HERO: PRESERVING STRUCTURE AND SEMANTICS IN HETEROGENEOUS CONTINUAL GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine learning on heterogeneous graphs has advanced rapidly, driven by the diverse entities, relations, and semantics inherent in real-world data. However, most existing studies assume static graphs, whereas real-world graph data are often continuously updated. Continual learning in this setting is particularly challenging because historical knowledge is encoded not only in target-node representations, but also in their heterogeneous structural contexts and relation-dependent semantic responses. To this end, we introduce HERO, a heterogeneous continual graph learning framework that preserves historical knowledge at both structural and semantic levels while adapting to incoming tasks. We analyze experience replay as an approximation to the unavailable historical gradient and decompose its error into target-selection and context-reconstruction terms. This analysis shows that replaying target nodes alone is generally insufficient for relation-sensitive HGNNs, as removing typed neighborhood context can alter both representations and historical gradients. Motivated by this, HERO employs DiSCo (Diversity Sampling with semantic Consistency) to select representative target nodes and reconstruct compact heterogeneous contexts through relation-aware multi-type neighbor expansion. To further preserve knowledge that cannot be captured by replayed subgraphs alone, HERO aligns task-conditioned predictions and heterogeneous semantic responses through multi-level knowledge distillation. A lightweight look-ahead adaptation step is additionally used to improve plasticity on incoming tasks. Experiments on four datasets with three HGNN backbones show that HERO achieves the best or tied-best average performance among strong continual graph learning baselines while maintaining competitive forgetting. Further analyses validate the roles of structural context preservation and semantic alignment in balancing continual adaptation and retention. Our code is available at https://anonymous.4open.science/r/HCGL-B37E/.

1 INTRODUCTION

Heterogeneous graphs model complex real-world systems through multiple node and relation types (Shi et al., 2018), while heterogeneous graph neural networks (HGNNs) learn expressive representations by capturing their type-specific structural and semantic interactions (Sun & Han, 2012; Wang et al., 2019; Hu et al., 2020). Most existing HGNNs, however, are developed under static settings, whereas many real-world applications require models to adapt to continuously evolving heterogeneous graphs without retaining full historical observations. This gives rise to heterogeneous continual graph learning (HCGL), where an HGNN must adapt to incoming tasks while preserving knowledge acquired from previous ones.

Directly applying existing continual learning methods to HCGL overlooks how historical knowledge is distributed across heterogeneous graph computation. Existing replay methods mainly preserve selected instances, while distillation methods typically constrain model outputs. However, an HGNN prediction also depends on typed neighborhoods and relation-specific semantic interactions, which can change even when the target nodes or output predictions are preserved. Consequently, preserving a target node does not fully reconstruct the historical computation that produced its prediction, while preserving its output does not guarantee that the underlying relation-dependent semantics remain unchanged. We argue that effective HCGL therefore requires preserving historical knowledge at both the structural and semantic levels.

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.

We analyze the structural challenge from an optimization perspective. When past data are unavailable, experience replay estimates the historical gradient using a limited-budget memory buffer. We characterize the resulting increase in historical risk through the gradient approximation error, which can be decomposed into two sources: target-selection error, caused by unrepresentative replay samples of past data, and context-reconstruction error, caused by incomplete heterogeneous neighborhoods. For relation-sensitive HGNNs, we further show that target-node-only replay is generally insufficient to recover the historical prediction function, since the same target nodes can induce different representations under different typed neighborhoods. Effective heterogeneous replay must therefore preserve both representative targets and the structural contexts needed to recover their historical predictions.

Based on this observation, we propose **HERO**, a framework for preserving heterogeneous structural contexts and semantic responses in continual graph learning. HERO first introduces **DiSCo** (Diversity Sampling with semantic Consistency), which reduces replay approximation error through representative target-node selection and relation-aware multi-type neighbor expansion, producing compact replay subgraphs that better reconstruct historical heterogeneous contexts. Yet context reconstruction does not constrain how the model uses these contexts: the roles of individual relations and semantic channels may still drift as new tasks are learned. HERO addresses this problem with **HKD** (Heterogeneity-aware Knowledge Distillation), which aligns the current model with historical teachers at two levels: task-conditioned predictions and heterogeneous semantic responses. A lightweight look-ahead step adapts the model to a representative subset of the incoming task, improving plasticity to new knowledge.

Our main contributions are summarized as follows:

1. We provide an optimization view of heterogeneous replay, showing that replay-based retention depends on approximating the unavailable historical gradient. We decompose this approximation error into target-selection and heterogeneous-context reconstruction terms, and establish that target-node-only replay is generally insufficient for relation-sensitive HGNNs.
2. Guided by this analysis, we develop HERO to preserve historical knowledge at both structural and semantic levels. DiSCo reconstructs compact heterogeneous replay contexts, while HKD aligns task-conditioned predictions and relation-dependent semantic responses across learning stages.
3. We evaluate HERO on four heterogeneous graph benchmarks with three representative HGNN backbones and different task granularities. HERO achieves the best or tied-best average performance among strong continual graph learning baselines while maintaining competitive forgetting. Extensive analyses validate the roles of heterogeneous context reconstruction and semantic preservation.

2 PRELIMINARIES

We study continual node classification over a growing heterogeneous graph. The graph is revealed as a sequence of snapshots, where each snapshot is associated with the arrival of a new task. At each step, the learner receives the data of the current task and may retain only a bounded memory buffer from previous tasks. The objective is to adapt a shared HGNN to the incoming task while preserving its performance on all previously learned tasks.

