are fully decoupled from the downstream task representations. $\|\cdot\|_F$ denotes the Frobenius norm. Simultaneously, $\mathcal{L}_{\text{align}}$ acts as a **Distillation Loss** that forces the lightweight generator's output to match the robust representation learned by the System 2 Diffusion Decoder D_ϕ for the same latent code:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{z\sim\mathcal{N}(0,\mathbf{I})} \left[\left\| H_\psi(z) - D_\phi(z) \right\|_F^2 \right]. \tag{13}$$

Finally, $\mathcal{L}_{\text{div}}$ prevents mode collapse by enforcing feature-space repulsion:

$$\mathcal{L}_{\text{div}} = -\frac{1}{B(B-1)} \sum_{i \neq j} \|E_{\text{mem}}(G_i) - E_{\text{mem}}(G_j)\|_2, \tag{14}$$

where $G_i, G_j \in \mathcal{G}_{\text{past}}$. This ensures the Hippocampus does not simply memorize the latest sample but maintains a diverse population of historical replay samples.

3.4 Generative Replay and Decoupled Optimization

Finally, we close the cognitive loop via a **Generative Replay** mechanism. To prevent catastrophic forgetting while ensuring adaptation to new patterns, the Fast System (Masker and Classifier) must be optimized on both the present reality and the internalized history.

Hybrid Experience Replay. Instead of isolating the learning process to the current graph G_t, which risks overfitting and forgetting, we construct a Hybrid Batch that combines the real current snapshot (G_t, Y_t) with the historical prototypes $\mathcal{G}_{\text{past}}$ synthesized by the Hippocampus. The Fast System is optimized to minimize the downstream task loss (e.g., Cross-Entropy) on this mixed distribution:

$$\mathcal{L}_{\text{replay}} = \mathcal{L}_{\text{task}} + \gamma \mathbb{E}_{(G_k,Y_k)\sim\mathcal{G}_{\text{past}}} \left[\mathcal{L}_{\text{task}}(f_{\text{pred}}(G_k \odot f_S(G_k)), Y_k) \right] \tag{15}$$

Here, the first term drives the Masker to adapt to novel causal shifts in the current stream, while the second term—the replay loss—anchors the Masker to the global causal invariants provided by the Hippocampus. This joint optimization effectively balances the plasticity-stability trade-off.

Decoupled Optimization Protocol. To coordinate the synergy between the Fast System (Sys1), Slow System (Sys2), and Hippocampus (Hippo), we express the total training objective in the following unified form for notational compactness:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{replay}} + \alpha_1 \mathcal{L}_{\text{cons}} + \alpha_2 \mathcal{L}_{\text{diff}} + \alpha_3 \mathcal{L}_{\text{evolve}}. \tag{16}$$

This objective orchestrates a decoupled tripartite cycle: **System 1** minimizes $\mathcal{L}_{\text{replay}}$ to perform the downstream task accurately on both present and past data, regularized by $\mathcal{L}_{\text{cons}}$ for smooth structural adaptation. **System 2** minimizes $\mathcal{L}_{\text{diff}}$ to internalize global invariants via diffusion. Hippocampal Memory minimizes $\mathcal{L}_{\text{evolve}}$ to distill knowledge from both systems into a generative replay model.

In practice, although Eq. (16) is written jointly for mathematical completeness, the optimization strictly follows a two-stage protocol that mirrors the wake–sleep cycle of biological memory consolidation: $\{\mathcal{L}_{\text{replay}}, \mathcal{L}_{\text{cons}}, \mathcal{L}_{\text{fidelity}}\}$ are minimized at every streaming step to handle the current shift, while $\{\mathcal{L}_{\text{diff}}, \mathcal{L}_{\text{align}}, \mathcal{L}_{\text{div}}\}$ are activated only during the memory consolidation stage that runs once after finishing the adaptation on each domain. Because the two groups update disjoint sub-networks (the masking adapter vs. the diffusion decoder and the memory generator), end-to-end gradient conflicts are avoided and optimization remains stable.

Concretely, substituting Eq. (11) into Eq. (16) and regrouping the six resulting terms by their activation phase yields the explicit two-stage decomposition

$$\underbrace{\mathcal{L}_{\text{replay}} + \alpha_1 \mathcal{L}_{\text{cons}} + \alpha_3 \mathcal{L}_{\text{fidelity}}}_{\text{(a) streaming stage: every step } t} + \underbrace{\alpha_2 \mathcal{L}_{\text{diff}} + \alpha_3 \lambda_1 \mathcal{L}_{\text{align}} + \alpha_3 \lambda_2 \mathcal{L}_{\text{div}}}_{\text{(b) consolidation stage: once per domain}}, \tag{17}$$

where stage (a) updates only the masking adapter (System 1) together with the fidelity head of H_ψ, while stage (b) updates only the diffusion decoder D_ϕ and the consolidation-aligned generator H_ψ. Because the two stages act on disjoint parameter supports, the schedule is mathematically equivalent to alternating block coordinate descent on Eq. (16), which preserves its descent guarantees while eliminating the cross-stage gradient interference that would arise under naive joint optimization.

By optimizing Eq. 16, DCDHippo ensures that local adaptation is continuously anchored to the evolving global invariant truth.

4 Experiments

In this section, we perform extensive experiments to assess the effectiveness of the proposed **DCDHippo** framework under challenging unsupervised continual graph domain adaptation scenarios. Specifically, we aim to answer: **RQ1**-how DCDHippo performs compared to existing baselines on multiple benchmarks under intensified shift settings (in terms of AP and AF); **RQ2**-how robust DCDHippo is against the proposed *Recursive Drift* phenomenon induced by poisoning topology injection in the temporal stream; **RQ3**-the contributions of key components (e.g., CAD and EHM) to overall performance; and **RQ4**-how core hyperparameters (e.g., λ, λ_1, and λ_2) influence the performance and stability of DCDHippo. We first introduce the experimental settings, and then answer these questions in the subsequent subsections.

