INVARIANT STRUCTURAL DISTILLATION VIA GRAPH CONDENSATION FOR HISTORY-AWARE GRAPH CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Continual Learning (GCL) tackles the stability-plasticity dilemma: models must retain historical knowledge while rapidly adapting to evolving graph distributions under restricted observability. Recent graph condensation methods improve stability by compressing past task data into compact, high-fidelity memory. We empirically demonstrate that fidelity-focused condensation can lead to *Temporally Illusory Invariance*: memories that overfit task-specific spurious motifs, which appear stable within a task but fail to transfer as the graph structure evolves. Meanwhile, real-world graph streams often remain structurally connected across tasks. Current-task nodes may still interact with locally observable but unlabeled historical neighbors, even when raw historical graphs and labels are no longer accessible. We formalize this realistic setting as *history-aware restricted observability*. Exposing historical neighbors creates a tension: it may provide additional context, but it also introduces cross-task neighborhood mixing, which can blur class boundaries and amplify the brittleness of shortcut-based decision rules, ultimately propagating bias and degrading model robustness. To address these challenges, we propose **ISD-GC** (Invariant Structure Distillation for Graph Condensation), which reframes the memory buffer as a *transferable structural prior* rather than a fidelity-optimized snapshot, following a *Connect-then-Condense* principle. First, ISD-GC performs memory-grounded context disambiguation through *Virtual Soft Edges*, which construct relative evidence between historical memory prototypes and current-task anchors. A calibrated context discriminator further identifies historical candidates and distinguishes beneficial, irrelevant, and harmful context for selective message passing. Second, ISD-GC condenses each task into *structure-invariant* prototypes by distilling cross-view consistent structural information across complementary graph views and stochastic structural perturbations. This approach suppresses brittle motif reliance and improves transferability under structural drift. Extensive experiments on multiple class-incremental GCL benchmarks, supported by a controlled *Motif Trap* diagnostic, demonstrate that ISD-GC improves long-term retention and robustness while maintaining strong performance under strict memory budgets.

1 INTRODUCTION

Graph Neural Networks (GNNs) Hamilton et al. (2017); Kipf & Welling (2016); Scarselli et al. (2008) excel at modeling relational dependencies on static graphs. Yet many real-world graphs are non-stationary: new nodes and edges arrive, interaction patterns drift, and a node's neighborhood context evolves over time. Graph Continual Learning (GCL) Wang et al. (2020) studies how to learn from such graph streams while balancing plasticity and stability. In class-incremental streams, the learner updates the model at step t using only the currently observed task subgraph and labels for newly arriving classes, while past raw graphs are discarded. Under such restricted observability, replay-based methods are often competitive, and recent works further combine replay with graph condensation to compress past experience into a compact set of prototype-like memories for efficient rehearsal or retraining Liu et al. (2023; 2025). A broader discussion of continual learning, condensation-based replay, and graph robustness and invariance is provided in Appendix A.

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

Despite strong empirical performance, existing condensation-based replay pipelines face two coupled challenges. **(C1) Fidelity-driven condensation under structural shift.** Many existing methods optimize memory to reproduce the currently observed step-t subgraph as closely as possible, for example via gradient Jin et al. (2021) or training-dynamics matching Zheng et al. (2023). However, in evolving graphs each step provides only a restricted and time-local view of an underlying process. A memory optimized for snapshot fidelity can therefore preferentially preserve short-horizon regularities that look stable within the current window but fail to transfer as topology and neighborhood context drift. We term this failure mode **Temporally Illusory Invariance**: time-local structural cues that are spuriously correlated with labels within a limited window and thus mistaken as invariant, yet break under subsequent structural shift. Moreover, condensation can crystallize such cues into memory and turn them into persistent supervision signals that are repeatedly replayed or used for retraining, which induces memory-driven bias propagation over time. **(C2) Context ambiguity under history-aware restricted observability.** Unlike conventional task-isolated protocols, we consider a realistic history-aware setting in which the step-t subgraph includes unlabeled historical neighbors that remain locally observable around current-task nodes. Without access to past raw graphs or historical labels, the learner cannot determine whether these neighbors provide beneficial semantic context or inject cross-task interference through message passing. This problem is not equivalent to standard out-of-distribution detection. When context nodes must be identified, the encoder has not yet been aligned to task t, and the target is a growing set of historical-task nodes rather than arbitrary distributional outliers. These challenges are intertwined: fidelity-driven condensation can crystallize time-local structural shortcuts into memory, while uncontrolled cross-task neighborhood mixing can further amplify the brittleness of such shortcut-based representations as the graph evolves.

To examine these challenges empirically, we conduct two diagnostic studies. First, for C1, we introduce a controlled *Motif Trap*, where a structural motif is predictive early in the task stream but loses its discriminative specificity in later tasks. We observe that memory-limited learners consolidate this temporally local structural correlation into their decision rule; when the correlation no longer transfers to subsequent tasks, performance deteriorates. This failure pattern provides empirical evidence for *Temporally Illusory Invariance*. Second, for C2, we examine how observable history enters the model through inter-task connectivity. We find that naively exposing cross-task neighbors generally increases neighborhood mixing and degrades performance, whereas completely removing such connectivity induces structural isolation and can misrepresent realistic history-aware streams. This tension motivates selective use of observable historical context rather than either indiscriminate inclusion or complete isolation.

To meet these challenges, we propose **ISD-GC**, short for **I**nvariant **S**tructure **D**istillation for **G**raph **C**ondensation. ISD-GC treats memory as a transferable structural prior rather than a fidelity-optimized snapshot and consists of two indispensable components. First, it performs **memory-grounded context disambiguation** by using condensed prototypes as semantic anchors and *Virtual Soft Edges* to identify useful historical context while suppressing cross-task interference. Second, it performs **structure-invariant prototype condensation** by distilling prototypes that remain consistent across complementary structural views and stochastic perturbations, reducing brittle motif reliance under structural drift. We note that our objective is not to reduce the computational cost of graph continual learning itself, but to improve the utility and transferability of compact memory under restricted budgets. The additional modules introduced by ISD-GC are designed around bounded prototype memory and do not require access to the full historical graph. Our contributions are summarized as follows.

- *Temporally illusory invariance and bias propagation.* We identify a memory-centric failure mode of fidelity-driven condensation, where time-local structural shortcuts are crystallized into memory and propagate as systematic bias under structural shift.
- *Historical context disambiguation.* We propose a prototype-anchored mechanism for identifying historical context under restricted observability and selectively reducing cross-task interference.
- *Structure-invariant prototype condensation.* We introduce an invariant distillation objective that learns compact prototype memory by enforcing consistency across complementary structural views, preventing task-local motifs from being consolidated as spurious invariants.
- *Targeted diagnostics and evaluation.* We propose the *Motif Trap* diagnostic and perform systematic evaluations under history-aware and isolated protocols with strict memory budgets, together with ablations on context identification and invariant condensation.

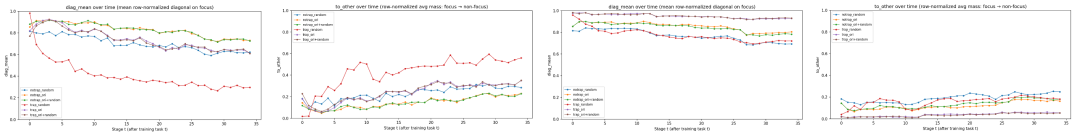
2 PROBLEM DIAGNOSIS: TEMPORALLY ILLUSORY INVARIANCE AND CROSS-TASK NEIGHBORHOOD MIXING

We empirically diagnose two coupled phenomena in graph continual learning (GCL) under restricted observability: **temporally illusory invariance** and **cross-task neighborhood mixing**. The first challenge arises when structural cues learned during the training of a task appear stable within the limited task subgraph, but these cues fail to transfer when the structure-label correlation drifts across tasks. Meanwhile, history-aware streams expose unlabeled historical neighbors through inter-task edges, creating a secondary source of information. This channel can either provide useful context or introduce interference. We study these effects through two controlled diagnostics: (i) *Motif Trap*, which injects a time-local motif-label correlation followed by a conflict phase to test whether learners over-commit to short-horizon structural shortcuts that later fail under drift; and (ii) *History-aware Connectivity Analysis*, which examines how inter-task edges affect cross-task neighborhood mixing and whether such mixing becomes more severe in shortcut-prone regimes.

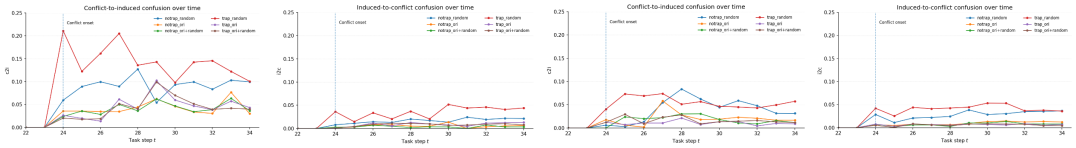
2.1 PROTOCOL

Following the CGLB Zhang et al. (2022a) setting, we compare an oracle reference model with a memory-limited learner on the CoraFull dataset. The oracle performs full-batch GNN training with access to all labeled nodes observed up to step t. The memory-limited regime follows PUMA: it trains on the current-task labeled data together with a bounded replay buffer. We consider three feature regimes: ORI uses the original attributes, RANDOM replaces them with i.i.d. Gaussian noise, and ORI+RANDOM adds Gaussian perturbations to the original attributes. The W/TRAP and W/OTRAP settings share the same wedge scaffold and differ only in triangle closure. We partition motif-bearing classes into an induced set and a later conflict set. In tasks $t \leq 13$ (through class 26), triangle closure occurs only for $\mathcal{C}_{\text{ind}}^{\triangle}$ and is therefore strongly correlated with the label. Conflict begins at $t = 24$ and intensifies over $t \in [24, 30]$, during which classes in $\mathcal{C}_{\text{con}}^{\triangle}$ also exhibit closure, rendering the motif non-specific. This yields a progressively weakening motif–label correlation, emulating gradual structural drift in streams. Remaining classes form the control set $\mathcal{C}_{\text{ctrl}}$. We ask whether prediction errors concentrate in the trap-relevant induced-conflict subspace rather than diffusing uniformly across classes. Let $\mathcal{C}_{\text{focus}} = \mathcal{C}_{\text{ind}}^{\triangle} \cup \mathcal{C}_{\text{con}}^{\triangle}$, and let $P_t(\hat{y} \mid y)$ denote the row-normalized confusion matrix over all observed test classes at step t. We characterize motif-induced error patterns using (i) within-focus correctness $\text{Diag}_{\text{focus}}$, (ii) leakage to control $m_{\text{focus}\to\text{ctrl}}$, and (iii) directed induced-conflict masses $m_{\text{con}\to\text{ind}}$ and $m_{\text{ind}\to\text{con}}$ computed from the row-normalized confusion matrix. Details are provided in Appendix B.

2.2 MOTIF TRAP: TEMPORALLY ILLUSORY INVARIANCE



(a) PUMA: `diag_mean` (b) PUMA: `to_other` (c) Oracle: `diag_mean` (d) Oracle: `to_other`



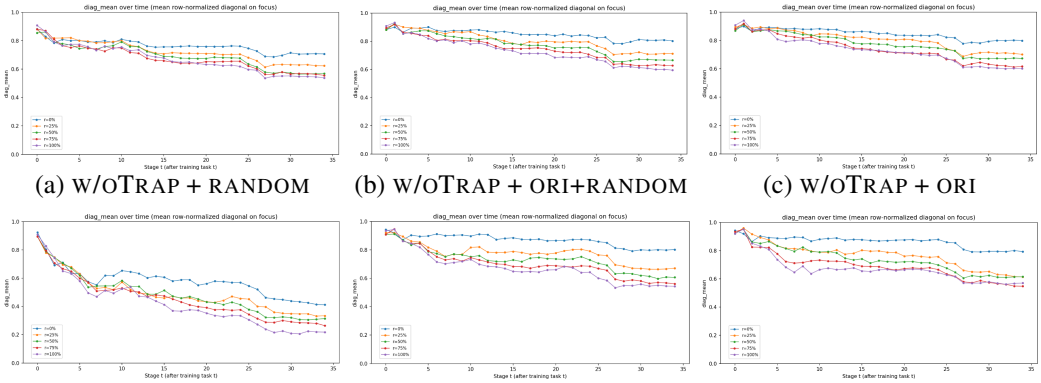
(e) PUMA: `c2i` (f) PUMA: `i2c` (g) Oracle: `c2i` (h) Oracle: `i2c`

Figure 1: Motif-trap diagnostics of information-flow drift and conflict-stage transitions.