Definition 1 (Growing Heterogeneous Graph). A heterogeneous graph snapshot at step t is denoted by $\mathcal{G}_t = (V_t, E_t, \mathcal{X}_t, \phi, \psi)$, where V_t and E_t are the node and edge sets, respectively, $\phi : V_t \to \mathcal{A}$ maps each node to a node type, and $\psi : E_t \to \mathcal{R}$ maps each edge to a relation type. The graph is heterogeneous if $|\mathcal{A}| + |\mathcal{R}| > 2$. For each node type $\tau \in \mathcal{A}$, let $V_{\tau,t} = \{v \in V_t \mid \phi(v) = \tau\}$ and let $X_{\tau,t} \in \mathbb{R}^{|V_{\tau,t}| \times d_\tau}$ denote its type-specific feature matrix. We consider a fixed heterogeneous schema consisting of $\mathcal{A}$ and $\mathcal{R}$, while the graph grows through newly introduced node and edge instances. In particular, consecutive snapshots satisfy $V_{t-1} \subseteq V_t$ and $E_{t-1} \subseteq E_t$ for $t = 2, \ldots, T$.

Definition 2 (Heterogeneous Continual Graph Learning). Each snapshot $\mathcal{G}_t$ introduces a node classification task $\mathcal{T}_t$ with task-specific classes. The tasks are class-disjoint, and each task contains K classes. Its target nodes are divided into training and test sets V_t^{tr} and V_t^{te}, respectively. All tasks sequentially update the same HGNN encoder h_φ and the K-way classifier g_ω, i.e., $f_\theta(v, \mathcal{G}_t) = g_\omega(h_\varphi(v, \mathcal{G}_t))$ with $\theta = (\varphi, \omega)$. At evaluation, the task context t is known to the evaluator and is

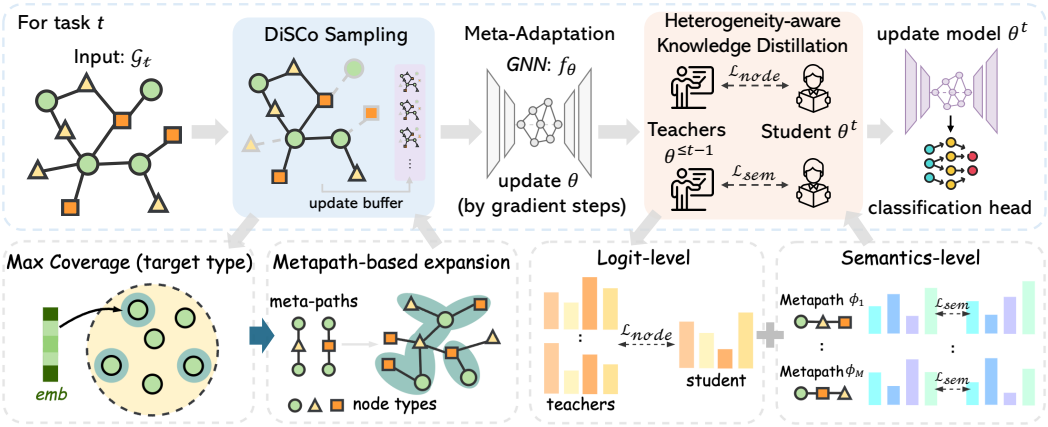


Figure 1: Overview of HERO for heterogeneous continual graph learning. For each incoming task, DiSCo constructs a compact replay subgraph by selecting representative target nodes and retaining their relation-aware multi-type neighborhoods, thereby preserving historical heterogeneous contexts. HKD complements replay by aligning task-conditioned predictions and heterogeneous semantic responses with historical teachers. A look-ahead meta-adaptation step further improves new-task plasticity under the joint current-task, replay, and distillation objectives.

used only to interpret the local prediction; it is not used to select task-specific model parameters. At step t, the model may retain a replay buffer $\mathcal{B}$ satisfying $|\mathcal{B}| \leq B$ (B is the buffer size).

3 Methodology

In this section, we present the HERO framework (see Figure 1). HERO addresses continual learning from two complementary aspects: retaining historical heterogeneous knowledge and maintaining plasticity to incoming tasks. We first introduce DiSCo to preserve representative heterogeneous contexts, and then present HKD to retain historical prediction and semantic responses beyond explicit replay. Finally, we describe a lightweight look-ahead adaptation mechanism that improves new-task plasticity under these historical constraints.

3.1 Diversity Sampling with Semantic Consistency

Experience replay provides a direct way to retain historical knowledge, but conventional replay strategies are not designed to preserve the heterogeneous computation contexts of HGNNs. In particular, storing representative target nodes alone may discard the typed neighborhoods and relation-specific interactions required to reproduce their historical training signals. To address this, we propose DiSCo (Diversity Sampling with semantic Consistency), which jointly considers target-node diversity and heterogeneous structural context. DiSCo proceeds in two stages: (1) selecting representative target-type nodes by maximizing diversity in the feature or embedding space, and (2) expanding to other types of nodes through relation-type-aware importance estimation, ensuring that both semantic diversity and heterogeneous structural patterns are preserved.

Replay quality as historical-gradient approximation. We first view experience replay as an approximation to the unavailable historical training objective. Let $F_{<t}(\theta)$ denote the full-data risk over tasks observed before $\mathcal{T}_t$. We define

$$g_o = \nabla_\theta F_{<t}(\theta), \qquad g_t = \nabla_\theta \mathcal{L}_{\mathcal{T}_t}(\mathcal{G}_t, \theta), \qquad g_{\mathcal{B}} = \nabla_\theta \mathcal{L}_{\text{er}}(\mathcal{G}_{\text{sub}}, \theta).$$

where $g_{\mathcal{B}}$ denote the gradient induced by the replay memory. A replay-based update can be written as

$$\theta^+ = \theta - \rho\,(g_t + \lambda_{\text{er}} g_{\mathcal{B}})\,,$$

where ρ is the learning rate and λ_{er} is the trade-off coefficient.