Methods	OGB-Arxiv (2-Year Gap)		Elliptic (Sudden Drift)		Twitch-Explicit (Cultural Shift)		Facebook-100 (Standard)	
	AP-ACC↑	AF↑	AP-F1↑	AF↑	AP-AUC↑	AF↑	AP-ACC↑	AF↑
1) Graph TTA (One-step Adaptation)								
TENT [47]	30.76±0.89	-3.29±0.54	32.13±1.59	-5.13±1.03	46.85±1.12	-2.47±1.36	46.62±0.52	0.67±0.22
GraphTTA [17]	34.62±0.58	-2.19±0.30	40.22±0.82	-3.82±0.62	48.91±0.56	-1.81±0.40	48.15±0.45	0.40±0.13
2) Invariant Learning (AF Inapplicable)								
EERM [56]	36.24±0.11	N/A	44.53±0.39	N/A	50.17±0.28	N/A	49.74±0.01	N/A
OOD-CGL [23]	37.10±0.25	N/A	45.80±0.41	N/A	51.26±0.31	N/A	50.89±0.21	N/A
3) Continual Graph Learning								
EWC [18]	35.57±0.40	-0.24±0.18	42.10±0.58	-0.58±0.20	49.50±0.43	0.19±0.07	48.56±0.35	**0.85±0.17**
CoTTA [49]	38.23±0.30	-1.95±0.24	43.03±0.46	-1.97±0.08	51.95±0.37	-0.39±0.49	50.17±0.10	0.50±0.15
CaT [29]	39.19±0.27	-1.02±0.18	45.61±0.33	-1.19±0.26	52.40±0.25	0.11±0.13	51.31±0.23	0.62±0.11
UCGL [12]	39.52±0.24	-0.80±0.14	46.22±0.22	-0.94±0.14	52.83±0.22	0.20±0.15	51.80±0.39	0.68±0.19
PDGNNs-TEM [65]	40.13±0.31	-0.73±0.19	47.13±0.31	-0.81±0.29	53.19±0.16	0.23±0.10	52.14±0.27	0.74±0.16
GCAL [36]	<u>41.27±0.26</u>	<u>-0.54±0.01</u>	<u>48.56±0.29</u>	<u>-0.48±0.15</u>	<u>53.59±0.17</u>	<u>0.47±0.12</u>	<u>52.77±0.37</u>	0.79±0.10
DCDHippo	**44.55±0.11**	**0.12±0.09**	**53.21±0.15**	**0.25±0.02**	**56.36±0.15**	**0.64±0.09**	**54.39±0.13**	<u>0.82±0.24</u>

Table 1: Overall performance (AP) and Average Forgetting (AF) on four datasets. Facebook-100 follows the standard setting, while Twitch, Elliptic, and OGB-Arxiv adopt intensified shift settings. DCDHippo (highlighted) shows strong robustness. Best results are bolded, and second-best are <u>underlined</u>.

4.1 Experimental Settings

In this section, we describe the experimental settings employed to evaluate the performance of our framework under challenging unsupervised continual graph domain adaptation scenarios. To rigorously test model robustness, we intentionally construct severe distribution shifts across temporal and regional dimensions. Implementation details are provided in Appendix A.3.

Datasets. We evaluate our approach on four benchmark datasets: Facebook-100 [46], Twitch-Explicit [38], OGB-Arxiv [13], and Elliptic [33]. These datasets are configured to capture two representative types of distributional shifts: *temporal* and *regional*.

For *temporal shift*, we employ OGB-Arxiv and Elliptic with intensified settings. In OGB-Arxiv, we utilize papers published prior to 2011 for pre-training. To simulate drastic evolution in scientific topics, we introduce a *two-year gap* between consecutive target domains (e.g., 2013, 2015, etc.) during the adaptation phase, rather than the standard one-year interval. For Elliptic, unlike previous works that discard the initial snapshots due to extreme class imbalance, we explicitly *restore* the first six snapshots and include them in the adaptation sequence. This introduces severe label distribution shifts and *sudden drifts* to challenge the model's stability.

For *regional shift*, we use Twitch-Explicit and Facebook-100. In Twitch-Explicit, we construct a *Cultural Shift* scenario by grouping networks based on geographical and cultural context. The model is pre-trained on Western region networks (DE, ENGB, ES, FR, PTBR) and adapted to Eastern region networks (RU, TW), creating a significant structural and feature-based divergence. For Facebook-100, we follow the standard setting [36], using Amherst41, Caltech36, and Johns Hopkins55 for pre-training, and continuously adapting to the remaining 11 university networks. Across all four benchmarks, the pre-training split is also used to warm up the slow diffusion decoder D_ϕ and to align-initialize the Hippocampal memory generator H_ψ, so that the first adaptation domain already enjoys a non-trivial global causal anchor (see Appendix A.3).

Metrics. Following standard evaluation protocols for continual learning [15], we employ a performance matrix $R \in \mathbb{R}^{T\times T}$ to record the testing results, where $R_{i,j}$ denotes the model's performance on the test set of domain j after the model has finished adapting to domain i. Based on this matrix, we report two key metrics:

- **Average Performance (AP):** This metric evaluates the model's overall efficacy across all domains after the entire adaptation process is completed. It is calculated as the mean of the final row of the performance matrix:

$$AP = \frac{1}{T}\sum_{j=1}^{T} R_{T,j} \tag{18}$$

  Higher AP indicates better overall adaptability and retention.
- **Average Forgetting (AF):** This metric quantifies the extent of knowledge loss on historical domains as the model adapts to new ones. It is defined as the average performance degradation of each domain from its initial learned state to the final state:

$$AF = \frac{1}{T-1}\sum_{j=1}^{T-1}(R_{T,j} - R_{j,j}) \tag{19}$$

  A value closer to zero indicates less forgetting, while a significantly negative value implies catastrophic forgetting.