Fig. 1 provides empirical evidence for *temporally illusory invariance*: (i) The trap induces *early* representation drift, not just late-stage conflict errors. Before any conflict classes appear ($t < 24$), the motif trap already changes the trajectory of information flow under memory limitation (Fig. 1(a,b)). In contrast, the oracle reference remains much more stable in both `diag_mean` $= \text{Diag}_{\text{focus}}(t)$ and `to_other` $= m_{\text{focus}\to\text{ctrl}}(t)$ (Fig. 1(c,d)), indicating that access to richer labeled history suppresses

shortcut-driven drift. (ii) Restricted history consolidates the shortcut more aggressively than full-history training. Across feature regimes, the gap between memory-limited and oracle behavior is largest when attributes are suppressed. Under RANDOM, the memory-limited learner is forced to rely on structural aggregation, and the trap becomes an especially attractive shortcut (Fig. 1(a,b)). The oracle exhibits the same directional effect but with substantially smaller magnitude (Fig. 1(c,d)). (iii) When conflict arrives, failures become *structured* and directional. For $t \geq 24$, errors do not diffuse arbitrarily; instead they concentrate along the induced-conflict axis. The directed block masses become non-zero and grow after conflict onset, with the strongest response under W/TRAP+RANDOM for the memory-limited learner (Fig. 1(e,f)). Moreover, the transition is typically asymmetric: $m_{\text{con}\to\text{ind}}$ (`c2i`) is larger than $m_{\text{ind}\to\text{con}}$ (`i2c`), suggesting that newly arriving conflict classes are systematically absorbed into the earlier induced decision regions that were shaped when the motif was highly predictive. The oracle shows the same qualitative transition signature but with weaker magnitude (Fig. 1(g,h)). Together, the core issue is not merely late-stage conflict errors, but the early formation and consolidation of a non-transferable structural rule. This diagnosis motivates methods that explicitly discourage shortcut consolidation in memory and favor representations/prototypes that remain transferable under structural drift. Additional analyses are provided in Appendix C.

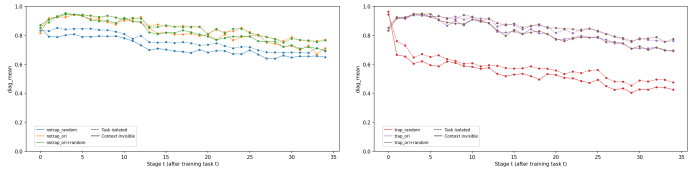
2.3 HISTORY-AWARE CONNECTIVITY: OBSERVABLE CONTEXT AND CROSS-TASK NEIGHBORHOOD MIXING



(a) W/OTRAP + RANDOM (b) W/OTRAP + ORI+RANDOM (c) W/OTRAP + ORI

(d) W/TRAP + RANDOM (e) W/TRAP + ORI+RANDOM (f) W/TRAP + ORI

Figure 2: Oracle dose-response analysis of inter-task edge restoration. We vary the inter-task edge recovery rate $r \in \{0, 25, 50, 75, 100\}\%$.



(a) W/OTRAP (b) W/TRAP

Figure 3: Memory-limited (PUMA): task-isolated vs. context-visible inputs.

Removing inter-task edges reduces cross-task message passing and can yield sharper decision boundaries by limiting cross-class neighborhood mixing, but it also induces *structural isolation* by restricting nodes to incomplete neighborhoods. This motivates a dose-response study that progressively restores inter-task edges to quantify how cross-task structural context trades off against cross-class interference and shortcut-driven signatures, and how this tradeoff interacts with the motif trap. We study this question under two complementary settings: (i) an **oracle** dose-response restoration of inter-task edges (Fig. 2), and (ii) a **history-aware** evaluation for memory-limited condensation/replay baselines PUMA under realistic cross-task connectivity (Fig. 3): at step t, the model receives the current-task subgraph together with all observable 1-hop historical neighbors and the inter-task edges connecting these historical neighbors to current-task nodes. Figure 2 and Figure 3 show that, relative to task-isolated inputs, exposing observable cross-task neighbors generally degrades performance. When attribute cues are weak and shortcut reliance is stronger, naive inclusion

of observable history mainly increases cross-class neighborhood mixing, which worsens shortcut-driven brittleness rather than providing net benefit. These results highlight the need for selective use of observable history. Methods that suppress interfering historical neighbors while retaining useful context are critical for robust history-aware continual GCL. Additional results and further analyses are provided in Appendix D.

3 METHOD

We propose **ISD-GC** (**I**nvariant **S**tructure **D**istillation for **G**raph **C**ondensation), a class-incremental GCL framework that treats memory as a *transferable structural prior* rather than a fidelity-optimized snapshot. Under restricted observability, the learner accesses only the current history-aware task graph and prototype memory $\mathcal{M}_{t-1}$, without raw historical graphs. ISD-GC follows a **Connect-then-Condense** principle and consists of two stages. **Stage I** (§3.1) performs *memory-grounded context disambiguation* to identify useful historical context and suppress interference. **Stage II** (§3.2) performs *structure-invariant prototype condensation* by enforcing consistency across disambiguated, history-aware, and perturbed views. After condensation, the predictor is trained solely on the accumulated prototype memory $\mathcal{M}_{\leq t}$ under the *pure-memory* protocol. The complete procedure is provided in Algorithm 1 in Appendix E.

3.1 MEMORY-GROUNDED CONTEXT IDENTIFICATION

As diagnosed in Sec. 2, history-aware class-incremental streams expose current-task nodes together with unlabeled historical neighbors, creating a *context–interference ambiguity*: historical context may provide useful semantic evidence or induce harmful cross-task neighborhood mixing. Under restricted observability, raw historical graphs and labels are unavailable, while the drifting label space makes absolute OOD or uncertainty scores unreliable. We therefore ground context disambiguation in bounded prototype memory: *Virtual Soft Edges (VSE)* provide relative evidence between historical-context and current-task hypotheses, from which a transferable context mask is learned. The detected historical candidates are further typed by their task-specific utility as beneficial, irrelevant, or harmful context.

3.1.1 RELATIVE-EVIDENCE TEACHER VIA VIRTUAL SOFT EDGES

Let $(\boldsymbol{X}_t, \boldsymbol{A}_t)$ denote the history-aware task graph at step t and $\mathcal{M}_{t-1}$ the prototype memory. A frozen encoder f_{ref} produces normalized task embeddings $\boldsymbol{H} = \text{norm}(f_{\text{ref}}(\boldsymbol{X}_t, \boldsymbol{A}_t)) \in \mathbb{R}^{N_t \times d}$. Let $\boldsymbol{U}_{\text{old}}$ denote normalized historical prototype embeddings. For each arriving class $c \in \mathbb{C}_t$, we construct current-task anchors $\boldsymbol{u}_{\text{new}}^{(c)} = \text{norm}\Big(\frac{1}{|\Omega_c|}\sum_{v\in\Omega_c} \boldsymbol{h}_v\Big)$, where $\Omega_c = \{v \in \mathbb{V}_t^{\text{tr}} : y_v = c\}$, and stack them into $\boldsymbol{U}_{\text{new}}$. We use lightweight projections $\boldsymbol{W}_Q, \boldsymbol{W}_K$ to align task nodes and anchors: $\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{W}_Q^\top, \boldsymbol{K}_{\text{old}} = \boldsymbol{U}_{\text{old}}\boldsymbol{W}_K^\top, \boldsymbol{K}_{\text{new}} = \boldsymbol{U}_{\text{new}}\boldsymbol{W}_K^\top$. The corresponding scaled compatibility matrices are $\boldsymbol{S}_{\text{old}} = \frac{\boldsymbol{Q}\boldsymbol{K}_{\text{old}}^\top}{\sqrt{d}}$ and $\boldsymbol{S}_{\text{new}} = \frac{\boldsymbol{Q}\boldsymbol{K}_{\text{new}}^\top}{\sqrt{d}}$. To reduce noisy historical matches, only the top-k entries of $\boldsymbol{S}_{\text{old}}$ are retained, yielding $\widetilde{\boldsymbol{S}}_{\text{old}}$. We compute relative evidence through $s_{\text{old}}(v) = \log\Big(\frac{1}{k}\sum_i \exp\Big([\widetilde{\boldsymbol{S}}_{\text{old}}]_{v,i}\Big)\Big)$ and $s_{\text{new}}(v) = \log\Big(\frac{1}{|\mathbb{C}_t|}\sum_j \exp([\boldsymbol{S}_{\text{new}}]_{v,j})\Big)$. The memory-grounded historical affinity is defined as $o(v) = \sigma\Big(\frac{s_{\text{old}}(v) - s_{\text{new}}(v)}{\tau_{\text{ctx}}}\Big) \in [0,1]$, where $\tau_{\text{ctx}} > 0$ is a temperature parameter. A larger $o(v)$ indicates that node v is better explained by historical memory anchors than by current-task anchors.

3.1.2 ANCHOR-CALIBRATED CONTEXT MASK

The VSE score $o(v)$ is a useful *teacher* signal, but its absolute scale drifts across tasks. To obtain a deployable context mask without per-task tuning, we distill this signal into a lightweight discriminator $q_\phi(\boldsymbol{z}_v)$ and calibrate it using a stream-specific structural prior.

In the class-incremental setting, labeled training nodes at step t belong to newly arriving classes and therefore constitute reliable non-context anchors, i.e. $\mathbb{N}_t = \mathbb{V}_t^{\text{tr}}$. The teacher threshold is obtained as $\delta_{\text{tea}} = \text{Quantile}\left(\{o(v) : v \in \mathbb{N}_t\}, q_{\text{tea}}\right)$ and candidate historical positives are selected by $\mathbb{P}_t = \{v :$

$o(v) > \delta_{\text{tea}}\}$, where q_{tea} is the teacher quantile controlling the confidence level of anchor selection. We construct the soft targets

$$y_v^{\text{tea}} = \begin{cases} o(v), & v \in \mathbb{P}_t \setminus \mathbb{N}_t, \\ 0, & v \in \mathbb{N}_t, \end{cases} \tag{1}$$

and optimize the discriminator with

$$L_{\text{ctx}} = \frac{1}{|\mathbb{P}_t \cup \mathbb{N}_t|} \sum_{v \in \mathbb{P}_t \cup \mathbb{N}_t} \text{BCEWithLogits}\left(\ell_\phi(\boldsymbol{z}_v), y_v^{\text{tea}}\right), \tag{2}$$

where $\ell_\phi(\boldsymbol{z}_v)$ is the discriminator logit. At inference, we compute $r_t(v) = q_\phi(\boldsymbol{z}_v)$ for all nodes and calibrate $\delta_t = \text{Quantile}(\{r_t(v) : v \in \mathbb{N}_t\}, q)$. The historical-context mask is then $[\boldsymbol{m}_t^{\text{hist}}]_v = \mathbf{1}_{r_t(v) > \delta_t}$, adapting to score-scale drift without task-specific threshold tuning.

3.1.3 Task-utility typing of historical candidates

The historical-candidate mask $\boldsymbol{m}_t^{\text{hist}}$ indicates which observed nodes are likely to originate from previous tasks, but historical origin alone does not determine how these nodes should participate in the current task. We therefore estimate the first-order utility of each detected historical candidate using a lightweight probe, defined as $u_t(v) = -\langle \nabla_{\boldsymbol{h}_v} L_{\text{probe}}, \boldsymbol{h}_v \rangle$. A positive utility indicates that removing v is predicted to increase the current-task loss, whereas a negative utility suggests potential interference. Accordingly, we partition historical candidates into beneficial subsets $[\boldsymbol{m}_t^{\text{ben}}]_v = [\boldsymbol{m}_t^{\text{hist}}]_v \mathbf{1}_{u_t(v) > \tau_u}$ and harmful subsets $[\boldsymbol{m}_t^{\text{harm}}]_v = [\boldsymbol{m}_t^{\text{hist}}]_v \mathbf{1}_{u_t(v) < -\tau_u}$, while the remaining historical candidates $[\boldsymbol{m}_t^{\text{irr}}]_v = [\boldsymbol{m}_t^{\text{hist}}]_v \mathbf{1}_{-\tau_u \le u_t(v) \le \tau_u}$ are treated as irrelevant. The resulting typed masks are used to construct the complementary structural views for Stage II. Further implementation details of Stage I are provided in Appendix F.1.

3.2 Structure-invariant Prototype Condensation

Most condensation-based replay methods optimize memory to reproduce the currently observed snapshot. In evolving graph streams, such fidelity-oriented objectives can be misaligned with continual transfer: each task exposes only a time-local view, and structural drift can invalidate correlations that appear stable within that window. As a result, faithfully compressing the step-t snapshot may consolidate transient motifs into memory and propagate bias across later tasks. We therefore reposition condensation toward **transferability**, operationalized as *structure-level invariance*: prototypes should preserve class-consistent structural signatures that remain stable under plausible structural variations of the same history-aware task graph.