Proposition 1 (One-step historical-risk bound) *Suppose $F_{<t}$ is L_F-smooth and define the replay-gradient approximation error as*

$$\epsilon_{\mathcal{B}} = \|g_o - g_{\mathcal{B}}\|_2.$$

Then

$$\begin{aligned} F_{<t}(\theta^+) - F_{<t}(\theta) \leq & -\rho\langle g_o, g_t\rangle \\ & -\rho\lambda_{\text{er}}\left(\|g_o\|_2^2 - \|g_o\|_2\epsilon_{\mathcal{B}}\right) \\ & +\frac{L_F\rho^2}{2}\|g_t + \lambda_{\text{er}} g_{\mathcal{B}}\|_2^2 . \end{aligned}$$

The first term captures current–historical gradient interference, while the second term shows that replay imposes a stronger historical constraint when $g_{\mathcal{B}}$ better approximates g_o. Therefore, replay quality depends not only on buffer size, but also on whether the buffered graph reproduces historical training signals. To identify the sources of $\epsilon_{\mathcal{B}}$, let $g_{\mathcal{B}}^{\text{orig}}$ be the hypothetical gradient obtained from the same selected target nodes while retaining their original historical graph contexts. This quantity is used only for analysis. By the triangle inequality,

$$\epsilon_{\mathcal{B}} \leq \underbrace{\|g_o - g_{\mathcal{B}}^{\text{orig}}\|_2}_{\epsilon_{\text{sel}}} + \underbrace{\|g_{\mathcal{B}}^{\text{orig}} - g_{\mathcal{B}}\|_2}_{\epsilon_{\text{ctx}}} .$$

Here, ϵ_{sel} measures how well the selected target nodes represent the historical data, whereas ϵ_{ctx} measures the distortion caused by reconstructing them with incomplete heterogeneous contexts.

Proposition 2 (Insufficiency of target-node-only replay) *For a relation-sensitive HGNN whose prediction is non-constant in at least one relation-specific message, there exist two heterogeneous graphs $\mathcal{G}$ and $\mathcal{G}'$ containing the same target-node feature and label (x_v, y_v) but inducing different representations and predictions:*

$$h_\theta(v, \mathcal{G}) \neq h_\theta(v, \mathcal{G}'), \qquad f_\theta(v, \mathcal{G}) \neq f_\theta(v, \mathcal{G}').$$

Thus, target-node features and labels alone are not sufficient, in general, to reconstruct the historical prediction function.

Proposition 2 establishes the need to retain heterogeneous neighborhoods. We further show in Appendix A.1 that, under standard Lipschitz conditions,

$$\epsilon_{\text{ctx}} \leq C \sum_{v\in\mathcal{B}} \sum_{r\in\mathcal{R}} \|W_r\|_2 \Delta_r(v; \mathcal{G}, \widetilde{\mathcal{G}}),$$

where $C > 0$ is a constant and $\Delta_r(v; \mathcal{G}, \widetilde{\mathcal{G}})$ measures omitted relation-specific messages and changes in aggregation weights in replay subgraph $\widetilde{\mathcal{G}}$. This analysis motivates the two stages of DiSCo. Coverage Maximization (CM) selects representative target nodes as a practical surrogate for reducing ϵ_{sel}, while multi-type neighbor expansion retains typed neighborhoods and relation-specific connections to reduce ϵ_{ctx}.

Step 1: Reducing target-selection error. The first term ϵ_{sel} arises when the retained target nodes fail to represent the historical target distribution. Prior research has shown that different data points contribute unequally to model training: some significantly enhance performance, while others offer limited benefit (Toneva et al., 2018). Motivated by this, we use Coverage Maximization as a practical surrogate for reducing this discrepancy by retaining class-separated target nodes in the feature (embedding) space. Formally, given the training node set V_t^{tr} of task $\mathcal{T}_t$, CM selects a subset by maximizing the coverage of the feature/embedding space:

$$\mathcal{N}(v_i) = \{v_j \mid \text{dist}(v_i, v_j) < d,\ \mathcal{Y}(v_i) \neq \mathcal{Y}(v_j)\}, \tag{1}$$

where $\mathcal{Y}(v_i)$ denotes the label of node v_i, and $\mathcal{N}(v_i)$ represents the set of nodes from different classes that are within a distance d of v_i. Given the memory buffer $\mathcal{B}_{\tau_t}$ for the target node type τ_t, we select e nodes per class with the smallest $|\mathcal{N}(v_i)|$ as representative experiences.

Step 2: Reducing heterogeneous-context error. Even for a representative target set, replaying isolated target nodes can produce a large ϵ_{ctx}, because an HGNN prediction depends on typed neighbors and relation-specific message passing. We therefore expand each selected target node toward multiple node types and retain relation-aware neighbors. The resulting induced subgraph provides a compact approximation to the original heterogeneous computation context. Under standard *Lipschitz* assumptions on message passing and the loss gradient, ϵ_{ctx} can be bounded by the relation-specific message discrepancy between the original graph and the replay subgraph. The detailed result is provided in Appendix A.1.