For the base metric of $R_{i,j}$, we use *Accuracy* on **Facebook-100** and **OGB-Arxiv**, *ROC-AUC* on **Twitch-Explicit**, and *F1* (illicit class) on **Elliptic**.

4.1.1 Baselines. We compare against representative baselines in three categories: **Graph TTA** (one-step graph test-time adaptation), **graph invariant learning**, and **Continual Graph Learning (CGL)**. Graph TTA methods adapt the model only to the current snapshot/domain (i.e., one-step), while Graph invariant learning methods aim to capture environment-invariant signals for source-to-target generalization. Note that many invariant-learning baselines are designed for Graph Out-of-Distribution settings and typically train new parameters for each target graph/domain; therefore, their Average Forgetting (AF) is not applicable.

- **Graph TTA (one-step): TENT** [47] and **GraphTTA** [17]. These methods update the model on the current target domain only.
- **Graph invariant learning: EERM** [56] and **OOD-CGL** [23]. These methods learn invariant representations across environments to improve OOD generalization, but are not tailored to continual evaluation of forgetting.
- **Continual Graph Learning (CGL): EWC** [18], **CaT** [29], **GCAL** [36], **CoTTA** [49], **UCGL** [12], and **PDGNNs-TEM** [65]. Among them, **GCAL** and **CoTTA** are standard Continual Test-Time Adaptation methods.

4.2 Overall Experiment

To answer **RQ1**, we conducted a comprehensive comparison against a range of representative baselines, covering both general test-time adaptation methods and specialized Continual Graph Learning techniques. From Table 1, we can draw several key conclusions:

- First of all, our proposed **DCDHippo** consistently and significantly outperforms all baselines across all four datasets, achieving the highest Average Performance (AP) and the lowest Average Forgetting (AF). This robust superiority is particularly evident in the *Cultural Shift* scenario of Twitch-Explicit and the *sudden drifts* setting of Elliptic. This highlights the efficacy of our *Causal-Anisotropic Diffusion* mechanism, which effectively disentangles invariant causal factors from environment-specific noise, allowing the model to adapt to drastic distribution shifts without the catastrophic collapse observed in methods like TENT and GraphTTA.
- Secondly, compared to Invariant Learning methods (EERM, OOD-CGL) which focus solely on generalization, and Graph TTA methods which focus purely on plasticity, CGL methods (GCAL, CoTTA, PDGNNs-TEM) strike a better balance between learning new tasks and retaining old ones. However, even among CGL competitors, DCDHippo exhibits superior stability. Methods like CaT rely on graph condensation, which may lose fine-grained structural details during severe shifts. In contrast, our *Evolving Hippocampal Memory* dynamically updates high-level prototypes, enabling more efficient knowledge retention and significantly lower (or even positive) Average Forgetting (AF) rates.
- Furthermore, a clear performance progression is observed when handling severe shifts. In the intensified OGB-Arxiv (2-year gap) and Elliptic (restored imbalance) settings, naive adaptation and CGL methods (Tent, CaT and UCGL) fail significantly due to the accumulation of errors. While recent baselines like PDGNNs-TEM show improved resilience, they still suffer from performance degradation when the domain gap becomes too wide. DCDHippo effectively mitigates this by bridging diverse domains through causal invariant learning, establishing a new state-of-the-art for unsupervised continual graph domain adaptation.
- Finally, we provide a deeper analysis of the AF metric. AF is reported as "N/A" for invariant-learning methods (e.g., EERM, OOD-CGL) since they often re-initialize or train domain-specific parameters for each graph, making forgetting along a single continual timeline ill-defined. Notably, DCDHippo as well as strong CGL baselines such as GCAL and CoTTA can yield **positive** AF, suggesting **positive backward transfer**: learning from later domains strengthens representations and improves performance on earlier domains, rather than merely preserving it.

4.3 Deeper analysis of Recursive Drift Phenomenon: Poisoning Topology Injection.

To answer **RQ2**, we analyze the performance evolution on OGB-Arxiv under the Poisoning Topology Injection setting with a **two-year gap interval**. Figure 3 illustrates the performance trajectories of DCDHippo compared to representative Continual Graph Learning (CGL) baselines (e.g., CaT, GCAL). The specific setting is detailed in Section A.1. The injection occurs at step $t = 3$ (Year 2017), representing the midpoint of the adaptation sequence, and we observe two distinct phases of model behavior:

- **Phase I: Immediate Impact (Robustness against Structural Noise).** At the injection step $t = 3$ (Year 2017), all models experience a performance drop due to the sudden shift towards heterophily. However, baselines relying on direct structural aggregation (e.g., standard GNN backbones used in CaT) suffer a precipitous collapse, as the message-passing mechanism propagates noisy neighborhood information indiscriminately. In contrast, DCDHippo exhibits significantly higher resistance. This validates the effectiveness of our *Causal-Anisotropic Diffusion*, which acts as a structural filter. By distinguishing between invariant causal connections and environment-specific noise (the injected heterophilous edges), our model suppresses the aggregation of harmful neighbors, maintaining a relatively stable representation even under attack.
- **Phase II: Long-term Consequence (Mitigating Recursive Drift).** The most critical behavior arises in the post-injection phase ($t > 3$, corresponding to 2019 and later), when the input data stream reverts to its natural distribution. Nevertheless, baseline methods exhibit a pronounced hysteresis effect: their performance does not recover and instead remains persistently degraded in subsequent time steps, indicating the presence of *Recursive Drift* (see Appendix 2.1). Because these methods indiscriminately condense historical graph instances, the poisoned topology introduced at $t = 3$ is stored as "toxic memories" in their replay buffers. During later training, these corrupted memories are repeatedly replayed, continually injecting biased gradients $\nabla_\theta \mathcal{L}$ that steer optimization away from the true objective and form a self-reinforcing feedback loop of accumulated errors.