3.2.1 Complementary Structural Views

Given the Stage-I masks $\boldsymbol{m}_t^{\text{ben}}$, $\boldsymbol{m}_t^{\text{irr}}$, and $\boldsymbol{m}_t^{\text{harm}}$, we construct three complementary views. The *Disambiguated* view applies typed gating: $[\boldsymbol{X}_t^{\text{dis}}]_{v,:} = g_t(v)[\boldsymbol{X}_t]_{v,:}$, where $g_t(v) = 1$ for beneficial or non-historical nodes, $g_t(v) = o(v)$ for irrelevant historical nodes, and $g_t(v) = 0$ for harmful historical nodes. The corresponding edge gate is $\gamma_t(u, v) = g_t(u)g_t(v)$, yielding $[\boldsymbol{A}_t^{\text{dis}}]_{u,v} = \gamma_t(u, v)[\boldsymbol{A}_t]_{u,v}$. The *History-aware* view retains the original observable topology $\boldsymbol{A}_t^{\text{hist}} := \boldsymbol{A}_t$, while suppressing harmful historical context in $\boldsymbol{X}_t^{\text{hist}}$. The *Perturbation* view is generated by applying stochastic edge perturbations to the Disambiguated view, yielding $\widetilde{\boldsymbol{A}}_t^{\text{pert}}$. The resulting three views are therefore $\mathcal{G}_t^{\text{dis}} = (\boldsymbol{A}_t^{\text{dis}}, \boldsymbol{X}_t^{\text{dis}})$, $\mathcal{G}_t^{\text{hist}} = (\boldsymbol{A}_t^{\text{hist}}, \boldsymbol{X}_t^{\text{hist}})$, and $\mathcal{G}_t^{\text{pert}} = (\widetilde{\boldsymbol{A}}_t^{\text{pert}}, \boldsymbol{X}_t^{\text{dis}})$. Cross-view consistency across these complementary views serves as a proxy for transferable structural invariance, reducing reliance on brittle task-local motifs.

3.2.2 Multi-view Structure-invariant Prototype Distillation

Let $\boldsymbol{b} = \{b_c\}_{c \in \mathbb{C}_{\le t}}$ denote the per-class prototype allocation for all seen classes. At step t, we optimize prototypes only for $c \in \mathbb{C}_t$. Let $B_t^{\text{new}} = \sum_{c \in \mathbb{C}_t} b_c$, and collect the learnable prototype features as $\boldsymbol{P}_t \in \mathbb{R}^{B_t^{\text{new}} \times F}$. We reuse the same frozen encoder as f_{ref} in Stage I and as Enc in Stage II (notation separated for clarity across stages). Using a frozen teacher encoder $\text{Enc}(\cdot)$, we obtain the normalized embeddings $\boldsymbol{H}^{\text{dis}} = \text{norm}(\text{Enc}(\boldsymbol{X}_t^{\text{dis}}, \boldsymbol{A}_t^{\text{dis}}))$, $\boldsymbol{H}^{\text{hist}} = \text{norm}(\text{Enc}(\boldsymbol{X}_t^{\text{hist}}, \boldsymbol{A}_t^{\text{hist}}))$,

and $\boldsymbol{H}^{\mathrm{pert}} = \mathrm{norm}(\mathrm{Enc}(\boldsymbol{X}_t^{\mathrm{dis}}, \widetilde{\boldsymbol{A}}_t^{\mathrm{pert}}))$ for the three views, respectively. Using the same teacher without message passing over prototype edges, we obtain the normalized prototype embeddings $\boldsymbol{U} = \mathrm{norm}(\mathrm{Enc}_{\mathrm{noedge}}(\boldsymbol{P}_t)) \in \mathbb{R}^{B_t^{\mathrm{new}} \times d}$.

For prototype slot (c, i), we compute assignment weights $\alpha_{v,c,i} = \frac{\exp(\boldsymbol{h}_v^\top \boldsymbol{u}_{c,i}/\tau_p)}{\sum_{j=1}^{b_c} \exp(\boldsymbol{h}_v^\top \boldsymbol{u}_{c,j}/\tau_p)}$, where τ_p is a temperature parameter and $i \in \{1, \ldots, b_c\}$ denotes the i-th prototype allocated to class c. Using the shared assignment weights $\alpha_{v,c,i}$, we aggregate the teacher embeddings within each class to obtain the view-specific target $\tilde{\boldsymbol{h}}_{c,i}^{(e)} = \frac{\sum_{v \in \mathbb{V}_c} \alpha_{v,c,i} \boldsymbol{h}_v^{(e)}}{\sum_{v \in \mathbb{V}_c} \alpha_{v,c,i}}$, where $e \in \{\mathrm{dis}, \mathrm{hist}, \mathrm{pert}\}$. To emphasize stable representation factors, we compute the cross-view variance $\bar{\boldsymbol{h}}_{c,i} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \tilde{\boldsymbol{h}}_{c,i}^{(e)}$ and the corresponding element-wise variance $\boldsymbol{s}_{c,i}^2 = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} (\tilde{\boldsymbol{h}}_{c,i}^{(e)} - \bar{\boldsymbol{h}}_{c,i})^{\odot 2}$. We then derive a parameter-free stability gate $\boldsymbol{g}_{c,i} = \frac{(1+\boldsymbol{s}_{c,i}^2)^{\odot(-1)}}{\mathrm{Mean}\left[(1+\boldsymbol{s}_{c,i}^2)^{\odot(-1)}\right]}$ such that representation dimensions with lower cross-view variation receive larger distillation weights. The prototypes are optimized by

$$\mathcal{L}_{\mathrm{ISD}} = \sum_{c \in \mathbb{C}_t} \sum_{i=1}^{b_c} \sum_{e \in \mathcal{E}} \lambda_e \left\| \boldsymbol{g}_{c,i}^{1/2} \odot \left(\boldsymbol{u}_{c,i} - \tilde{\boldsymbol{h}}_{c,i}^{e} \right) \right\|_2^2, \tag{3}$$

where $\mathcal{E} = \{\mathrm{dis}, \mathrm{hist}, \mathrm{pert}\}$, with $\lambda_{\mathrm{dis}} = \lambda_{\mathrm{pert}} = 1$ and λ_{hist} controlling the historical-view contribution. Unlike snapshot-fidelity condensation, Eq. 3 explicitly favors prototype representations that are stable across disambiguation, history-aware topology, and stochastic perturbation, thereby aligning memory construction with transferability under drift. After optimization, the prototypes are committed to $\mathcal{M}_t$ under the given budget and the predictor is retrained on $\mathcal{M}_{\leq t}$ under the *pure-memory* protocol. Implementation details are provided in Appendix F.2, and the computational overhead is discussed in Appendix G.

4 Experiments

We evaluate **ISD-GC** on class-incremental graph streams to answer four questions:

- *RQ1 (Effectiveness under history-aware restricted observability).* Can ISD-GC effectively learn under the proposed history-aware restricted-observability setting by selectively exploiting locally observable historical context while mitigating harmful cross-task interference?
- *RQ2 (Effectiveness under the isolated-control protocol).* Does ISD-GC remain effective when observable historical neighbors and cross-task edges are removed?
- *RQ3 (Transferability).* Does temporally invariant prototype condensation improve the transferability of condensed memory under structural drift and shortcut-prone regimes?
- *RQ4 (Mechanism validation and component interaction).* How do memory-grounded context disambiguation and structure-invariant prototype condensation contribute to the overall performance?

4.1 Experimental Setup

Datasets and task construction. We evaluate on four node-classification benchmarks: Cora-Full McCallum et al. (2000), ogbn-arxiv, ogbn-products Hu et al. (2020), and Reddit Hamilton et al. (2017). Following CGLB Zhang et al. (2022a), each dataset is organized into class-incremental tasks with two classes per task, using a $60\%/20\%/20\%$ train/validation/test split. All experiments follow the Class-IL setting. Dataset statistics are provided in Appendix H.1.

Learning and observation protocols. We adopt the *restricted-observability* setting, where the learner at step t accesses only the current task graph and bounded memory $\mathcal{M}_{t-1}$, without raw historical graphs. After task-t condensation, the predictor is trained solely on the accumulated memory $\mathcal{M}_{\leq t}$. Our primary *history-aware* protocol additionally exposes locally observable one-hop unlabeled historical neighbors and their inter-task edges, whereas the *isolated-control* protocol removes them and retains only current-task nodes and intra-task connectivity. This comparison separates the benefit of observable historical context from that of the condensation mechanism itself.

Baselines, metrics, and reporting. We compare ISD-GC with representative regularization-, distillation-, replay-, and condensation-based methods, including EWC Kirkpatrick et al. (2017),

LwF Li & Hoiem (2017), MAS Aljundi et al. (2018), GEM Lopez-Paz & Ranzato (2017), ER-GNN Zhou & Cao (2021), TWP Liu et al. (2021), SSM Zhang et al. (2022b), CAT Liu et al. (2023), and PUMA Liu et al. (2025). Joint training Caruana (1997) and fine-tuning Girshick et al. (2014) serve as upper and lower reference baselines, respectively. Unless otherwise specified, replay-based methods use a memory-budget ratio of 0.005, evenly allocated across tasks/classes, while Joint-train retains all training data. We report AvgAcc, LastAcc, and Forgetting, with higher AvgAcc/LastAcc and lower Forgetting indicating better performance. Results are reported as mean $\pm$ standard deviation over five random seeds. Backbone architecture, optimization budget, and model-selection protocol are matched across methods unless otherwise stated. Formal metric definitions and additional experimental details are provided in Appendix H.2.

4.2 Main Results under the History-aware Protocol (RQ1)

Table 1 reports the overall class-incremental performance under the history-aware protocol. ISD-GC consistently outperforms all continual-learning baselines across the four datasets. Despite using only a 0.005 memory-budget ratio, it remains competitive with Joint-train and surpasses it on several metrics on ogbn-arxiv, ogbn-products, and Reddit. On CoraFull, ISD-GC achieves the strongest performance among continual-learning methods but falls slightly behind Joint-train in AvgAcc, likely because the extremely small budget retains only two replay nodes over a substantially longer 35-task stream. Notably, while condensation-based CAT and PUMA already outperform most replay-, regularization-, and distillation-based alternatives, ISD-GC further improves upon them across all four datasets. These results indicate that effective history-aware continual learning depends not only on memory compactness, but also on selectively exploiting observable historical context and preserving transferable structural information in condensed memory.

Table 1: Overall performance under the history-aware protocol. Best results are shown in **bold**.

Method	CoraFull			ogbn-arxiv		
	AvgAcc↑	LastAcc↑	Forget.↓	AvgAcc↑	LastAcc↑	Forget.↓
Fine-tune	2.286±0.142	11.083±0.060	96.748±0.080	5.017±0.057	17.424±0.029	96.827±0.118
GEM	2.873±0.640	11.113±0.100	92.194±0.200	4.980±0.003	17.416±0.016	91.768±0.092
EWC	2.652±0.606	11.356±0.268	95.388±1.219	4.969±0.012	17.441±0.123	96.394±0.738
LWF	2.225±0.230	11.074±0.091	96.465±0.303	4.989±0.023	17.376±0.103	96.740±0.309
MAS	3.137±0.997	11.857±0.320	94.500±0.933	4.977±0.003	17.234±0.084	95.632±0.325
TWP	11.961±3.025	21.970±0.986	68.852±4.112	6.005±2.743	17.799±1.291	55.504±3.272
ER-GNN	2.510±0.597	11.058±0.012	95.696±0.516	16.635±2.550	33.292±0.904	84.589±2.647
SSM	2.448±0.227	11.079±0.110	95.896±0.131	16.565±2.961	34.859±1.070	84.606±3.065
CAT	65.716±3.180	70.641±1.776	15.909±2.336	62.329±2.317	74.729±1.258	17.038±1.745
PUMA	78.383±0.403	83.404±0.144	5.661±0.491	74.779±0.255	79.784±0.337	9.594±0.271
Joint-train	82.919±0.205	85.955±0.232	5.379±0.170	75.861±0.467	80.266±0.306	9.231±0.344
ISD-GC	**81.549±0.246**	**87.326±0.046**	**5.247±0.284**	**75.846±0.269**	**80.651±0.170**	**8.821±0.243**
Method	ogbn-products			Reddit		
	AvgAcc↑	LastAcc↑	Forget.↓	AvgAcc↑	LastAcc↑	Forget.↓
Fine-tune	4.310±0.000	15.899±0.013	97.267±0.200	4.964±0.004	18.082±0.354	99.414±0.027
GEM	6.315±1.006	16.237±0.157	93.633±0.125	4.983±0.030	20.470±1.285	91.207±1.861
EWC	4.531±0.203	16.061±0.166	96.712±0.681	5.058±0.205	18.162±0.335	99.256±0.187
LWF	4.677±0.504	15.969±0.050	96.760±0.731	4.967±0.006	18.122±0.300	99.437±0.038
MAS	4.280±0.032	15.913±0.096	96.778±0.333	4.932±0.012	18.554±0.968	98.911±0.122
TWP	5.046±0.056	18.932±1.803	49.514±5.734	5.033±0.056	18.249±0.295	98.616±0.906
ER-GNN	50.402±0.948	43.549±0.479	40.398±1.280	20.974±1.607	49.952±0.497	82.602±1.690
SSM	63.453±0.936	57.133±0.150	25.216±1.515	32.639±4.219	60.329±1.312	70.340±4.455
CAT	70.960±0.891	83.032±0.379	10.241±0.510	96.862±0.283	97.565±0.105	0.667±0.127
PUMA	81.930±0.265	87.670±0.083	5.461±0.133	97.009±0.086	98.003±0.024	0.346±0.059
Joint-train	86.257±0.163	89.991±0.245	3.453±0.300	98.722±0.041	99.833±0.022	0.382±0.049
ISD-GC	**83.261±0.518**	**88.870±0.059**	**3.406±0.638**	**98.009±0.113**	**99.059±0.012**	**0.255±0.127**

4.3 Results under the Isolated-control Protocol (RQ2)

Table 5 in the Appendix H.2 reports the complete results under the isolated-control protocol, where observable historical neighbors and cross-task edges are removed. With a 0.005 memory-budget ratio, ISD-GC consistently outperforms the continual-learning baselines across all four datasets and remains competitive with Joint-train on several metrics. Figure 4 further compares memory-based methods under budget ratios of 0.005, 0.01, 0.05, and 0.1. Across all four datasets, ISD-GC with only a 0.005 budget can match or outperform sampling-based replay methods using a 0.1 budget,

while remaining robust as the budget varies. These results show that ISD-GC does not depend on observable cross-task context or large replay budgets; instead, its condensation mechanism preserves transferable historical knowledge within compact prototype memory. On Reddit, performance changes little with increasing budget because it is already near saturation.