Let $\mathcal{S}_t \subseteq V_t^{\text{tr}}$ denote the representative target nodes selected in Step 1, and let $\tau_t \in \mathcal{A}$ denote their target node type. For each non-target node type $\tau \in \mathcal{A} \setminus \{\tau_t\}$, let $\mathcal{R}(\tau_t, \tau) \subseteq \mathcal{R}$ denote the set of relation types connecting τ_t and τ. We collect the candidate context nodes of type τ as

$$\mathcal{C}_{\tau,t} = \bigcup_{v \in \mathcal{S}_t} \bigcup_{r \in \mathcal{R}(\tau_t, \tau)} \{u \in V_{\tau,t} \mid (v,u) \in E_{r,t} \vee (u,v) \in E_{r,t}\}, \tag{2}$$

where $V_{\tau,t}$ denotes the nodes of type τ available at task t, and $E_{r,t}$ denotes the edges of relation type r in the current graph snapshot. We score each candidate node by its relation-specific connectivity:

$$\pi_t(v) = \sum_{r \in \mathcal{R}} \deg_{r,t}(v), \tag{3}$$

where $\deg_{r,t}(v)$ measures the connectivity strength of relation type r for node v at task t. For non-target node type τ with buffer budget b_τ, we select top-b_τ candidate nodes by $\pi_t(v)$:

$$\mathcal{S}_{\tau,t} = \text{TopK}_{v \in \mathcal{C}_{\tau,t}} (\pi_t(v), b_\tau). \tag{4}$$

Using the representative target set $\mathcal{S}_t$ and the selected context nodes $\{\mathcal{S}_{\tau,t}\}_{\tau \in \mathcal{A} \setminus \{\tau_t\}}$, we construct the induced heterogeneous replay subgraph $\mathcal{G}_{\text{sub}}^t = (V_t', E_t')$ as

$$V_t' = \mathcal{S}_t \cup \bigcup_{\tau \in \mathcal{A} \setminus \{\tau_t\}} \mathcal{S}_{\tau,t}, \qquad E_t' = \{(u,v) \in E_t \mid u, v \in V_t'\}. \tag{5}$$

For the selected target nodes in $\mathcal{S}_t$, we retain their features and labels, while the sampled non-target nodes and induced edges are retained as heterogeneous message-passing context. The constructed subgraphs are stored in the replay memory. When learning task $\mathcal{T}_t$, the model combines the current-task loss with a replay loss evaluated on the labeled historical target nodes using their reconstructed heterogeneous contexts:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\mathcal{T}_t}(\mathcal{G}_t, \theta) + \lambda_{er} \mathcal{L}_{er}(\mathcal{G}_{sub}, \theta) \\ &= -\sum_{v_i \in V_t^{tr}} \mathcal{Y}(v_i) \log \hat{\mathcal{Y}}(v_i) - \lambda_{er} \sum_{v_j \in \mathcal{B}} \mathcal{Y}(v_j) \log \hat{\mathcal{Y}}(v_j), \end{aligned} \tag{6}$$

where $\hat{\mathcal{Y}}(v_i)$ is the predicted label of node v_i and λ_{er} controls the contribution of buffered data in the overall loss. Equation 6 uses the DiSCo subgraph to instantiate $g_{\mathcal{B}}$. The two stages of DiSCo target the selection and context components of the replay approximation error, respectively.

3.2 Heterogeneity-aware Knowledge Distillation for Task Alignment

While DiSCo preserves representative heterogeneous computation contexts, replay alone cannot fully constrain the learner's semantic changes across tasks. Existing knowledge distillation methods are often developed for homogeneous scenarios and mainly focus on prediction-level alignment. For HGNNs, however, relation- and metapath-specific responses may drift even when output behavior is partially preserved. In contrast, our HKD (Heterogeneity-aware Knowledge Distillation) is explicitly tailored for HCGL. Instead of relying solely on logit-based distillation, we propose a two-level alignment strategy that leverages both prediction and semantic signals to capture graph heterogeneity. The logit-level distillation transfers soft label distributions to retain discriminative knowledge, while the semantic-level alignment matches metapath-aware attention scores, enabling the student model to preserve high-order structural patterns unique to heterogeneous graphs.

Logit-level. Inspired by previous distillation-based approaches (Li & Hoiem, 2017; Tian et al., 2023), we transfer the soft knowledge from the teacher model (previous task classifier) to the student model (current task classifier), enabling the student model to learn both new knowledge from the current task and retained knowledge from past tasks. Mimicking the teacher's prediction results enables the student model to learn secondary information from previous tasks that cannot be expressed by the experience replay data alone. Soft knowledge from teacher model is formulated as the predicted probability of the labels in the current task data:

$$p_{T,l}(z_i, t) = \text{Softmax}\big(f_{T,l}(z_i), t\big) = \frac{\exp\left[f_{T,l}(z_i)/t\right]}{\sum_j \exp\left[f_{T,l}(z_j)/t\right]}, \tag{7}$$

where z_i is the embedding of node v_i in $\mathcal{V}_i^{tr}$, $f_{T,i}(z_i)$ is the score logit of z_i obtained from l-th teacher model, and t is the temperature index to soften the peaky softmax distribution (Hinton, 2015). Thus, the knowledge distillation loss of the teacher model and the student model is defined as follows:

$$\mathcal{L}_{\text{logit}} = \frac{t^2}{N} \sum_{l}^{N} \sum_{v_i \in \mathcal{V}_i^{tr}} p_{T,l}(z_i, t) \log \frac{p_{T,l}(z_i, t)}{p_S(z_i, t)}, \tag{8}$$

where N is the number of used teacher models, $p_{T,l}$ and p_S are the predicted distributions of teacher model and student model respectively.