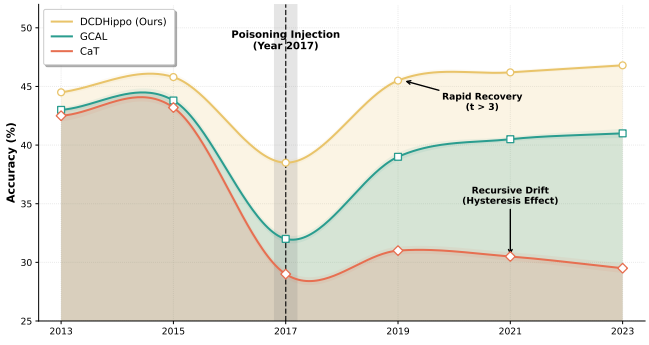


Figure 3: Performance evolution under Poisoning Topology Injection in OGB-Arxiv. Injection at $t = 3$ (2017) triggers a drop; many baselines fail to recover (recursive drift), while DCDHippo rebounds in later steps.

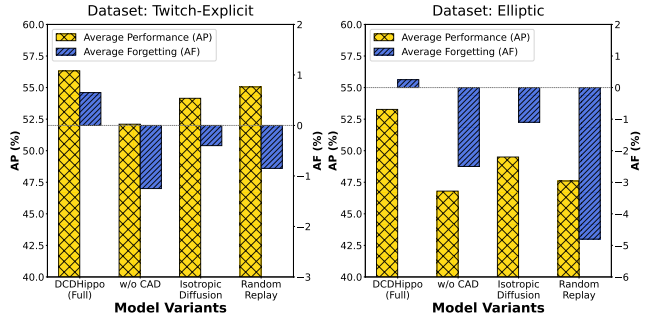


Figure 4: Ablation study on Twitch-Explicit and Elliptic. We compare the full model with three key variants, reporting Average Performance (AP) and Average Forgetting (AF).

Conversely, DCDHippo recovers quickly and returns to near-optimal performance at $t = 5$. We attribute this to the *Evolving Hippocampal Memory*, which reactivates past knowledge via a lightweight generator and applies confidence-aware updates to filter unreliable signals induced by the poisoned topology. By avoiding indiscriminate replay of corrupted instances, DCDHippo prevents memory pollution and breaks the recursive-drift feedback loop.

4.4 Ablation Study

To answer **RQ3**, we verify the effectiveness of each component in DCDHippo via ablation studies on two representative datasets: Twitch-Explicit (Regional Shift) and Elliptic (Sudden Drift). We compare the full model with three variants: 1) **w/o CAD**: Removes the Causal-Anisotropic Diffusion module, relying solely on the GNN backbone. 2) **Isotropic Diffusion**: Replaces the causal-guided diffusion with standard isotropic diffusion, where structural information is propagated uniformly without differentiating causal neighbors. 3) **Random Replay**: Replaces the Evolving Hippocampal Memory with a standard replay buffer that stores randomly sampled raw graph snapshots.

As shown in Fig. 4, removing or altering key components leads to varying degrees of performance degradation:

- **Impact of Causal-Anisotropic Diffusion (CAD):** The variant **w/o CAD** exhibits the lowest performance, confirming that standard GNN aggregation is insufficient to handle severe distribution shifts. Crucially, simply adding diffusion (**Isotropic Diffusion**) improves AP but results in negative Forgetting (AF). This suggests that isotropic propagation indiscriminately aggregates environmental noise and spurious correlations along with structural information. In contrast, our **Full Model** with causal guidance effectively filters out structural noise, achieving the best stability.
- **Impact of Evolving Hippocampal Memory (EHM):** Compared with **Random Replay**, our prototype-based memory yields consistently stronger stability. Although Random Replay achieves reasonable Average Performance on Twitch, it incurs severe forgetting on Elliptic (AF drops to −4.80%). We attribute this degradation to *memory pollution*: the random buffer indiscriminately stores samples from the restored early snapshots, which exhibit extreme class imbalance and label-distribution shifts. Replaying these toxic samples in later steps biases the update signals and amplifies error accumulation. In contrast, EHM maintains a compact set of evolving prototypes and performs outlier-aware updates aligned with the inferred causal structure, thereby mitigating the propagation of noisy history and preventing recursive drift.

4.5 Hyperparameter Sensitivity

To address **RQ4**, we conducted systematic sensitivity analyses to investigate the influence of three core hyperparameters in DCDHippo: the causal protection strength λ, the alignment weight λ_1, and the diversity weight λ_2. These experiments were carried out on representative datasets including Twitch-Explicit, OGB-Arxiv, and Elliptic, covering regional, temporal, and class-imbalance shifts. The results are summarized in Figure 5. These analyses shed light on the optimal parameter configurations and further elucidate the internal mechanisms underpinning our model's robustness against recursive drift.

(i) We first examined the impact of the **Causal Protection Strength** λ in our Causal-Anisotropic Diffusion module (Equation (7)), varying λ in the range of $[0, 1]$. As shown in Figure 5(a) and (b), the results on both Twitch-Explicit and OGB-Arxiv exhibit a consistent bell-shaped trend, peaking between 0.6 and 0.8. Notably, when $\lambda \to 0$, performance drops significantly, indicating that isotropic diffusion indiscriminately aggregates structural noise, failing to filter the "poisoned" topology. Conversely, when $\lambda \to 1$, the model becomes overly rigid, hindering the plasticity required for adapting to genuine distribution shifts. This confirms that a balanced protection mechanism is crucial for distinguishing invariant causal structures from environmental noise.