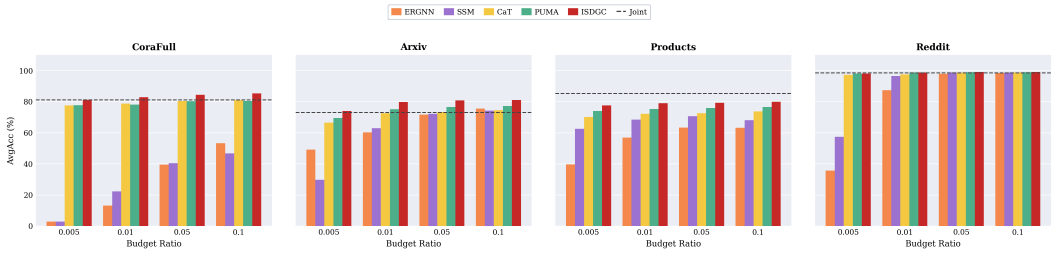


Figure 4: Performance under different memory budget ratios.

4.4 Transferability under Shortcut-Prone Regimes (RQ3)

To evaluate memory transferability under structural drift, we compare ISD-GC with PUMA under the controlled conditions in Sec. 2.2. As shown in Table 2, ISD-GC consistently improves performance on CoraFull, with particularly clear gains under the shortcut-prone w/Trap settings. The gains are smaller or less consistent under w/oTrap, where node attributes already provide strong predictive signals. Overall, ISD-GC exhibits stronger memory transferability under structural drift and shortcut-prone regimes. Results on ogbn-arxiv, ogbn-products, and Reddit show similar overall trends and are reported in Appendix I.2.

Table 2: Performance under structural drift and shortcut-prone regimes on CoraFull (RQ3).

Setting	Method	AvgAcc↑	LastAcc↑	Forget.↓
w/oTrap, ori	PUMA	74.939±0.877	82.231±0.006	7.182±0.794
	ISD-GC	**78.984±0.483**	**83.638±0.266**	**6.519±0.487**
w/Trap, ori	PUMA	56.010±0.831	65.764±0.405	9.695±0.982
	ISD-GC	**63.008±0.751**	**71.037±0.265**	**7.863±0.847**
w/oTrap, random	PUMA	68.279±0.305	75.013±0.135	3.800±0.496
	ISD-GC	**71.664±0.221**	**77.260±0.143**	**2.825±0.424**
w/Trap, random	PUMA	49.544±0.452	59.104±0.255	7.261±0.894
	ISD-GC	**55.004±0.572**	**62.759±0.188**	**5.901±0.713**

4.5 Ablation Analysis (RQ4)

ISD-GC consists of two key stages: memory-grounded context identification (Stage I) and structure-invariant prototype condensation (Stage II). Table 3 evaluates their contributions on CoraFull. Removing either stage degrades overall performance, while the complete model achieves the highest AvgAcc and LastAcc and the lowest Forgetting. These results suggest that Stage I and Stage II make complementary contributions to memory construction. Similar trends are observed on the other datasets, as reported in Appendix I.3. Moreover, sensitivity analysis in Appendix I.4 further shows that ISD-GC is not overly sensitive to key hyperparameter choices.

Table 3: Ablation study of the two stages of ISD-GC on CoraFull (RQ4).

Stage I	Stage II	AvgAcc↑	LastAcc↑	Forget.↓
×	×	79.237±0.242	84.919±0.131	5.475±0.242
×	✓	81.394±0.436	87.292±0.054	5.335±0.538
✓	×	81.380±0.187	86.941±0.132	5.326±0.082
✓	✓	**81.549±0.246**	**87.326±0.046**	**5.247±0.284**

5 Conclusion and Future Work

This paper studies class-incremental graph continual learning under history-aware restricted observability and proposes ISD-GC, which combines memory-grounded context identification with structure-invariant prototype condensation. By selectively exploiting historical context and preserving transferable structural knowledge, ISD-GC improves continual learning under compact memory budgets and structural drift. Our evaluation focuses on class-incremental benchmarks with simulated structural drift. Future work will extend ISD-GC to more realistic evolving graph streams and explore adaptive context selection and dynamic memory allocation.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 144–161, 2018.

Weizhi An, Wenliang Zhong, Feng Jiang, Hehuan Ma, and Junzhou Huang. Causal subgraphs and information bottlenecks: Redefining ood robustness in graph neural networks. In *European Conference on Computer Vision*, pp. 473–489, 2024.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Kaituo Feng, Changsheng Li, Xiaolu Zhang, and Jun Zhou. Towards open temporal graph neural networks. *arXiv preprint arXiv:2303.15015*, 2023.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Xiaoxue Han, Zhuo Feng, and Yue Ning. A topology-aware graph coarsening framework for continual graph learning. *Advances in Neural Information Processing Systems*, 37:132491–132523, 2024.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. volume 33, pp. 22118–22133, 2020.

Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580*, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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

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

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

Huihui Liu, Yiding Yang, and Xinchao Wang. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8653–8661, 2021.

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

Yilun Liu, Ruihong Qiu, Yanran Tang, Hongzhi Yin, and Zi Huang. Puma: Efficient continual graph learning for node classification with graph condensation. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):449–461, 2025.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Sohir Maskey, Raffaele Paolino, Fabian Jogl, Gitta Kutyniok, and Johannes Lutzeyer. Graph representational learning: When does more expressivity hurt generalization? In *International Conference on Learning Representations*, volume 2026, pp. 105345–105392, 2026.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

Chaoxi Niu, Guansong Pang, and Ling Chen. Graph continual learning with debiased lossless memory replay. In *ECAI 2024: 27th European Conference on Artificial Intelligence, 19–24 October 2024, Santiago de Compostela, Spain–Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, pp. 1808–1815, 2024.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz Khan. Continual deep learning by functional regularisation of memorable past. *Advances in neural information processing systems*, 33:4453–4464, 2020.

Zhen Peng, Xu Hua, Jingchen Hao, Qika Lin, Bo Dong, and Chao Shen. Self-supervised continual graph learning via adaptive spaced replay on node proxies. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2303–2314, 2025.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Junwei Su, Difan Zou, Zijun Zhang, and Chuan Wu. Towards robust graph incremental learning on evolving graphs. In *International Conference on Machine Learning*, pp. 32728–32748. PMLR, 2023.

Jerry Sun, Mohamed Abubakr Hassan, Yaoyu Zhang, Wanying Zhang, and Chi-Guhn Lee. Diverse and sparse mixture-of-experts for causal subgraph–based out-of-distribution graph learning. In *International Conference on Learning Representations*, volume 2026, pp. 79573–79597, 2026.

Binh Tang and David S Matteson. Graph-based continual learning. *arXiv preprint arXiv:2007.04813*, 2020.

Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. Streaming graph neural networks via continual learning. *ACM International Conference on Information & Knowledge Management*, 2020.

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

Yidi Wang, Ziyue Qiao, Jiawei Gu, Xubin Zheng, Pengyang Wang, Xiao Luo, et al. Out-of-distribution graph models merging. In *International Conference on Learning Representations*, volume 2026, pp. 53506–53532, 2026.

Zhizheng Wang, Yuanyuan Sun, Xiaokun Zhang, Bo Xu, Zhihao Yang, and Hongfei Lin. Continual learning with high-order experience replay for dynamic network embedding. *Pattern Recognition*, 159:111093, 2025.

Qiao Yuan, Sheng-Uei Guan, Pin Ni, Tianlun Luo, Prudence Wong, Victor Chang, and Ka Lok Man. Continual graph learning: A survey. *Pattern Recognition*, pp. 113600, 2026.

Kexin Zhang, Shuhan Liu, Song Wang, Weili Shi, Chen Chen, Pan Li, Sheng Li, Jundong Li, and Kaize Ding. A survey of deep graph learning under distribution shifts: from graph out-of-distribution generalization to adaptation. *ACM Transactions on Knowledge Discovery from Data*, 20(2):1–38, 2026.

Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. Continual learning on dynamic graphs via parameter isolation. pp. 601–611, 2023a.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Cglb: Benchmark tasks for continual graph learning. *Advances in Neural Information Processing Systems*, 35:13006–13021, 2022a.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Sparsified subgraph memory for continual graph representation learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 1335–1340. IEEE, 2022b.

Xikun Zhang, Dongjin Song, Yixin Chen, and Dacheng Tao. Topology-aware embedding memory for continual learning on expanding networks. pp. 4326–4337, 2024.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, and Wenwu Zhu. Out-of-distribution generalized dynamic graph neural network with disentangled intervention and invariance promotion. *arXiv preprint arXiv:2311.14255*, 2023b.

Xin Zheng, Miao Zhang, Chunyang Chen, Quoc Viet Hung Nguyen, Xingquan Zhu, and Shirui Pan. Structure-free graph condensation: From large-scale graphs to condensed graph-free data. *Advances in Neural Information Processing Systems*, 36:6026–6047, 2023.

Fan Zhou and Chengtai Cao. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4714–4722, 2021.

AI USE STATEMENT

In this work, generative AI tools were used during the preparation of this manuscript for language refinement and writing assistance. The use of these tools was limited to improving clarity and presentation; all scientific claims, experimental designs, implementations, analyses, and conclusions were developed and verified by the authors. The authors did not use generative AI tools to generate experimental results, perform undisclosed data analysis, create fabricated citations, or replace the authors' responsibility for the research process. All AI-assisted suggestions were manually reviewed, revised, and validated by the authors. The authors take full responsibility for the final content of this work, including all text, claims, and technical contributions.

ETHICS STATEMENT

This work does not involve human subjects, user studies, or the collection of private or sensitive personal data. We evaluate our method using publicly available graph benchmark datasets that are widely used in the graph learning community. The proposed method focuses on algorithmic development for graph continual learning and graph condensation. We do not introduce new data collection procedures, profiling mechanisms, or applications involving individual-level decision making. All experiments are conducted for research purposes, and we follow the usage guidelines and licenses associated with the employed datasets. The authors are responsible for the integrity, reproducibility, and proper interpretation of the reported results.

REPRODUCIBILITY STATEMENT

The proposed ISD-GC framework, including the memory-grounded context identification procedure, structure-invariant prototype condensation objective, and the overall learning algorithm, are described in Section 3 and Algorithm 1. Additional implementation details, including the parameterization of the context discriminator, Virtual Soft Edge construction, utility-based context typing, prototype optimization procedure, and memory update strategy, are provided in Appendix F. The experimental settings, including dataset statistics, task construction, evaluation protocols, baseline configurations, and metric definitions, are described in Section 4 and Appendix H. Additional ablation studies, sensitivity analyses, and extended experimental results are reported in Appendix I. The supplementary materials include the source code and configuration files required to reproduce the reported experiments.

A RELATED WORK

To situate our contributions, we review three lines of research most relevant to our setting: (i) graph continual learning under restricted observability, (ii) replay with condensed or synthetic memories, and (iii) robustness and invariance learning on graphs. This discussion clarifies why snapshot-fidelity condensation and generic OOD-style heuristics are insufficient in history-aware class-incremental streams, and motivates our memory-grounded context identification and structure-invariant condensation.

A.1 GRAPH CONTINUAL LEARNING UNDER RESTRICTED OBSERVABILITY

Graph continual learning Yuan et al. (2026); Zhang et al. (2026) studies learning from sequential graph tasks when past raw graphs and labels are unavailable, requiring a balance between retaining historical knowledge and adapting to newly arriving classes. Existing methods broadly fall into replay-based, regularization-based, and architecture or parameter-isolation paradigms Zhang et al. (2022a). Regularization-based methods preserve previous knowledge by adding penalty terms to the training objective, constraining either important model parameters or node embeddings to prevent drastic changes across tasks Su et al. (2023). For example, topology-aware regularization methods such as Liu et al. (2021) identify parameters that are important for preserving both prediction performance and topological information. However, such methods are often vulnerable to the "brittleness" problem Tang & Matteson (2020); Pan et al. (2020), as overly strong constraints may restrict the model's ability to acquire new knowledge. Architecture-based methods Zhang et al.

(2023a) allocate task-specific parameters or modules to different tasks, reducing interference but incurring increasing storage and computational costs as the task stream grows. Rehearsal-based methods Wang et al. (2025); Peng et al. (2025), by contrast, maintain a memory buffer or generated replay data to approximate historical distributions. Representative methods such as ER-GNN Zhou & Cao (2021) store selected historical nodes or subgraphs for experience replay, while later approaches replay diverse graph structures, including walk sequences Wang et al. (2022), neighborhood sets, and different motif structures Feng et al. (2023), to maintain learned knowledge.