Semantic-level. To preserve metapath-induced semantic patterns, we propose an attention-based structural distillation method. Given a predefined metapath set $\mathcal{P} = \{P_1, \ldots, P_M\}$, let $\boldsymbol{\alpha}_{P_m}^T(v_i)$ and $\boldsymbol{\alpha}_{P_m}^S(v_i)$ denote the attention coefficients for metapath P_m computed by the teacher and student models, respectively. These attention coefficients reflect the importance of different semantic contexts encoded by heterogeneous structural patterns. To align the semantic-level knowledge between the teacher and the student, we define the semantic alignment loss as:

$$\mathcal{L}_{\text{sem}} = \frac{1}{N} \sum_{l=1}^{N} \sum_{m=1}^{M} \|\boldsymbol{\alpha}_{l,P_m}^T - \boldsymbol{\alpha}_{P_m}^S\|_2 \tag{9}$$

where $\boldsymbol{\alpha}_{P_m} \in \mathbb{R}^{|V^{tr}|}$ is the normalized attention vector over all nodes for metapath P_m. We provide a detailed discussion in Appendix A.3 on how this module can be extended to arbitrary HGNNs beyond metapath-based architectures.

This loss guides the student model to capture relational importance and structural dependencies aligned with the teacher model, thereby preserving semantic information that is critical in heterogeneous graphs. Afterwards, we combine the logit level loss and the semantic level loss:

$$\mathcal{L}_{kd} = \lambda_{\text{logit}} \mathcal{L}_{\text{logit}} + \lambda_{\text{sem}} \mathcal{L}_{\text{sem}}, \tag{10}$$

where λ_{logit} and λ_{sem} are both trade-off weights for balancing the losses. The final objective of node representation learning is to minimize the joint loss including current task loss $\mathcal{L}_{\mathcal{T}_i}$, experience replay loss $\mathcal{L}_{er}$, and the KD loss $\mathcal{L}_{kd}$:

$$\mathcal{L}_{\text{meta}} = \mathcal{L}_{\mathcal{T}_t} + \lambda_{er} \mathcal{L}_{er} + \lambda_{kd} \mathcal{L}_{kd}, \tag{11}$$

where λ_{kd} is a trade-off weight for balancing the losses.

3.3 Fast Adaptation with Gradient-based Meta-Learning

The previous two components primarily focus on retaining historical knowledge. We now consider the complementary problem of maintaining plasticity when a new task arrives. Inspired by the look-ahead optimization principle of La-MAML (Gupta et al., 2020), we adopt a simplified meta-adaptation strategy tailored to heterogeneous graphs. The main idea is to first perform a fast update on a small representative support graph and then use the joint continual objective to regulate the adaptation direction. Specifically, for the current task $\mathcal{T}_t$, we construct a support graph $\mathcal{G}_{\text{sup}}^t$ from e representative nodes selected by the sampling method introduced in Section 3.1. At optimization step j, we obtain the fast-adapted parameters through

$$\theta_{\text{fast}}^j = \theta^j - \alpha^j \odot \nabla_{\theta^j} \mathcal{L}_{\text{sup}}(\mathcal{G}_{\text{sup}}^t, \theta^j), \tag{12}$$

where α^j denotes the parameter-wise adaptation rate.

Rather than directly accepting this fast update, we evaluate it using the joint continual learning objective $\mathcal{L}_{\text{meta}}$ in Equation 11. Following the look-ahead principle, the adaptation rates are updated as

$$\alpha^{j+1} = \alpha^j - \eta \odot \nabla_{\alpha^j} \mathcal{L}_{\text{meta}}(\mathcal{G}_t, \theta_{\text{fast}}^j), \tag{13}$$

and then update the model parameters as

$$\theta^{j+1} = \theta_{\text{fast}}^j - \max(0, \alpha^{j+1}) \odot \nabla_{\theta_{\text{fast}}^j} \mathcal{L}_{\text{meta}}(\mathcal{G}_t, \theta_{\text{fast}}^j). \tag{14}$$

This look-ahead procedure complements the two retention mechanisms introduced above. The support graph provides an informative direction for adapting to the incoming task, while the joint

Table 1: Performance comparison on four benchmark datasets using HAN as the backbone. Results on Yelp are reported under the 5 tasks/3-way setting. The symbol ↑ (↓) indicates that higher (lower) is better. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	DBLP		IMDB		Freebase		Yelp	
	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓
Finetune	82.8 ± 4.6	26.6 ± 9.1	68.8 ± 2.2	26.0 ± 2.8	53.8 ± 4.7	25.7 ± 8.8	24.5 ± 0.5	44.8 ± 1.6
Joint Train	95.2 ± 0.6	1.7 ± 1.0	80.4 ± 0.7	3.9 ± 0.8	65.6 ± 0.9	2.8 ± 2.0	63.0 ± 2.3	0.0 ± 1.6
EWC	85.4 ± 5.5	21.5 ± 10.6	69.6 ± 1.2	25.8 ± 2.4	58.3 ± 2.4	18.4 ± 4.2	25.3 ± 2.1	44.8 ± 0.9
MAS	91.1 ± 0.8	8.3 ± 1.5	72.0 ± 2.3	20.5 ± 3.9	59.0 ± 1.0	14.8 ± 0.9	36.7 ± 10.9	27.6 ± 16.2
TWP	90.5 ± 1.3	10.1 ± 2.6	74.8 ± 2.7	14.5 ± 4.4	60.9 ± 4.3	**10.8 ± 6.9**	32.0 ± 12.1	36.4 ± 15.1
ER-GNN	89.7 ± 2.9	13.1 ± 6.1	74.9 ± 2.3	12.7 ± 4.2	56.5 ± 1.3	19.6 ± 0.9	40.2 ± 14.2	22.5 ± 18.2
MetaCLGraph	89.9 ± 0.9	11.4 ± 2.3	74.8 ± 3.2	14.8 ± 5.7	59.4 ± 2.5	13.0 ± 4.1	37.4 ± 12.3	30.7 ± 15.9
FTF-ER	90.8 ± 1.4	9.8 ± 2.4	72.0 ± 4.2	19.7 ± 6.4	58.1 ± 0.7	14.6 ± 0.6	32.8 ± 12.5	36.2 ± 16.0
DMSG	93.5 ± 0.6	**0.7 ± 0.6**	68.8 ± 3.9	25.1 ± 7.8	57.1 ± 2.4	17.5 ± 3.6	50.3 ± 0.6	**5.6 ± 0.5**
HERO	**93.6 ± 0.8**	4.3 ± 1.6	**78.4 ± 1.7**	**7.3 ± 3.0**	**60.9 ± 1.5**	11.1 ± 1.6	**55.9 ± 0.5**	7.5 ± 1.2