(ii) Next, we investigated the **Alignment Weight** λ_1 in the Hippocampal Evolutionary Objective (Equation (11)), varying its value from 0.01 to 5.0. Figure 5(c) reveals that performance on OGB-Arxiv improves rapidly as λ_1 increases and stabilizes around $\lambda_1 = 1.0$. This parameter governs the distillation strength from the slow System 2 (Diffusion) to the fast Hippocampal Memory. If λ_1 is

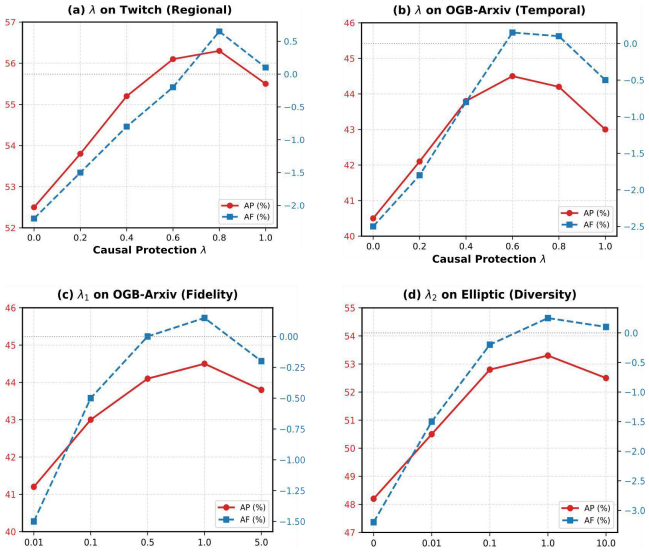


Figure 5: Hyperparameter sensitivity of DCDHippo. (a,b) Causal Protection λ on Twitch-Explicit/OGB-Arxiv. (c) Alignment weight λ_1 on OGB-Arxiv. (d) Diversity weight λ_2 on Elliptic.

set too low, the lightweight memory generator fails to capture the complex global invariants distilled by the diffusion model, leading to memory drift and reduced replay quality.

(iii) Finally, we studied the effect of the **Diversity Weight** λ_2 on the Elliptic dataset, which is characterized by severe class imbalance and sudden drifts. Figure 5(d) demonstrates a sharp performance gain as λ_2 increases from 0 to 1.0. Crucially, setting $\lambda_2 = 0$ results in suboptimal F1 scores and high forgetting rates. This is attributable to *mode collapse*, where the memory stores only dominant prototypes (normal transactions) and ignores rare but critical patterns (illicit transactions). Introducing explicit diversity forcing ensures the retention of a diverse population of prototypes, which is essential for robust replay in imbalanced streams.

5 Related Work

In this section, we briefly review two closely related research lines: graph invariant learning under distribution shifts and continual graph learning.

5.1 Graph Invariant learning

Learning invariant graph representations under distribution shifts is central to robust graph learning [5, 24], aiming to capture stable causal patterns while suppressing spurious correlations. Existing literatures [11, 55, 63] can be broadly grouped into: (i) *Explicit Structure Disentanglement* and (ii) *Implicit Invariance Regularization*. The first line identifies invariant subgraphs [3, 4, 8, 31, 59] that causally determine labels. For example, DIR [57] and AIA [42] employ rationale generators to split graphs into causal vs. environmental parts and emphasize causal information, while GIL [25] learns structural invariance via a maximal invariant subgraph generator. The second line enforces invariance through objectives or robust training without directly editing topology [16, 28, 58, 62]. Inspired by IRM [1], GRM [50] and EERM [56] penalize risk variance across environments; contrastive methods [6, 53, 71] further encourage agreement across augmented views. Despite progress, explicit methods may still be unstable due to discrete optimization and reliance on strong environment assumptions.

5.2 Continual Graph Learning

Continual Graph Learning (CGL) aims to enable GNNs to learn from a task sequence or evolving graph snapshots while mitigating catastrophic forgetting [44, 69]. Unlike i.i.d. Euclidean streams, graph data exhibit non-stationary topology and cross-task dependencies, making it difficult to balance stability and plasticity. Existing approaches broadly fall into three paradigms: regularization/distillation, parameter isolation, and replay. Regularization-based methods preserve past knowledge by constraining updates or distilling structure-aware semantics (e.g., MSCGL [2] and SEM [68]). Parameter-efficient isolation [39, 61, 64] reduces interference by freezing backbones and learning lightweight prompts/adapters, such as PI-GNN [64] and G-Adapter [10]. Replay-based methods rehearse compact buffers, synthesize pseudo-graphs (e.g., Generative Replay [32, 41, 48]), or select representative samples via gradient matching [22, 26, 29, 30, 36, 45]. Nevertheless, most CGL frameworks still treat stored structures or prompts as static evidence and lack principled mechanisms to filter spurious, context-dependent patterns, which can induce error accumulation over long horizons.

6 Conclusion

We studied graph continual learning under non-stationary distribution shifts and identified *recursive error accumulation* as a key driver of long-term performance collapse. To address this challenge, we proposed **DCDHippo**, a cognitive-bifurcation framework that couples a fast Progressive Causal Masking module for online noise pruning with a slow Causal-Anisotropic Diffusion process that internalizes causal skeletons into a global invariant pattern. An Evolving Hippocampal Memory further closes the loop by re-activating generative knowledge and distilling it back to the fast adapter, improving both adaptation and retention over time. Extensive experiments across diverse benchmarks demonstrate consistent gains over strong baselines in accuracy and forgetting metrics, while also providing interpretable causal subgraphs. In future work, we will explore more scalable diffusion backbones and extend the framework to broader streaming settings such as heterophilic and heterogeneous graphs.

7 Acknowledgement

This work was supported in part by the of China No. 62476101, 62406100 ,62572199, 92467109, the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515140137, in part by the Major Key Project of PCL (Grant No. PCL2025A11 and No. PCL2025A13), Tianjin Natural Science Foundation under Grants No. 24JCQNJC00320, Beijing Postdoctoral Research Foundation, China Postdoctoral Science Foundation under Grant No. 424018. Carl Yang was not supported by any fund from China.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019).

[2] Jie Cai, Xin Wang, Chaoyu Guan, Yateng Tang, Jin Xu, Bin Zhong, and Wenwu Zhu. 2022. Multimodal continual graph learning with neural architecture search. In *Proceedings of the ACM Web Conference 2022*. 1292–1300.