A representative benchmark is CGLB Zhang et al. (2022a), which formalizes continual task constructions and provides a common ground for evaluating stability–plasticity trade-offs. However, many established protocols either adopt task-isolated subgraphs or do not explicitly address a key ambiguity in history-aware inputs: unlabeled historical neighbors may provide useful context but can also inject cross-task interference through message passing. Our work targets this ambiguity directly and builds a deployable mechanism that operates under restricted observability and evolving topology.

A.2 Replay with Condensed or Synthetic Memories

A central challenge in replay-based graph continual learning is the storage and computation cost of retaining historical graphs. Graph condensation tackles this by distilling a small set of synthetic or prototypical nodes whose training effect approximates that of the original task data.

CaT Liu et al. (2023) proposes a condense-and-train paradigm, where each task is condensed into a compact synthetic graph and the model is updated primarily using accumulated condensed memories, improving efficiency under strict budgets. PUMA Liu et al. (2025) further advances condensation-based replay via prototype-based memory and feature alignment, leveraging both labeled and unlabeled nodes and adopting a training-from-scratch strategy to alleviate class imbalance. Beyond condensation, DeLoMe Niu et al. (2024) explores debiased loss-preserving replay to reduce replay bias, while topology-aware embedding memories such as TEM Zhang et al. (2024) compress ego-subnetworks into compact embeddings for expanding networks. Orthogonally, topology-aware coarsening methods such as TACO Han et al. (2024) provide another form of structural compression in streaming settings.

While these approaches substantially improve efficiency and retention, they largely construct memories to reproduce the *currently observed snapshot* as faithfully as possible. In evolving graph streams, snapshot-fidelity objectives can be misaligned: time-local structural regularities may appear stable within a task yet fail under topology drift. Our work explicitly addresses this mismatch by shifting condensation from fidelity toward transferability via structure-invariant distillation.

A.3 Spurious Correlations, Invariance, and Robustness on Graphs

Robust learning on graphs has highlighted that GNN predictions can rely on spurious correlations encoded in local substructures or motifs, which can break under distribution shift Maskey et al. (2026). OOD generalization Sun et al. (2026); Wang et al. (2026); Zhang et al. (2026) and invariant learning methods mitigate such brittleness by identifying stable substructures, enforcing invariance across environments, or adopting information-bottleneck-style objectives Li et al. (2025). Recent advances include causal or invariant subgraph learning approaches such as CSIB An et al. (2024), and invariant dynamic information discovery methods such as I-DIDA Zhang et al. (2023b), which study invariance under evolving data distributions.

Most robustness and OOD studies are conducted in single-shot regimes where the model can be trained and calibrated on the target distribution before applying OOD detection or invariance enforcement. In class-incremental graph streams, this assumption is violated at step t: before learning the current task, the encoder is aligned only to past tasks and a bounded memory, while the input subgraph exposes new classes together with unlabeled historical neighbors. Moreover, existing robustness work typically does not model how spurious structural cues can be *consolidated, replayed, and amplified over time* under strict memory budgets. Our work complements this literature by introducing a continual, memory-centric failure mode, temporally illusory invariance, and by proposing a mechanism that identifies historical context via relative evidence *before* current-task alignment,

followed by structure-invariant distillation to prevent spurious motifs from being consolidated as transferable memory.

Summary and transition. In summary, prior continual learning on graphs and condensation-based replay have largely emphasized efficiency and forgetting mitigation. However, two gaps remain under history-aware class-incremental streams with restricted observability. First, the learner cannot reliably distinguish whether unlabeled historical neighbors provide useful context or cause cross-task interference during message passing. Second, optimizing condensed memory for snapshot fidelity can inadvertently consolidate time-local structural shortcuts, which may appear stable early on but break as the graph and neighborhood context drift. Before presenting our method, we therefore conduct a targeted diagnosis to make these failure modes observable and quantifiable. Specifically, we introduce a controlled *Motif Trap* to test shortcut formation and breakdown over time, and an inter-task connectivity analysis to characterize how history-aware neighborhood mixing reshapes the error patterns. These findings then motivate the design requirements that our approach satisfies.

B MOTIF-TRAP CONSTRUCTION

We introduce a controlled *structural shortcut* by attaching an identical auxiliary scaffold to *every* original node and varying only one motif indicator—*triangle closure*—for selected classes over time. For each original node v, we add two auxiliary nodes $\{a_v, b_v\}$ and edges (v, a_v) and (v, b_v), forming an open wedge. We optionally close the wedge by adding the edge (a_v, b_v), yielding a triangle motif (triadic closure). Auxiliary nodes are unlabeled and excluded from supervision and evaluation; they do not contribute to the training loss or any reported metrics. Thus, the intervention affects learning only through message passing. Unless stated otherwise, the wedge scaffold is attached in both W/TRAP and W/OTRAP; the only difference is whether the closing edge (a_v, b_v) is added for selected classes. We use an L-layer GNN with $L \geq 2$ so that triangle closure can influence the representation of v through multi-hop propagation.

To avoid introducing unintended attribute cues through auxiliary nodes, we define their features consistently with the feature regime. In ORI, we copy the feature of v to both auxiliaries, i.e., $\mathbf{x}_{a_v} = \mathbf{x}_{b_v} = \mathbf{x}_v$. In RANDOM, we sample $\mathbf{x}_{a_v}$ and $\mathbf{x}_{b_v}$ i.i.d. from the same distribution used to randomize node attributes, matching the marginal statistics of randomized features. In ORI+RANDOM, we copy $\mathbf{x}_v$ and add Gaussian perturbations $\mathbf{x}_{a_v} = \mathbf{x}_v + \boldsymbol{\epsilon}_a$ and $\mathbf{x}_{b_v} = \mathbf{x}_v + \boldsymbol{\epsilon}_b$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. For RANDOM and ORI+RANDOM, we sample noise from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. For both noise-based regimes, σ is scaled to match the average per-dimension energy of the original features:

$$\sigma \;=\; \frac{\mathbb{E}\big[\|\mathbf{x}_{\text{orig}}\|_2\big]}{\sqrt{F}}, \tag{4}$$

where $\mathbf{x}_{\text{orig}}$ denotes the original node features and F is the feature dimension. For CORAFULL, this yields $\sigma = 0.078$.

This design keeps the intervention *structurally controlled* and prevents auxiliary-node attribute artifacts beyond the intended feature setting.

Triangle closure is applied at the class level: once a class appears, all its nodes either always receive closure or never do. We partition motif-bearing classes into an induced set and a later conflict set,

$$\mathcal{C}^{\triangle}_{\text{ind}} = \{0, 2, 4, 6, 20, 22, 24, 26\}, \qquad \mathcal{C}^{\triangle}_{\text{con}} = \{48, 50, 52, 54, 56, 58, 60\},$$

thereby inducing a *time-local* motif–label correlation. In tasks $t \leq 13$ (through class 26), triangle closure occurs only for $\mathcal{C}^{\triangle}_{\text{ind}}$ and is therefore strongly correlated with the label. Conflict begins at task $t = 24$ (first appearance of class 48) and intensifies over $t \in [24, 30]$, during which classes in $\mathcal{C}^{\triangle}_{\text{con}}$ also exhibit closure, rendering the motif non-specific. This yields a progressively weakening motif–label correlation, emulating gradual structural drift in streams. All remaining classes form the control set $\mathcal{C}_{\text{ctrl}} = \{0, \ldots, 69\} \setminus (\mathcal{C}^{\triangle}_{\text{ind}} \cup \mathcal{C}^{\triangle}_{\text{con}})$.

C FINDINGS: EVIDENCE OF TEMPORALLY ILLUSORY INVARIANCE

Figure 1 reveals a coherent pattern that aligns with *temporally illusory invariance*: a time-local structural cue is (i) learned quickly, (ii) consolidated into the decision rule under limited history, and (iii) collapses into structured errors once the cue loses specificity.

- *The trap induces* early *representation drift, not just late-stage conflict errors.* Even before any conflict classes arrive ($k < 24$), the motif trap already changes the trajectory of information flow under memory limitation. In Fig. 1(a,b), W/TRAP+RANDOM shows a sharp and persistent drop in within-focus self-consistency (`diag_mean`) together with substantially higher leakage to non-focus classes (`to_other`), compared with W/OTRAP+RANDOM. This is precisely the signature of *illusory invariance formation*: the model treats triangle closure as a stable cue within the early tasks, and its predictions start drifting and diffusing *well before* the stream introduces the conflict set. In contrast, the oracle reference remains much more stable in both `diag_mean` and `to_other` (Fig. 1(c,d)), indicating that access to richer labeled history suppresses shortcut-driven drift.
- *Restricted history consolidates the shortcut more aggressively than full-history training.* Across feature regimes, the gap between memory-limited and oracle behavior is largest when attributes are suppressed. Under RANDOM, the memory-limited learner is forced to rely on structural aggregation, and the trap becomes an especially attractive shortcut: W/TRAP yields markedly lower `diag_mean` and higher `to_other` than W/OTRAP throughout the stream (Fig. 1(a,b)). The oracle exhibits the same directional effect but with substantially smaller magnitude (Fig. 1(c,d)). This disparity supports our interpretation that bounded replay under restricted observability can *repeatedly reinforce* a time-local motif–label correlation, turning it into an overly confident decision rule that later becomes brittle.
- *When conflict arrives, failures become* structured *and directional.* When conflict tasks first appear ($k \geq 24$), errors do not diffuse arbitrarily; instead they concentrate along the induced–conflict axis. In Fig. 1, the directed block masses become non-zero and grow after conflict onset, with the strongest response under W/TRAP+RANDOM for the memory-limited learner (Fig. 1(e,f)). Moreover, the transition is typically asymmetric: $m_{\text{con}\to\text{ind}}$ (`c2i`) is larger than $m_{\text{ind}\to\text{con}}$ (`i2c`), suggesting that newly arriving conflict classes are systematically absorbed into the earlier induced decision regions that were shaped when the motif was highly predictive. The oracle shows the same qualitative transition signature but with weaker magnitude (Fig. 1(g,h)), consistent with richer historical evidence mitigating the collapse of a shortcut-based rule.

Together, Figs. 1 provides direct empirical evidence for *temporally illusory invariance* in memory-limited class incremental GCL. Specifically, a structural cue (triangle closure) that is only *temporally predictive* can be treated as if it were invariant under restricted observability: the learner begins to drift *before* conflict classes appear, and the later conflict phase exposes this false invariance as structured, directional misclassification flows. Therefore, the core issue is not merely late-stage conflict errors, but the *early formation and consolidation of a non-transferable structural rule*. This diagnosis motivates methods that explicitly discourage shortcut consolidation in memory and favor representations/prototypes that remain transferable under structural drift.

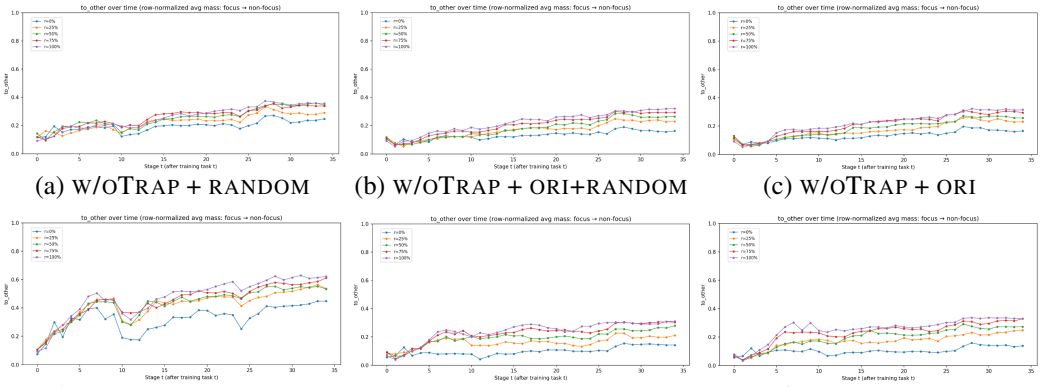
D DIAGNOSTIC STUDY OF HISTORY-AWARE CONNECTIVITY

D.1 INTER-TASK EDGE DOSE–RESPONSE STUDY

Setup. We define inter-task edges as $E_{\text{inter}} = \{(u, v) \in E : \tau(u) \neq \tau(v)\}$ and progressively restore a fraction $r \in \{0, 25, 50, 75, 100\}\%$.

Restoration policies. We consider random restoration and similarity-ordered restoration. Similarity scores are computed once on original attributes to fix the restoration order across feature regimes.

Results. Figures 2 and D1 report dose-response curves for within-focus correctness and focus-to-control leakage. Figure D2 further reports induced-conflict transition masses.



(a) W/OTRAP + RANDOM (b) W/OTRAP + ORI+RANDOM (c) W/OTRAP + ORI

(d) W/TRAP + RANDOM (e) W/TRAP + ORI+RANDOM (f) W/TRAP + ORI

Figure D1: **Inter-task edge recovery increases focus-to-control leakage.** We report $m_{\text{focus}\to\text{ctrl}}(k)$ (`to_other`), the mean row-normalized probability mass from focus classes $\mathcal{C}_{\text{focus}}$ to control classes $\mathcal{C}_{\text{ctrl}}$ after training task k (lower is better). From left to right, the columns correspond to RANDOM, ORI+RANDOM, and ORI. The top row reports W/OTRAP, while the bottom row reports W/TRAP. Curves vary the inter-task edge recovery rate $r \in \{0, 25, 50, 75, 100\}\%$ as in Fig. 2. Leakage increases with r, indicating stronger cross-class mixing as inter-task connectivity is restored.