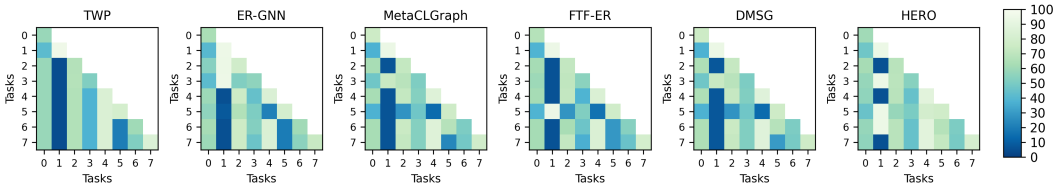


Figure 2: Accuracy matrix of selected continual graph learning methods on the Yelp dataset.

objective evaluates whether this fast update remains compatible with the historical constraints supplied by DiSCo and HKD. Compared with directly optimizing the joint loss, the parameter-wise adaptation rates allow the update magnitude to be adjusted according to this compatibility, providing a lightweight mechanism for maintaining new-task plasticity while respecting historical knowledge. The complete training procedure of HERO is summarized in Algorithm 1.

4 Experiments

Experiment Setup. In this section, we describe the experiments we perform to validate our proposed method. We evaluate HERO on four heterogeneous graph benchmarks: Yelp (Yang et al., 2020), IMDB, DBLP, and Freebase (Lv et al., 2021). Following the continual-learning protocol, we partition their target classes into sequential tasks and construct the corresponding growing graph snapshots. This results in 5, 2, 2, and 3 tasks for Yelp, IMDB, DBLP, and Freebase, respectively. For Yelp, which contains a larger number of target classes, we consider two settings to examine different sequence granularities: 5 tasks/3-way and 8 tasks/2-way. We adopt three widely used HGNN backbones in our experiments: HAN (Wang et al., 2019), MAGNN (Fu et al., 2020), and HGT (Hu et al., 2020). For baselines, we select multiple methods from the Continual Graph Learning Benchmark (CGLB) (Zhang et al., 2022c), including EWC (Kirkpatrick et al., 2017), MAS (Aljundi et al., 2018), ER-GNN (Zhou & Cao, 2021), and TWP (Liu et al., 2021). Additionally, we include several memory-replay methods: MetaCLGraph (Unal et al., 2023), FTF-ER (Pang et al., 2024) and DMSG (Qiao et al., 2025). We use Finetune (without any continual learning techniques) as a lower bound, and Joint Train (trained jointly on data from all tasks) as an approximate upper bound (Caruana, 1997). For evaluation, we first construct an accuracy matrix $W^{Acc} \in \mathcal{R}^{T\times T}$, where $W^{Acc}_{i,j}(i \geq j)$ denotes the accuracy on the j-th task after learning task i. To summarize the final performance after learning all tasks, we report the Average Performance (AP): $\mathrm{AP} = \frac{1}{T}\sum_{i=1}^{T} W^{\mathrm{Acc}}_{T,i}$ and the Average Forgetting (AF): $\mathrm{AF} = \frac{1}{T-1}\sum_{i=1}^{T-1}(W^{\mathrm{Acc}}_{i,i} - W^{\mathrm{Acc}}_{T,i})$. For a more detailed description of the experimental setup, please refer to Appendix B.4.

Performance Evaluation. The overall performance on four heterogeneous graph datasets using HAN as the backbone is reported in Table 1. Results with additional HGNN backbones are provided in Appendix D.1. Task-wise accuracy retention results on Yelp are further illustrated in Figure 3 and Figure 2. In addition, we conduct extended experiments to assess the robustness and scalability of HERO, including GPU memory usage and the teacher-storage cost over longer task sequences, with

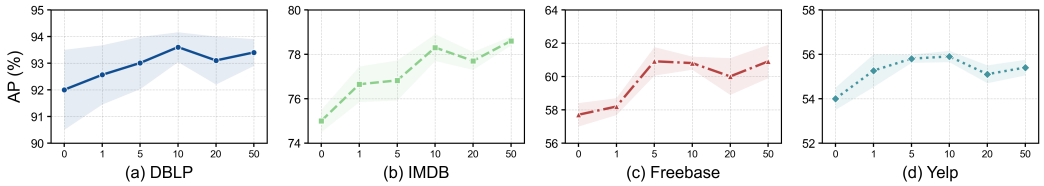


Figure 4: Effect of the shot number used by G-MM across four datasets.