[3] Wei Chen, Yiqing Wu, Zhao Zhang, Fuzhen Zhuang, Zhongshi He, Ruobing Xie, and Feng Xia. 2024. FairGap: Fairness-aware recommendation via generating counterfactual graph. *ACM Transactions on Information Systems* 42, 4 (2024), 1–25.

[4] Wei Chen, Meng Yuan, Zhao Zhang, Ruobing Xie, Fuzhen Zhuang, Deqing Wang, and Rui Liu. 2025. FairDgcl: Fairness-aware recommendation with dynamic graph contrastive learning. *IEEE Transactions on Knowledge and Data Engineering* (2025).

[5] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, MA Kaili, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. 2022. Learning causally invariant representations for out-of-distribution generalization on graphs. *Advances in Neural Information Processing Systems* 35 (2022), 22131–22148.

[6] Hongyi Du, Xuewei Li, and Minglai Shao. 2025. Graph out-of-distribution generalization through contrastive learning paradigm. *Knowledge-Based Systems* 315 (2025), 113316.

[7] Kaile Du, Fan Lyu, Linyan Li, Fuyuan Hu, Wei Feng, Fenglei Xu, Xuefeng Xi, and Hanjing Cheng. 2023. Multi-label continual learning using augmented graph convolutional network. *IEEE Transactions on Multimedia* 26 (2023), 2978–2992.

[8] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. 2022. Debiasing graph neural networks via learning disentangled causal substructure. *Advances in Neural Information Processing Systems* 35 (2022), 24934–24946.

[9] Lukas Galke, Iacopo Vagliano, Benedikt Franke, Tobias Zielke, Marcel Hoffmann, and Ansgar Scherp. 2023. Lifelong learning on evolving graphs under the constraints of imbalanced classes and new classes. *Neural Networks* 164 (2023), 156–176.

[10] Anchun Gui, Jinqiang Ye, and Han Xiao. 2024. G-adapter: Towards structure-aware parameter-efficient transfer learning for graph transformer networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12226–12234.

[11] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. 2022. Good: A graph out-of-distribution benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 2059–2073.

[12] Thanh Duc Hoang, Do Viet Tung, Duy-Hung Nguyen, Bao-Sinh Nguyen, Huy Hoang Nguyen, and Hung Le. 2023. Universal graph continual learning. *arXiv preprint arXiv:2308.13982* (2023).

[13] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

[14] Wenyue Hua and Yongfeng Zhang. 2022. System 1+ system 2= better world: Neural-symbolic chain of logic reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 601–612.

[15] Khurram Javed and Martha White. 2019. Meta-learning representations for continual learning. *Advances in neural information processing systems* 32 (2019).

[16] Tianrui Jia, Haoyang Li, Cheng Yang, Tao Tao, and Chuan Shi. 2024. Graph invariant learning with subgraph co-mixup for out-of-distribution generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8562–8570.

[17] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. 2022. Empowering graph representation learning with test-time graph transformation. *arXiv preprint arXiv:2210.03561* (2022).

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

[19] Xiaoyu Kou, Yankai Lin, Shaobo Liu, Peng Li, Jie Zhou, and Yan Zhang. 2020. Disentangle-based continual graph representation learning. *arXiv preprint arXiv:2010.02565* (2020).

[20] Guojie Li, Zhiwen Yu, Ziwei Fan, Kaixiang Yang, and C. L. Philip Chen. 2026. Weighted Subspace Graph Learning for High-Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering* 38, 3 (2026), 2094–2107. doi:10.1109/TKDE.2026.3656436

[21] Guojie Li, Zhiwen Yu, Kaixiang Yang, C. L. Philip Chen, and Xuelong Li. 2025. Ensemble-Enhanced Semi-Supervised Learning With Optimized Graph Construction for High-Dimensional Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47, 2 (2025), 1103–1119. doi:10.1109/TPAMI.2024.3486319

[22] Guojie Li, Zhiwen Yu, Kaixiang Yang, Jianming Lv, and C. L. Philip Chen. 2026. Pseudo-Label Similarity Graph-Driven Multi-View Contrastive Clustering. *IEEE Transactions on Multimedia* (2026), 1–13. doi:10.1109/TMM.2026.3660170

[23] Haoyang Li, Xin Wang, Zeyang Zhang, Haibo Chen, Ziwei Zhang, and Wenwu Zhu. 2024. Disentangled graph self-supervised learning for out-of-distribution generalization. In *Forty-first International Conference on Machine Learning*.

[24] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. 2025. Out-of-distribution generalization on graphs: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[25] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. 2022. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems* 35 (2022), 11828–11841.

[26] Huihui Liu, Yiding Yang, and Xinchao Wang. 2021. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 8653–8661.

[27] Tang Liu, Baijun Wu, Wenzheng Xu, Xianbo Cao, Jian Peng, and Hongyi Wu. 2021. RLC: A reinforcement learning-based charging algorithm for mobile devices. *ACM Transactions on Sensor Networks (TOSN)* 17, 4 (2021), 1–23.

[28] Yang Liu, Xiang Ao, Fuli Feng, Yunshan Ma, Kuan Li, Tat-Seng Chua, and Qing He. 2023. FLOOD: A flexible invariant learning framework for out-of-distribution generalization on graphs. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 1548–1558.

[29] Yilun Liu, Ruihong Qiu, and Zi Huang. 2023. Cat: Balanced continual graph learning with graph condensation. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1157–1162.

[30] Yilun Liu, Ruihong Qiu, Yanran Tang, Hongzhi Yin, and Zi Huang. 2024. PUMA: Efficient continual graph learning for node classification with graph condensation. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[31] Yanhu Mo, Xiao Wang, Shaohua Fan, and Chuan Shi. 2024. Graph contrastive invariant learning from the causal perspective. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 8904–8912.