D.2 INTER-TASK EDGES: STRUCTURAL CONTEXT VS. STRUCTURAL ISOLATION

The Motif Trap study isolates a time-local structural shortcut and shows that shortcut reliance can persist and later fail under progressive conflict. We next examine a complementary factor that is specific to graph streams: beyond an explicit replay buffer, historical information can also be conveyed *implicitly* through message passing over *inter-task edges*. In our class-incremental stream, tasks contain disjoint label sets.

Removing inter-task edges reduces cross-task message passing and can yield sharper decision boundaries by limiting cross-class neighborhood mixing, but it also induces *structural isolation* by restricting nodes to incomplete neighborhoods. This motivates a dose–response study that progressively restores inter-task edges to quantify how cross-task structural context trades off against cross-class interference and shortcut-driven signatures, and how this tradeoff interacts with the motif trap.

We study this question under two complementary settings: (i) an **oracle** dose–response restoration of inter-task edges, and (ii) a **context-visible** evaluation for memory-limited condensation/replay baselines under realistic cross-task connectivity.

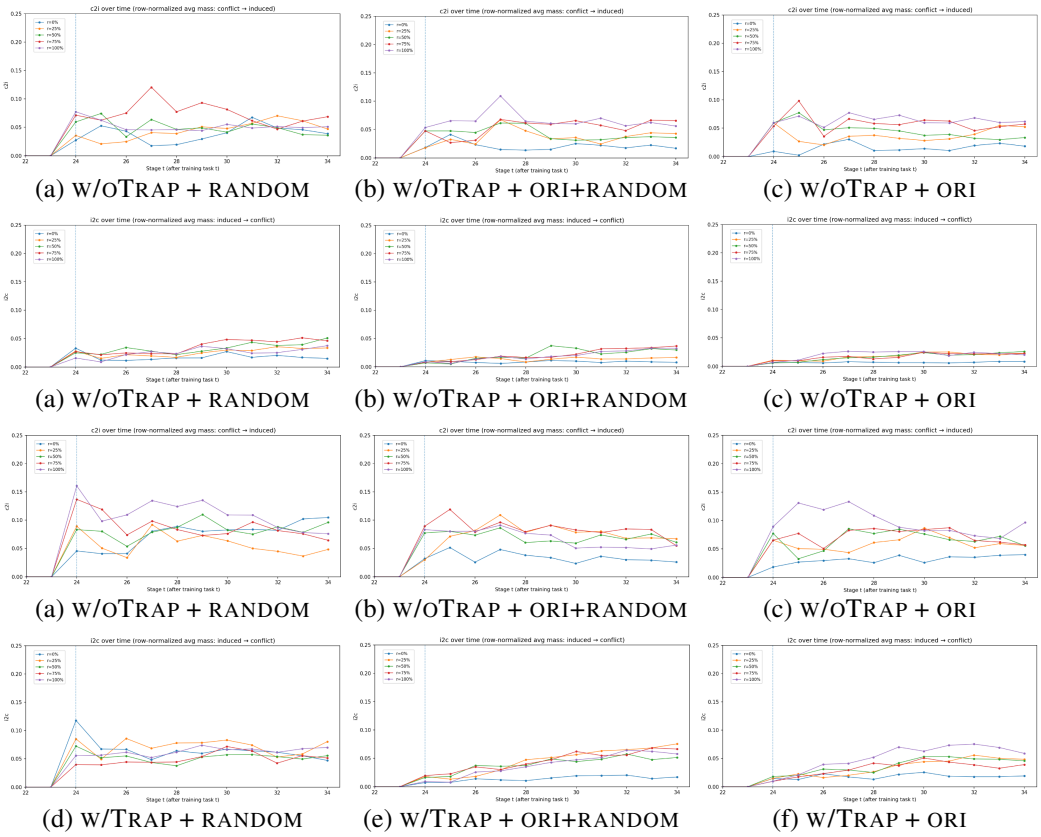
D.2.1 ORACLE: CONTROLLED INTER-TASK EDGE RESTORATION (DOSE–RESPONSE).

Let $\tau(v) \in \{0, \ldots, 34\}$ denote the task ID of node v under the class-incremental stream. We define the inter-task edge set

$$E_{\text{inter}} = \{(u, v) \in E : \tau(u) \neq \tau(v)\},$$

i.e., edges whose endpoints belong to different tasks. In the oracle (full-history) setting, we start from a *structurally isolated* graph where all inter-task edges are removed ($r = 0\%$), and progressively restore a fraction $r \in \{25\%, 50\%, 75, 100\}\%$ of E_{inter} to obtain a dose–response characterization of cross-task connectivity. For each r, we evaluate both W/TRAP and W/OTRAP, and reuse the same three feature regimes (ORI, RANDOM, ORI+RANDOM) as in Sec. 2.2.

Restoration policies. We consider two policies to select restored inter-task edges. **Random restoration** samples edges uniformly from E_{inter}. **Similarity-ordered restoration** restores edges from low to high feature similarity, where similarity is computed on the original attributes to keep the restoration order fixed across feature regimes. We use the original attributes only to compute the similarity scores that determine the restoration order; training under RANDOM/ORI+RANDOM still uses the corresponding randomized attributes.



(a) W/OTRAP + RANDOM (b) W/OTRAP + ORI+RANDOM (c) W/OTRAP + ORI

(a) W/OTRAP + RANDOM (b) W/OTRAP + ORI+RANDOM (c) W/OTRAP + ORI

(a) W/OTRAP + RANDOM (b) W/OTRAP + ORI+RANDOM (c) W/OTRAP + ORI

(d) W/TRAP + RANDOM (e) W/TRAP + ORI+RANDOM (f) W/TRAP + ORI

Figure D2: **Within-focus redistribution under inter-task edge recovery: induced–conflict transitions.** We report directed confusion block masses between the induced and conflict subsets within $\mathcal{C}_{\text{focus}} = \mathcal{C}^{\triangle}_{\text{ind}} \cup \mathcal{C}^{\triangle}_{\text{con}}$. The first and third rows show $\texttt{c2i} = m_{\text{con}\to\text{ind}}$, while the second and fourth rows show $\texttt{i2c} = m_{\text{ind}\to\text{con}}$. From left to right, the three columns correspond to W/OTRAP+RANDOM, W/OTRAP+ORI, and W/OTRAP+ORI+RANDOM in the top half, and W/TRAP+RANDOM, W/TRAP+ORI, and W/TRAP+ORI+RANDOM in the bottom half. Curves vary the inter-task edge recovery rate $r \in \{0, 25, 50, 75, 100\}\%$. Before conflict onset ($k < 24$), the corresponding transition mass is undefined and is shown as zero for visualization.

Scope. Restoration is applied only between original nodes in E_{inter}. Auxiliary nodes introduced by the Motif Trap remain locally attached to their source node (and optionally to each other) and never participate in cross-task connectivity. Thus, the intervention isolates the effect of inter-task connectivity among original nodes.

D.2.2 MEMORY-LIMITED BASELINES: CONTEXT-VISIBLE CROSS-TASK NEIGHBORHOOD.

For memory-limited condensation/replay methods, past tasks are stored only in a compressed form rather than as raw historical graphs. As a result, reproducing the oracle-style *global* edge-restoration schedule (varying an overall recovery ratio r over inter-task edges) is not well-defined without additional structural assumptions. Instead, we evaluate baselines under realistic cross-task connectivity using a unified *context-visible* protocol.

At step t, the input subgraph contains the current-task nodes together with any previously arrived historical nodes that are 1-hop connected to them in the original stream, and all original edges among these nodes. Importantly, we do *not* edit the topology: no edges are added, removed, or reweighted; we only expose the cross-task neighbors that are directly revealed by the observed graph. Historical nodes included at step t are treated as unlabeled and are excluded from supervision. For all baselines, we keep the original training objective and hyperparameters unchanged, and only replace the step-t input subgraph with this context-visible one.

D.2.3 EVALUATION AND DIAGNOSTIC RATIONALE.

We reuse the trap-specific diagnostics from Section 2.2—within-focus correctness ($\mathrm{Diag}_{\mathrm{focus}}$), focus-to-control leakage ($m_{\mathrm{focus}\to\mathrm{ctrl}}$), and induced–conflict transition masses ($m_{\mathrm{ind}\to\mathrm{con}}$, $m_{\mathrm{con}\to\mathrm{ind}}$). In the *oracle dose–response* setting, we vary the inter-task edge recovery rate $r \in \{0, 25, 50, 75, 100\}\%$. In the *memory-limited* setting, we compare two input protocols: a task-isolated subgraph and a context-visible subgraph that exposes all observable 1-hop historical neighbors of current-task nodes as unlabeled message-passing context.

In our class-incremental stream, tasks have disjoint label sets, so inter-task edges are cross-class by construction. Increasing cross-task connectivity can therefore provide additional context through message passing but can also intensify cross-class mixing. Changes in $\mathrm{Diag}_{\mathrm{focus}}$ and $m_{\mathrm{focus}\to\mathrm{ctrl}}$ serve as diagnostics for which effect dominates, with the clearest separation expected when attribute cues are suppressed (e.g., RANDOM) and/or when the motif trap is present.

D.2.4 ORACLE RESULTS (DOSE–RESPONSE).

Figures 2–D1 show a clear dose–response to inter-task edge recovery in the oracle setting. As the recovery rate r increases, within-focus correctness $\mathrm{Diag}_{\mathrm{focus}}$ (`diag_mean`) decreases (Fig. 2), while focus-to-control leakage $m_{\mathrm{focus}\to\mathrm{ctrl}}$ (`to_other`) increases (Fig. D1). Together, these trends indicate that, in this disjoint-class stream, restoring E_{inter} primarily strengthens cross-class message passing, reducing self-consistency within the focus set and shifting probability mass toward control classes.

The magnitude of this degradation exhibits a consistent interaction across conditions: it is mildest under W/OTRAP+ORI and strongest under W/TRAP+RANDOM, with ORI+RANDOM consistently intermediate (Figs. 2–D1). This mirrors the Motif Trap study: when attribute cues are suppressed, the learner relies more on structural signals, and increasing cross-task connectivity amplifies shortcut-driven brittleness by enlarging cross-class receptive fields. We report within-focus induced–conflict transitions (`c2i`/`i2c`) in Fig. D2; compared to the global mixing trends in Figs. 2–D1, their dependence on r is weaker and less monotonic.

E COMPLETE LEARNING PROCEDURE OF ISD-GC

Algorithm 1 provides the complete step-wise learning procedure of ISD-GC, including memory-grounded context disambiguation, structure-invariant prototype condensation, and the subsequent pure-memory update.

F IMPLEMENTATION DETAILS OF ISD-GC

This appendix provides implementation details omitted from the main text for brevity. We describe the parameterization, calibration, and optimization details of memory-grounded context disambiguation and structure-invariant prototype condensation.

F.1 IMPLEMENTATION DETAILS OF STAGE I

Reference encoder and anchor construction. The reference encoder f_{ref} is evaluated without gradient updates when constructing the VSE teacher. Historical prototypes in $\mathcal{M}_{t-1}$ are embedded as independent semantic anchors without message passing over the memory graph. For the current task, each class anchor is constructed by averaging the normalized embeddings of labeled nodes belonging to the corresponding newly arriving class. The labeled subset may be subsampled when constructing Ω_c. When valid current-task anchors are unavailable, we use the all-historical fallback in the context-identification procedure.

VSE retrieval-space parameterization. The VSE module uses two lightweight linear projections $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{d\times d}$ to map task embeddings and prototype anchors into a common retrieval space. In our implementation, $\boldsymbol{W}_Q$ and $\boldsymbol{W}_K$ are initialized as identity projections and kept fixed during Stage-I optimization. The reference encoder is also frozen, and the resulting VSE teacher signal $o(v)$ is detached from gradient optimization. Consequently, Stage-I discriminator learning cannot reduce its loss by modifying the teacher signal itself.