Table 2: Ablation results of HERO. We remove Experience Replay (ER), Heterogeneity-aware Knowledge Distillation (HKD), and the Gradient-based Meta-learning Module (G-MM) individually. And "w/o DiSCo" denotes replacing the proposed *Diversity Sampling with semantic Consistency* with Coverage Maximization applied to all node types.

Methods	DBLP		IMDB		Freebase		Yelp	
	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓
HERO	**93.6 ± 0.8**	**4.3 ± 1.6**	**78.4 ± 1.7**	**7.3 ± 3.0**	**60.9 ± 1.5**	**11.1 ± 1.6**	**55.9 ± 0.6**	**7.5 ± 1.2**
w/o ER	89.7 ± 0.6	12.0 ± 0.7	75.1 ± 0.9	12.8 ± 1.7	58.6 ± 1.2	16.3 ± 2.2	53.9 ± 2.4	6.8 ± 2.4
w/o HKD	89.6 ± 1.4	12.1 ± 2.6	72.2 ± 1.5	19.3 ± 2.3	59.4 ± 1.1	14.7 ± 1.8	44.9 ± 14.3	19.3 ± 17.0
w/o G-MM	92.6 ± 1.7	5.2 ± 0.9	75.0 ± 2.7	13.3 ± 4.5	59.7 ± 0.9	12.0 ± 2.2	55.6 ± 1.9	8.0 ± 2.0
w/o DiSCo	92.4 ± 1.4	6.0 ± 3.1	75.4 ± 2.2	12.1 ± 5.0	56.2 ± 0.4	18.0 ± 0.5	45.7 ± 14.0	23.7 ± 15.6

detailed results provided in Appendix C and D.2. We highlight the following key observations: **(1)** Sequentially fine-tuning HGNN backbones without CL mechanisms leads to substantial forgetting. For example, the Finetune baseline yields an AF of 26.6% on DBLP with HAN, while similar forgetting is observed with the other backbones (Appendix D.1). This confirms that heterogeneous graph models are not immune to catastrophic forgetting under sequential task updates. **(2)** Across the reported backbones, HERO achieves the highest or tied-highest AP among the continual learning baselines. The gains are particularly clear on IMDB and Yelp, while the results on DBLP and Freebase are also competitive. With HAN, HERO achieves the lowest or second-lowest AF among continual-learning baselines on all four datasets. Additional experiments with randomized task orders on HAN and analyses using the Forgetting-aware Gap metric indicate that HERO generally maintains its advantages under task-sequence variations and often attains a favorable stability–plasticity trade-off.

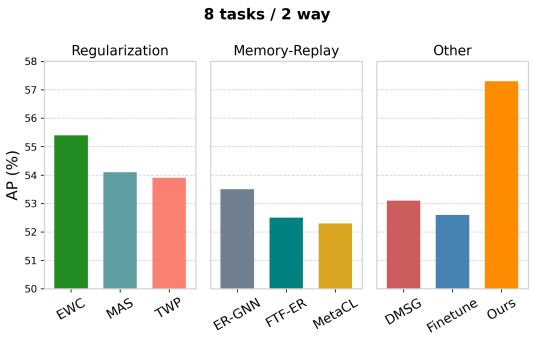


Figure 3: Performance comparison on Yelp under the 8-task/2-way setting.

(3) Some replay-based methods obtain lower AF than HERO in individual settings despite having lower AP. For example, DMSG achieves lower AF on DBLP and Yelp, whereas HERO obtains higher AP. This indicates that different methods operate at different points of the stability–plasticity trade-off: stronger retention does not necessarily lead to higher overall performance. HERO aims for a balanced operating point, achieving consistently strong AP while keeping forgetting competitive.

Ablation Study. To evaluate the effectiveness of individual modules in HERO, we conducted comprehensive ablation experiments using HAN as the backbone network across four datasets. The results are presented in Table 2. We first examined the impact of each component by sequentially removing Experience Replay (ER), Heterogeneity-aware Knowledge Distillation (HKD), and the Gradient-based Meta-learning Module (G-MM) while keeping other components intact. Removing every component decreases AP on all four datasets, while most ablations also increase AF. An exception occurs on Yelp, where removing ER slightly decreases AF from 7.5% to 6.8% but also reduces AP from 55.9% to 53.9%. This illustrates that AP and AF capture different aspects of the stability–plasticity trade-off. The ablations reveal different roles of the three components. Removing G-MM causes a relatively smaller AP reduction (0.3 to 3.4 percentage points), consistent with its role as a lightweight adaptation component. In contrast, removing either HKD or DiSCo leads to larger degradation on several datasets. HKD has a particularly strong effect on DBLP and IMDB, whereas DiSCo causes larger forgetting increases on Freebase and Yelp. These results support the complementary roles of semantic-response preservation

and heterogeneous-context replay in historical knowledge retention. Replacing DiSCo with Coverage Maximization consistently lowers AP and generally increases AF, with particularly large degradation on Freebase and Yelp. This result is consistent with our analysis that retaining typed neighborhood context improves replay quality beyond target-node diversity alone. Detailed analyses of HKD submodules are provided in Appendix D.3.