[32] Arnab Kumar Mondal, Jay Nandy, Manohar Kaul, and Mahesh Chandran. 2024. Stochastic Experience-Replay for Graph Continual Learning. In *The Third Learning on Graphs Conference*.

[33] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 5363–5370.

[34] Massimo Perini, Giorgia Ramponi, Paris Carbone, and Vasiliki Kalavri. 2022. Learning on streaming graphs with experience replay. In *Proceedings of the 37th ACM/SIGAPP symposium on applied computing*. 470–478.

[35] Brad E Pfeiffer. 2020. The content of hippocampal "replay". *Hippocampus* 30, 1 (2020), 6–18.

[36] Ziyue Qiao, Qianyi Cai, Hao Dong, Jiawei Gu, Pengyang Wang, Meng Xiao, Xiao Luo, and Hui Xiong. 2025. Gcal: Adapting graph models to evolving domain shifts. *arXiv preprint arXiv:2505.16860* (2025).

[37] Yixin Ren, Li Ke, Dong Li, Hui Xue, Zhao Li, and Shuigeng Zhou. 2023. Incremental graph classification by class prototype construction and augmentation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2136–2145.

[38] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.

[39] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016).

[40] Qinghua Shen, Weijieying Ren, and Wei Qin. 2023. Graph Relation Aware Continual Learning. *arXiv preprint arXiv:2308.08259* (2023).

[41] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems* 30 (2017).

[42] Yongduo Sui, Qitian Wu, Jiancan Wu, Qing Cui, Longfei Li, Jun Zhou, Xiang Wang, and Xiangnan He. 2023. Unleashing the power of graph data augmentation on covariate distribution shift. *Advances in Neural Information Processing Systems* 36 (2023), 18109–18131.

[43] Li Sun, Junda Ye, Hao Peng, Feiyang Wang, and Philip S Yu. 2023. Self-supervised continual graph learning in adaptive riemannian spaces. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 4633–4642.

[44] Zonggui Tian, Du Zhang, and Hong-Ning Dai. 2024. Continual learning on graphs: A survey. *arXiv preprint arXiv:2402.06330* (2024).

[45] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. 2022. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 99–108.

[46] Amanda L Traud, Peter J Mucha, and Mason A Porter. 2012. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications* 391, 16 (2012), 4165–4180.

[47] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726* (2020).

[48] Junshan Wang, Wenhao Zhu, Guojie Song, and Liang Wang. 2022. Streaming graph neural networks with generative replay. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 1878–1888.

[49] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition. 7201–7211.
[50] Song Wang, Zhen Tan, Yaochen Zhu, Chuxu Zhang, and Jundong Li. 2025. Generative risk minimization for out-of-distribution generalization on graphs. *arXiv preprint arXiv:2502.07968* (2025).
[51] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).
[52] Yuxin Wang, Yuanning Cui, Wenqiang Liu, Zequn Sun, Yiqiao Jiang, Kexin Han, and Wei Hu. 2022. Facing changes: continual entity alignment for growing knowledge graphs. In *International Semantic Web Conference*. Springer, 196–213.
[53] Zixu Wang, Bingbing Xu, Yige Yuan, Huawei Shen, and Xueqi Cheng. 2024. Negative as positive: enhancing out-of-distribution generalization for graph contrastive learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2548–2552.
[54] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.
[55] Man Wu, Xin Zheng, Qin Zhang, Xiao Shen, Xiong Luo, Xingquan Zhu, and Shirui Pan. 2024. Graph learning under distribution shifts: A comprehensive survey on domain adaptation, out-of-distribution, and continual learning. *arXiv preprint arXiv:2402.16374* (2024).
[56] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. 2022. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466* (2022).
[57] Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872* (2022).
[58] Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. 2022. Learning substructure invariance for out-of-distribution molecular representations. *Advances in Neural Information Processing Systems* 35 (2022), 12964–12978.
[59] Yonghui Yang, Le Wu, Yuxin Liao, Zhuangzhuang He, Pengyang Shao, Richang Hong, and Meng Wang. 2025. Invariance matters: Empowering social recommendation via graph invariant learning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2038–2047.
[60] XU Yishi, Yingxue Zhang, GUO Huifeng, Ruiming Tang, and GENG Yanhui. 2023. Graph structure aware incremental learning for recommender system. US Patent App. 18/111,066.
[61] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. 2017. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547* (2017).
[62] Guixian Zhang, Shichao Zhang, and Guan Yuan. 2024. Bayesian graph local extrema convolution with long-tail strategy for misinformation detection. *ACM Transactions on Knowledge Discovery from Data* 18, 4 (2024), 1–21.
[63] Kexin Zhang, Shuhan Liu, Song Wang, Weili Shi, Chen Chen, Pan Li, Sheng Li, Jundong Li, and Kaize Ding. 2024. A survey of deep graph learning under distribution shifts: from graph out-of-distribution generalization to adaptation. *ACM Transactions on Knowledge Discovery from Data* (2024).
[64] Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. 2023. Continual learning on dynamic graphs via parameter isolation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*. 601–611.
[65] Xikun Zhang, Dongjin Song, Yixin Chen, and Dacheng Tao. 2024. Topology-aware embedding memory for continual learning on expanding networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4326–4337.
[66] Xikun Zhang, Dongjin Song, and Dacheng Tao. 2022. Hierarchical prototype networks for continual graph representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 4 (2022), 4622–4636.
[67] Xikun Zhang, Dongjin Song, and Dacheng Tao. 2022. Sparsified subgraph memory for continual graph representation learning. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1335–1340.
[68] Xikun Zhang, Dongjin Song, and Dacheng Tao. 2023. Ricci curvature-based graph sparsification for continual graph representation learning. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
[69] Xikun Zhang, Dongjin Song, and Dacheng Tao. 2024. Continual learning on graphs: Challenges, solutions, and opportunities. *arXiv preprint arXiv:2402.11565* (2024).
[70] Fan Zhou and Chengtai Cao. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4714–4722.
[71] Yun Zhu, Haizhou Shi, Zhenshuo Zhang, and Siliang Tang. 2024. Mario: Model agnostic recipe for improving ood generalization of graph contrastive learning. In *Proceedings of the ACM Web Conference 2024*. 300–311.