Algorithm 1 ISD-GC at step t (Connect-then-Condense with Pure-Memory Update)

Input: history-aware task graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ with adjacency $\boldsymbol{A}_t$ and features $\boldsymbol{X}_t$; labels on $\mathbb{V}_t^{\mathrm{tr}}$; new class set $\mathbb{C}_t$; memory $\mathcal{M}_{t-1}$; per-class budget $\mathbf{b} = \{b_c\}_{c \in \mathbb{C}_{\le t}}$.
Output: updated memory $\mathcal{M}_t$ and predictor for step t.
Stage I: Memory-grounded context identification
if $\mathcal{M}_{t-1} \neq \emptyset$ **then**
    Compute reference embeddings $\boldsymbol{H} = \mathrm{norm}(f_{\mathrm{ref}}(\boldsymbol{X}_t, \boldsymbol{A}_t))$ and memory anchor embeddings $\boldsymbol{U}_{\mathrm{old}}$ (independent anchors; no message passing on memory).
    Construct current-task anchors $\boldsymbol{U}_{\mathrm{new}}$ from labeled nodes in $\mathbb{V}_t^{\mathrm{tr}}$.
    Compute historical affinity $o(v)$ via VSE evidence gap.
    Train the context discriminator q_ϕ using VSE soft targets and reliable-negative anchoring (Eq. 1–2).
    Infer the historical-candidate mask $\mathbf{m}_t^{\mathrm{hist}}$ via quantile self-thresholding.
    Estimate task-specific utility for historical candidates and partition them into $\boldsymbol{m}_t^{\mathrm{ben}}$, $\boldsymbol{m}_t^{\mathrm{irr}}$, and $\boldsymbol{m}_t^{\mathrm{harm}}$.
else
    Set $\mathbf{m}_t^{\mathrm{hist}} = \boldsymbol{m}_t^{\mathrm{ben}} = \boldsymbol{m}_t^{\mathrm{irr}} = \boldsymbol{m}_t^{\mathrm{harm}} = \mathbf{0}$.
end if
Construct $(\boldsymbol{X}_t^{\mathrm{dis}}, \boldsymbol{A}_t^{\mathrm{dis}})$ by retaining beneficial context, down-weighting irrelevant context, and suppressing harmful context.
Construct $(\boldsymbol{X}_t^{\mathrm{hist}}, \boldsymbol{A}_t^{\mathrm{hist}})$ by retaining beneficial and irrelevant historical context while suppressing harmful context.
Stage II: Structure-invariant prototype condensation
Sample perturbation adjacency $\widetilde{\boldsymbol{A}}_t^{\mathrm{pert}}$ by random edge dropping from $\boldsymbol{A}_t^{\mathrm{dis}}$.
Compute multi-view teacher embeddings $\boldsymbol{H}^{\mathrm{dis}}, \boldsymbol{H}^{\mathrm{hist}}, \boldsymbol{H}^{\mathrm{pert}}$.
Optimize prototype features $\boldsymbol{P}_t$ under budgets $\mathbf{b}$ by minimizing $\mathcal{L}_{\mathrm{ISD}}$ (Eq. 3).
Commit optimized prototypes to memory $\mathcal{M}_t$ (independent nodes with self-loops) and merge into the memory bank under budget $\mathbf{b}$.
Pure-memory update: retrain the learner using only $\mathcal{M}_{\le t}$ to obtain the step-t predictor.

Context-discriminator features. The context discriminator operates on a fixed-dimensional, task-agnostic descriptor $\boldsymbol{z}_v$ computed from the currently observed task graph. The descriptor summarizes two groups of signals: (i) prediction statistics, including uncertainty, margin, and energy-related quantities derived from model logits over current and previously observed classes; and (ii) local structural-consistency statistics obtained by aggregating class-probability vectors over the observed neighborhood together with lightweight degree information. These features provide the discriminator with a fixed-dimensional input even as the number and composition of observed nodes change across tasks.

Reliable-anchor constraints. At task t, all labeled current-task nodes belong to newly arriving classes and are therefore treated as reliable non-context anchors:

$$\mathbb{N}_t = \mathbb{V}_t^{\mathrm{tr}}.$$

During soft-target construction, these nodes are assigned target zero even when their VSE affinity exceeds the teacher threshold. After inference, we likewise enforce

$$\mathbf{m}_t^{\mathrm{hist}}(v) = 0, \qquad \forall v \in \mathbb{N}_t.$$

This hard anchor constraint prevents current-task labeled nodes from being incorrectly incorporated into the historical-context set.

Utility probe and adaptive typing. To estimate task-specific utility, we construct one-hop representations for labeled current-task nodes and fit a lightweight linear probe using only current-task supervision. After fitting the probe, its parameters are frozen before evaluating the utility score $u_t(v) = -\left\langle \frac{\partial \mathcal{L}_{\mathrm{probe}}}{\partial \boldsymbol{h}_v}, \boldsymbol{h}_v \right\rangle$. The utility threshold τ_u is determined adaptively from the magnitude distribution of non-zero historical-candidate utility scores. Nodes whose utility exceeds τ_u are treated as beneficial, those below $-\tau_u$ as harmful, and the remaining historical candidates as irrelevant.

Algorithm 2 Stage I: Memory-grounded Context Identification

Input: history-aware task graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ with adjacency $\boldsymbol{A}_t$ and features $\boldsymbol{X}_t$; labeled new-class nodes $\mathbb{V}_t^{\mathrm{tr}}$; memory $\mathcal{M}_{t-1}$; reference encoder f_{ref}; VSE parameters $(\boldsymbol{W}_Q, \boldsymbol{W}_K, k, \tau_{\mathrm{ctx}})$; quantiles (q_{tea}, q).
Output: context scores $\mathbf{r}_t$; historical-candidate mask $\mathbf{m}_t^{\mathrm{hist}}$; typed masks $\boldsymbol{m}_t^{\mathrm{ben}}$, $\boldsymbol{m}_t^{\mathrm{irr}}$, $\boldsymbol{m}_t^{\mathrm{harm}}$; disambiguated view $(\boldsymbol{X}_t^{\mathrm{dis}}, \boldsymbol{A}_t^{\mathrm{dis}})$ and history-aware topology $\boldsymbol{A}_t^{\mathrm{hist}}$.
Stage I: Memory-grounded context identification
Initialize $\boldsymbol{A}_t^{\mathrm{hist}} \leftarrow \boldsymbol{A}_t$ and reliable non-context anchors $\mathbb{N}_t \leftarrow \mathbb{V}_t^{\mathrm{tr}}$.
if $\mathcal{M}_{t-1} = \emptyset$ **then**
  Set $\mathbf{m}_t^{\mathrm{hist}} = \boldsymbol{m}_t^{\mathrm{ben}} = \boldsymbol{m}_t^{\mathrm{irr}} = \boldsymbol{m}_t^{\mathrm{harm}} = \mathbf{0}$ and $(\boldsymbol{X}_t^{\mathrm{dis}}, \boldsymbol{A}_t^{\mathrm{dis}}) \leftarrow (\boldsymbol{X}_t, \boldsymbol{A}_t)$.
else
  Compute reference embeddings $\boldsymbol{H} = \mathrm{norm}(f_{\mathrm{ref}}(\boldsymbol{X}_t, \boldsymbol{A}_t))$ and construct historical memory anchors $\boldsymbol{U}_{\mathrm{old}}$ and current-task anchors $\boldsymbol{U}_{\mathrm{new}}$.
  Compute historical affinity $o(v)$ via top-k VSE retrieval and relative evidence gap.
  Obtain teacher threshold δ_{tea} via quantile calibration on reliable anchors and construct candidate context set $\mathbb{P}_t$.
  Form VSE soft targets with anchor override: $y_v^{\mathrm{tea}} = o(v)$ for $v \in \mathbb{P}_t \setminus \mathbb{N}_t$ and $y_v^{\mathrm{tea}} = 0$ for $v \in \mathbb{N}_t$ (Eq. 1).
  Train context discriminator q_ϕ using anchored BCE (Eq. 2) and infer context scores $r_t(v) = q_\phi(\boldsymbol{z}_v)$.
  Obtain historical-candidate mask $\mathbf{m}_t^{\mathrm{hist}}$ via quantile self-thresholding and enforce $\mathbf{m}_t^{\mathrm{hist}}(v) = 0$ for $v \in \mathbb{N}_t$.
  Estimate first-order task utility for historical candidates and partition them into $\boldsymbol{m}_t^{\mathrm{ben}}$, $\boldsymbol{m}_t^{\mathrm{irr}}$, and $\boldsymbol{m}_t^{\mathrm{harm}}$.
end if
Construct $(\boldsymbol{X}_t^{\mathrm{dis}}, \boldsymbol{A}_t^{\mathrm{dis}})$ by retaining beneficial context, down-weighting irrelevant context, and suppressing harmful context.
Construct $(\boldsymbol{X}_t^{\mathrm{hist}}, \boldsymbol{A}_t^{\mathrm{hist}})$ by retaining the observable history-aware topology while suppressing harmful historical context.
return $\mathbf{r}_t, \mathbf{m}_t^{\mathrm{hist}}, \boldsymbol{m}_t^{\mathrm{ben}}, \boldsymbol{m}_t^{\mathrm{irr}}, \boldsymbol{m}_t^{\mathrm{harm}}, \boldsymbol{X}_t^{\mathrm{dis}}, \boldsymbol{A}_t^{\mathrm{dis}}, \boldsymbol{A}_t^{\mathrm{hist}}$

Algorithmic summary and outputs to Stage II Algorithm 2 summarizes the implementation-critical Stage-I procedure. In brief, Stage I first builds a VSE-based relative-evidence teacher from memory prototypes and current-task labeled prototypes, then distills this signal into a lightweight context discriminator using reliable non-context anchors, and finally produces a task-adaptive context mask $\mathbf{m}_t^{\mathrm{hist}}$ via quantile calibration on the anchor set. For the detected historical candidates, Stage I further estimates their task-specific utility and partitions them into beneficial, irrelevant, and harmful context, represented by $\mathbf{m}_t^{\mathrm{ben}}$, $\mathbf{m}_t^{\mathrm{irr}}$, and $\mathbf{m}_t^{\mathrm{harm}}$, respectively.

Given $\mathbf{m}_t^{\mathrm{hist}}$, $\mathbf{m}_t^{\mathrm{ben}}$, $\mathbf{m}_t^{\mathrm{irr}}$, and $\mathbf{m}_t^{\mathrm{harm}}$, Stage I outputs the inputs required by Stage II (§3.2) through mask-guided feature/edge gating. Specifically, we retain beneficial context, down-weight irrelevant context, and suppress harmful context in both node features and incident edges to obtain the *Disambiguated* feature matrix $\boldsymbol{X}_t^{\mathrm{dis}}$ and adjacency $\boldsymbol{A}_t^{\mathrm{dis}}$. We also retain the original history-aware adjacency as $\boldsymbol{A}_t^{\mathrm{hist}} := \boldsymbol{A}_t$. These outputs instantiate the Disambiguated and History-aware views used in Stage II, and reduce contamination from cross-task neighborhood mixing during prototype condensation.

F.2 Implementation Details of Stage II

Teacher encoder. The same frozen encoder used as f_{ref} in Stage I is reused as the teacher encoder Enc in Stage II; the notation is separated only to distinguish their roles across the two stages. All real-node representations $\boldsymbol{H}^{\mathrm{dis}}$, $\boldsymbol{H}^{\mathrm{hist}}$, and $\boldsymbol{H}^{\mathrm{pert}}$ are treated as fixed teacher targets during prototype optimization.

Prototype encoding. Prototype nodes are embedded independently using $\mathrm{Enc}_{\mathrm{noedge}}$. Thus, their embeddings are generated without message passing over historical raw graphs or unbounded replay

structures. Only the newly allocated prototype features $\boldsymbol{P}_t$ for classes $c \in \mathbb{C}_t$ are optimized at step t; previously stored prototypes remain unchanged.

Structural views. The Disambiguated view applies the Stage-I typed masks to node features and incident edges: beneficial and non-historical context is retained, irrelevant historical context is down-weighted, and harmful context is suppressed. The History-aware view restores the observable history-aware topology $\boldsymbol{A}_t^{\mathrm{hist}} = \boldsymbol{A}_t$ while suppressing harmful historical features. The Perturbation view is generated by applying random structural perturbation to $\boldsymbol{A}_t^{\mathrm{dis}}$ while retaining $\boldsymbol{X}_t^{\mathrm{dis}}$.

Gradient flow. During structure-invariant prototype condensation, the teacher encoder and all real-node teacher targets are fixed. Gradients from $\mathcal{L}_{\mathrm{ISD}}$ are propagated only to the prototype parameters $\boldsymbol{P}_t$. This separation prevents the teacher representations from adapting to the prototype objective and ensures that optimization distills the cross-view structural signal into the bounded prototype memory.

Memory update. After optimizing $\boldsymbol{P}_t$, the resulting prototypes are committed to the memory bank according to the per-class allocation $\mathbf{b}$. Previously stored prototypes are preserved, while the prototypes for the current newly arriving classes are added under their designated budgets. The step-t predictor is subsequently obtained using only the accumulated prototype memory under the pure-memory protocol.

G Computational Cost Discussion

ISD-GC introduces additional operations compared with standard condensation-based replay methods, including memory-grounded context identification and multi-view structure-invariant prototype distillation. This section discusses the practical computational overhead introduced by these components.

G.1 Memory-grounded Context Identification

The context identification stage operates under the restricted-observability setting and does not require access to raw historical graphs. Instead, the Virtual Soft Edge (VSE) module retrieves relative evidence between current task nodes and the stored prototype memory M_{t-1}. Since the prototype memory size is bounded by the predefined memory budget, the retrieval operation depends on the current task size and the allocated number of historical prototypes rather than the total number of nodes observed throughout the entire stream. Moreover, the context discriminator is trained using fixed-dimensional task-agnostic descriptors and lightweight classification objectives, introducing only a small additional overhead.

G.2 Prototype Distillation Overhead

The structure-invariant prototype condensation stage constructs several structural views, including the disambiguated view, history-aware view, and perturbation view. Although multiple views require additional forward passes through the frozen teacher encoder, the number of views remains fixed and does not increase with the length of the task stream. Furthermore, ISD-GC optimizes prototypes only for newly arriving classes at each incremental step. Previously stored prototypes remain unchanged and are directly reused in subsequent tasks. Therefore, the optimization cost is determined by the number of newly allocated prototypes and the optimization iterations, rather than the size of the accumulated historical data.

G.3 Overall Discussion

Overall, ISD-GC trades a moderate additional optimization cost for more transferable condensed memories. Importantly, the introduced components do not require storing or repeatedly processing the full historical graph. Instead, their overhead is bounded by the prototype memory budget and the current task scale, preserving the practical advantage of condensation-based replay under strict memory constraints.

H Experimental Details

H.1 Dataset Statistics

Table 4 summarizes the statistics of the four node-classification benchmarks used in our experiments. Following the task construction of CGLB Zhang et al. (2022a), each dataset is divided into a sequence of class-incremental tasks with two classes per task.

Table 4: Statistics of the datasets.

Dataset	Nodes	Edges	Features	Classes	Tasks
CoraFull	19,793	130,622	8,710	70	35
ogbn-arxiv	169,343	1,166,243	128	40	20
Reddit	227,853	114,615,892	602	40	20
ogbn-products	2,449,028	61,859,036	100	46	23

H.2 Metric Definitions

We report **Average Accuracy (AvgAcc)**, **Last Accuracy (LastAcc)**, and **Forgetting**. Let $\mathbf{M} \in \mathbb{R}^{K\times K}$ denote the performance matrix, where $R_{i,j}$ denotes the classification accuracy on task T_j after learning task T_i. After learning task T_k, AvgAcc measures the average performance over all tasks observed up to the current step:

$$\mathrm{AvgAcc}_k = \frac{1}{k}\sum_{i=1}^{k} R_{k,i}. \tag{5}$$

LastAcc further summarizes the average performance over the continual-learning trajectory up to task T_k:

$$\mathrm{LastAcc}_k = \frac{1}{k}\sum_{i=1}^{k} \mathrm{AvgAcc}_i. \tag{6}$$

Forgetting quantifies the degradation of previously learned tasks after subsequent task learning:

$$\mathrm{Forgetting}_k = \frac{1}{k-1}\sum_{i=1}^{k-1} \left(R_{i,i} - R_{k,i}\right). \tag{7}$$

Higher AvgAcc and LastAcc indicate better continual-learning performance, whereas lower Forgetting indicates better retention of previously acquired knowledge. Unless otherwise stated, the backbone architecture, optimization budget, and model-selection protocol are matched across methods for fairness.

I Additional Experimental Results

I.1 Results under the Isolated-control Protocol

Table 5 shows the overall performance under the isolated-control protocol. Compared with the history-aware protocol, this setting provides a structurally simplified control by eliminating cross-task neighborhood mixing and restricting each step to current-task nodes and intra-task edges. ISD-GC can match the Joint method with a 0.005 budget ratio. This indicates that the improvement of ISD-GC does not arise solely from exploiting observable historical neighbors or cross-task connectivity. Instead, the structure-invariant condensation mechanism itself produces compact and transferable memory that remains effective even under the task-isolated setting.

Overall, the results answer RQ2 positively: ISD-GC is not specifically tied to the history-aware protocol, but retains strong continual-learning performance when cross-task context is unavailable. This further supports the general effectiveness of the proposed condensation strategy beyond the newly introduced history-aware setting.

Table 5: Overall class-incremental performance under the isolated-control protocol.Best results are shown in **bold**.

Method	CORAFULL			OGBN-ARXIV		
	AvgAcc↑	LastAcc↑	Forget.↓	AvgAcc↑	LastAcc↑	Forget.↓
Fine-tune	2.540±0.000	11.020±0.050	96.123±0.079	4.950±0.000	17.395±0.035	96.670±0.096
GEM	2.540±0.000	10.982±0.042	95.741±0.032	4.955±0.004	17.435±0.012	96.814±0.045
EWC	2.839±0.265	11.199±0.097	95.167±0.419	4.953±0.011	17.411±0.022	96.398±0.575
LWF	2.413±0.174	10.998±0.074	95.709±0.227	4.955±0.004	17.404±0.031	96.698±0.029
MAS	3.447±0.429	12.111±0.451	93.725±0.698	4.955±0.008	17.293±0.050	95.679±0.192
TWP	16.278±3.614	23.246±1.621	66.043±2.934	3.885±2.416	17.645±1.434	58.288±9.592
ER-GNN	2.871±0.671	10.248±0.645	44.037±3.298	49.114±1.027	56.290±1.443	17.275±1.264
SSM	2.878±0.591	11.023±0.068	94.670±0.377	29.674±1.492	40.594±1.131	69.621±1.533
CAT	67.517±1.131	78.018±0.911	7.809±0.553	66.494±0.587	73.957±0.444	12.429±0.688
PUMA	77.722±0.443	84.989±0.423	4.387±0.357	69.508±0.403	76.420±0.182	10.705±0.496
Joint-train	81.130±0.433	85.506±0.265	2.491±0.216	73.046±0.122	79.534±0.145	9.209±0.130
ISD-GC	**81.176±0.149**	**87.164±0.083**	**3.200±0.463**	**73.934±0.128**	**79.798±0.110**	**8.008±0.340**

Method	OGBN-PRODUCTS			REDDIT		
	AvgAcc↑	LastAcc↑	Forget.↓	AvgAcc↑	LastAcc↑	Forget.↓
Fine-tune	4.197±0.000	15.852±0.031	96.341±0.188	4.989±0.002	18.183±0.160	99.521±0.006
GEM	4.197±0.000	15.855±0.010	96.418±0.198	4.986±0.003	18.147±0.026	99.561±0.016
EWC	4.222±0.046	16.022±0.074	96.495±0.273	5.166±0.456	18.690±1.022	99.182±0.497
LWF	4.203±0.014	15.883±0.028	96.705±0.162	4.986±0.004	17.847±2.437	98.683±1.819
MAS	4.932±1.645	16.028±0.214	95.382±1.651	4.949±0.004	18.850±1.016	99.344±0.042
TWP	2.823±1.195	17.605±1.357	51.210±6.228	2.096±1.485	22.007±2.037	52.542±4.523
ER-GNN	39.613±0.489	36.663±0.556	48.258±0.938	35.688±2.672	49.607±0.524	67.135±2.807
SSM	62.513±0.402	62.636±0.215	22.645±0.401	57.370±3.734	71.938±1.123	44.335±3.945
CAT	70.125±0.510	82.374±0.184	4.971±0.234	97.181±0.068	98.294±0.035	0.565±0.057
PUMA	73.985±0.351	84.893±0.050	3.308±0.397	97.995±0.045	98.847±0.024	0.292±0.037
Joint-train	85.332±0.938	85.906±0.571	2.721±0.096	98.457±0.014	99.093±0.012	0.330±0.013
ISD-GC	**77.607±0.054**	**86.933±0.054**	**2.883±0.437**	**98.006±0.014**	**99.036±0.005**	**0.174±0.026**

I.2 ADDITIONAL RESULTS UNDER STRUCTURAL DRIFT

Table 6 reports the remaining RQ3 results on OGBN-ARXIV, OGBN-PRODUCTS, and REDDIT. ISD-GC consistently improves AvgAcc and LastAcc over PUMA across all evaluated conditions, while reducing Forgetting in most cases. Under W/TRAP+ORI, the AvgAcc gains are 9.523, 3.174, and 2.974 points, respectively; under the more structure-dominant W/TRAP+RANDOM setting, the gains are 6.957, 11.509, and 11.135 points. These results are consistent with the main-text observation that structure-invariant condensation improves memory transferability when time-local structural cues become less reliable across tasks.

I.3 COMPLETE ABLATION RESULTS

Table 8 reports the ablation results on OGBN-ARXIV, OGBN-PRODUCTS, and REDDIT. The overall trends are consistent with those on CORAFULL: removing either Stage I or Stage II generally degrades performance, while the complete model achieves the highest AvgAcc and lowest Forgetting across the three datasets. The gains are more pronounced on OGBN-PRODUCTS and more modest on OGBN-ARXIV and REDDIT, indicating that the relative contributions of the two stages vary with the underlying graph stream.

I.4 PARAMETER SENSITIVITY

We further analyse the sensitivity of ISD-GC to several major hyperparameters, including the VSE top-k, the teacher quantile q_{tea}, the inference quantile q, the history-view weight λ_{hist}, and the drop-edge rate. Table 7 reports the corresponding sensitivity results. For each parameter, we vary only its value while keeping the remaining parameters fixed at the default configuration. The results show

Table 6: Additional results under structural drift and shortcut-prone regimes (RQ3). Results on CORAFULL are reported in the main text.

		OGBN-ARXIV		
Setting	Method	AvgAcc↑	LastAcc↑	Forget.↓
w/oTrap, ori	PUMA	71.752±0.237	73.805±0.016	9.855±0.387
	ISD-GC	**75.886±0.287**	**79.613±0.208**	**9.032±0.227**
w/Trap, ori	PUMA	52.491±0.364	61.139±0.333	9.467±0.317
	ISD-GC	**62.014±0.243**	**71.209±0.465**	**8.471±0.846**
w/oTrap, random	PUMA	14.675±0.781	25.507±0.135	15.790±1.265
	ISD-GC	**23.884±0.891**	**33.828±0.150**	**15.739±1.401**
w/Trap, random	PUMA	9.511±0.558	21.474±0.262	14.081±0.552
	ISD-GC	**16.468±0.524**	**29.335±0.177**	**13.951±1.025**
		OGBN-PRODUCTS		
Setting	Method	AvgAcc↑	LastAcc↑	Forget.↓
w/oTrap, ori	PUMA	70.932±0.242	83.441±0.010	9.717±0.265
	ISD-GC	**80.784±0.177**	**87.417±0.039**	**4.407±0.301**
w/Trap, ori	PUMA	73.781±0.147	83.259±0.054	5.420±0.186
	ISD-GC	**76.955±0.244**	**86.170±0.100**	**2.684±0.551**
w/oTrap, random	PUMA	21.284±0.892	21.233±0.153	11.996±0.644
	ISD-GC	**31.123±1.857**	**32.011±0.376**	**11.243±2.018**
w/Trap, random	PUMA	18.247±0.520	20.939±0.276	11.948±0.307
	ISD-GC	**29.756±0.987**	**31.630±0.273**	**11.272±1.067**
		REDDIT		
Setting	Method	AvgAcc↑	LastAcc↑	Forget.↓
w/oTrap, ori	PUMA	97.327±0.101	94.957±0.001	0.398±0.061
	ISD-GC	**97.905±0.061**	**98.991±0.021**	**0.271±0.108**
w/Trap, ori	PUMA	94.146±0.135	97.300±0.061	1.349±0.148
	ISD-GC	**97.120±0.027**	**98.431±0.029**	**0.562±0.057**
w/oTrap, random	PUMA	93.104±0.066	95.817±0.036	0.910±0.038
	ISD-GC	**95.824±0.059**	**97.543±0.022**	**0.434±0.056**
w/Trap, random	PUMA	58.827±0.287	70.220±0.065	5.386±0.302
	ISD-GC	**69.962±0.203**	**81.275±0.067**	**4.785±0.210**

that ISD-GC maintains relatively stable performance over a reasonable range of parameter values, indicating that the proposed method is not overly sensitive to hyperparameter selection.

Table 7: Sensitivity analysis of key hyperparameters.

Hyperparameter	Values	Chosen value	AvgAcc range
VSE top-k	[2, 4, 8, 16]	8	[81.440, 81.549]
q_{tea}	[0.90, 0.95, 0.98]	0.98	[81.294, 81.549]
q	[0.90, 0.95, 0.98]	0.98	[81.161, 81.549]
λ_{hist}	[0.25, 0.5, 1.0, 2.0]	0.5	[81.259, 81.549]
Drop-edge rate	[0.1, 0.2, 0.3, 0.5]	0.3	[81.238, 81.549]

Table 8: Complete ablation results of the two stages of ISD-GC (RQ4). Results on CORAFULL are reported in the main text.

OGBN-ARXIV				
Stage I	Stage II	AvgAcc↑	LastAcc↑	Forget.↓
×	×	75.725±0.313	79.802±0.256	8.966±0.400
×	✓	75.761±0.246	79.872±0.147	8.833±0.289
✓	×	75.789±0.230	79.855±0.259	8.864±0.255
✓	✓	**75.846±0.269**	**80.651±0.170**	**8.821±0.243**
OGBN-PRODUCTS				
Stage I	Stage II	AvgAcc↑	LastAcc↑	Forget.↓
×	×	82.244±0.370	87.928±0.067	3.703±0.319
×	✓	82.531±0.089	87.911±0.036	3.583±0.380
✓	×	83.024±0.351	87.873±0.065	3.522±0.327
✓	✓	**83.261±0.518**	**88.870±0.059**	**3.406±0.638**
REDDIT				
Stage I	Stage II	AvgAcc↑	LastAcc↑	Forget.↓
×	×	97.972±0.119	98.042±0.007	0.337±0.098
×	✓	97.999±0.088	99.044±0.010	0.352±0.090
✓	×	98.003±0.086	**99.059±0.017**	0.256±0.140
✓	✓	**98.009±0.113**	**99.059±0.012**	**0.255±0.127**