A Appendix

A.1 Experiment setting of Recursive Drift Phenomenon

To empirically investigate the *Recursive Drift Phenomenon* and evaluate the model's resilience against error propagation, we design a specific **Poisoning Topology Injection** experiment on the OGB-Arxiv dataset. While standard temporal shifts involve natural evolution, this experiment introduces a deliberate *structural distribution shift* to induce catastrophic failure modes in memory-based methods. Specifically, at an intermediate time step t_{poison} (e.g., the year 2014, representing the mid-point of the adaptation sequence), we inject a "structural poison" by altering the graph topology $G_{t_{poison}}$ while preserving node features. We employ a *Heterophily Injection Strategy*: we randomly rewire 50% of the intra-class edges (connecting nodes with the same label) to connect nodes with different labels. This manipulation drastically reduces the homophily ratio, creating a noisy graph structure that contradicts the underlying semantic patterns. This setting serves as a critical stress test: if a Continual Learning model blindly condenses this poisoned graph into its memory bank, the erroneous structural bias will be replayed in subsequent steps ($t > t_{poison}$), theoretically triggering the recursive accumulation of errors even after the data stream returns to a normal distribution.

A.2 Baselines

The baseline methods we compared are categorized into three main types: Graph Test-Time Adaptation (Graph TTA), Invariant Learning, and Graph Continual Learning (GCL). Concretely, **Graph TTA** includes TENT and GraphTTA, which adapt to the current test graph in a one-step manner without explicitly preserving historical knowledge; **Invariant Learning** includes EERM and OOD-GCL, which aim to extract environment-invariant information for source-to-target generalization (and thus their Average Forgetting (AF) results are inapplicable because they typically train new parameters for each graph); and **GCL** includes EWC, CaT, GCAL, CoTTA, UGCL, and PDGNNs-TEM, which continuously adapt over the stream while mitigating catastrophic forgetting.

- **TENT [47]:** A fully test-time adaptation method that updates model parameters by minimizing the entropy of model predictions on test data, serving as a representative instance-level adaptation baseline.
- **GraphTTA [17]:** A test-time adaptation framework specifically designed for Graph Neural Networks (GNNs). It utilizes self-supervised tasks to adjust the model to the target graph structure and node features without accessing source domain labels.
- **EERM [56]:** A graph-specific method tailored for handling distribution shifts, which maximizes the variance of risk to simulate diverse environments and enhance the model's generalization ability to Out-Of-Distribution (OOD) data.
- **OOD-GCL [23]:** An Out-Of-Distribution Graph Contrastive Learning method that learns invariant graph representations. It employs a contrastive objective to distinguish causal, invariant subgraphs from environment-specific correlations, thereby improving robustness against distribution shifts.

- **EWC [18]:** A classic regularization-based continual learning method (Elastic Weight Consolidation) that mitigates forgetting by penalizing changes to parameters that are important for previous tasks, estimated via the Fisher Information Matrix.
- **CaT [29]:** A graph condensation-based continual learning method that addresses data imbalance and memory efficiency by synthesizing small, informative graphs for replay instead of storing raw samples.
- **GCAL [36]:** A graph continual adaptive learning method that employs a bilevel optimization strategy. It combines an information maximization approach for adaptation with a variational memory graph generation module to condense original graphs into memories for replay.
- **CoTTA [49]:** A standard continual test-time adaptation framework that combines weight-averaged predictions with stochastic neuron restoration to mitigate error accumulation and catastrophic forgetting during the continuous adaptation process.
- **UGCL [12]:** A universal graph continual learning framework that maintains knowledge by enforcing consistency in both local and global structural representations through a specialized distillation mechanism.
- **PDGNNs-TEM [65]:** A parameter-decoupled GNN framework combined with a Topology-aware Embedding Memory (TEM), designed to effectively store and replay complete topological information using compact embedding vectors.

A.3 Implementation Details

We follow the common continual graph adaptation protocol in the referenced setting (see the paper excerpt in the main text) while slightly tuning hyperparameters for our stronger shift configurations. Hyperparameters are selected on the validation split of each target domain.

Backbone. We use a 2-layer GCN as the backbone for Facebook-100, Twitch-Explicit, and Elliptic, and use a 2-layer GraphSAGE for OGB-Arxiv. The hidden dimension is set to $d = 256$ with dropout 0.5.

Optimization. We use AdamW with gradient clipping (max-norm = 1.0). For pre-training, we set lr = 2×10^{-4} and weight decay = 10^{-3}, and train for 150 epochs with early stopping (patience 20). During this pre-training phase, we additionally warm up the diffusion decoder D_ϕ on the source-domain snapshot G_0 and initialize the Hippocampal generator H_ψ by minimizing $\mathcal{L}_{\text{align}}$ against this initial D_ϕ. This ensures that the earliest online adaptation steps already benefit from a meaningful global causal anchor through $\mathcal{L}_{\text{replay}}$, avoiding the cold-start problem in which no global manifold would otherwise exist before the first consolidation phase. For online adaptation, we set lr = 8×10^{-4} and weight decay = 5×10^{-4}, and adapt for 5 epochs per domain (early stopping enabled).

Memory and diffusion. The Evolving Hippocampal Memory stores at most 200 prototypes per domain (up to 1,000 in total); the memory is updated once after finishing adaptation on each domain. For Causal-Anisotropic Diffusion, we use $K = 8$ diffusion steps. Unless otherwise stated, we set $(\lambda, \lambda_1, \lambda_2) = (0.7, 1.0, 1.0)$.

Hardware. All experiments are conducted on a single NVIDIA 5090 GPU (32GB memory).