Hyperparameters Analysis. We analyze the replay buffer size, shot number (i.e., the number of support samples used by G-MM), and the distillation loss weight λ_{kd}. Due to space limitations, we visualize only the effect of the shot number in Figure 4 and provide the remaining results in Appendix D.5. The experimental findings are as follows: **Distillation Loss Weight (λ_{kd}).** The preferred distillation strength is dataset dependent. IMDB benefits from relatively larger λ_{kd}, whereas Freebase performs better with smaller values around 0.1 to 0.3. Yelp also shows stronger performance under relatively larger distillation weights. Overall, these results indicate that the balance between current-task learning and historical constraints varies with the dataset. **Buffer Size.**: AP varies non-monotonically with the replay budget. Moderate buffers perform best on DBLP, IMDB, and Freebase, whereas Yelp remains relatively stable for buffer sizes of 10, 20, and 100. Thus, increasing the buffer does not guarantee higher AP, and the preferred budget depends on the dataset. **Shot Number.**: The number of sampled nodes impacts the stability-plasticity trade-off. As shown in Figure 4, model performance steadily improves up to 10-shot, but declines at 20-shot, likely due to overfitting to the current task's data patterns. The optimal shot number is 10, balancing historical information retention and computational efficiency.

5 Related Work

Heterogeneous Graph Neural Networks. HGNNs model type-dependent structures and semantics through either metapath-based aggregation (Wang et al., 2019; Fu et al., 2020; Zhang et al., 2019) or metapath-free relation modeling (Hu et al., 2020; Lv et al., 2021; Zhu et al., 2019). These models are commonly studied in offline settings where the training graph is jointly accessible. In contrast, our work studies how existing HGNN backbones can be sequentially updated under limited access to historical data. **Continual Graph Learning.** Existing CGL methods mainly rely on parameter regularization (Kirkpatrick et al., 2017; Aljundi et al., 2018; Liu et al., 2021; Qiu et al., 2025), experience replay (Zhou & Cao, 2021; Choi et al., 2024; Zhang et al., 2024a; Qiao et al., 2025; Song et al., 2025), or knowledge distillation (Xu et al., 2020; Dong et al., 2021; Tian et al., 2026). Many of these methods are designed for homogeneous graphs or do not explicitly preserve heterogeneous computation contexts (Zhang et al., 2023b; Guo et al., 2026). In HGNNs, a target-node prediction depends on typed neighborhoods and relation- or metapath-specific semantic channels. Consequently, replaying isolated nodes may not reconstruct historical training signals, while output-level alignment alone may not preserve internal heterogeneous semantics. HERO addresses these limitations through heterogeneous subgraph replay and multi-level knowledge distillation. **Meta-Learning.** Optimization-based meta-learning supports rapid task adaptation by learning transferable initializations or update rules (Finn et al., 2017; Nichol & Schulman, 2018; Gupta et al., 2020). It has also been applied to continual learning and graph learning (Javed & White, 2019; Unal et al., 2023; Zhang et al., 2022a). HERO adopts a lightweight look-ahead adaptation mechanism to improve plasticity on incoming tasks, while replay and distillation provide historical constraints.

6 Conclusion

In this work, we studied heterogeneous continual graph learning, where HGNNs learn sequential tasks without full access to historical data. We showed that replaying target nodes alone can be insufficient because historical predictions depend on typed neighborhood contexts and relation-dependent semantics. Based on this observation, we proposed HERO to preserve historical knowledge at both structural and semantic levels. DiSCo selects representative target nodes and reconstructs compact heterogeneous contexts through relation-aware multi-type expansion, while HKD aligns task-conditioned predictions and semantic responses across learning stages. A lightweight look-ahead adaptation step further supports plasticity on incoming tasks. Experiments on four datasets with three representative HGNN backbones show that HERO achieves the best or tied-best average performance among strong baselines while maintaining competitive forgetting, supporting the importance of jointly preserving structure and semantics in HCGL.

7 AI USAGE DISCLOSURE

The generative AI tools employed in this work are solely used for language polishing and refinement. These tools do not contribute to the development of the core ideas, methodologies, or results presented in the work.

REFERENCES

Kian Ahrabian, Yishi Xu, Yingxue Zhang, Jiapeng Wu, Yuening Wang, and Mark Coates. Structure aware experience replay for incremental learning in graph-based recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2832–2836, 2021.

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)*, September 2018.

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

Seungyoon Choi, Wonjoong Kim, Sungwon Kim, Yeonjun In, Sein Kim, and Chanyoung Park. DSLR: Diversity enhancement and structure learning for rehearsal-based graph continual learning. In *The Web Conference 2024*, 2024. URL https://openreview.net/forum?id=D8Mb4c7V0t.

Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:1255–1263, 05 2021.

Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Chaofan Fu, Guanjie Zheng, Chao Huang, Yanwei Yu, and Junyu Dong. Multiplex heterogeneous graph neural network with behavior pattern modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 482–494, 2023.

Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, pp. 2331–2341, 2020.

Zihao Guo, Qingyun Sun, Ziwei Zhang, Haonan Yuan, Huiping Zhuang, Xingcheng Fu, and Jianxin Li. Graphkeeper: Graph domain-incremental learning via knowledge disentanglement and preservation. *Advances in Neural Information Processing Systems*, 38:1145–1171, 2026.

Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. *Advances in Neural Information Processing Systems*, 33:11588–11598, 2020.

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.

Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

Huiting Hong, Hantao Guo, Yucheng Lin, Xiaoqing Yang, Zang Li, and Jieping Ye. An attention-based graph neural network for heterogeneous structural learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4132–4139, 2020.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pp. 2704–2710, 2020.

Khurram Javed and Martha White. Meta-learning representations for continual learning. *Advances in neural information processing systems*, 32, 2019.

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.

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

Ziming Li, Xiaoming Wu, Zehong Wang, Jiazheng Li, Yijun Tian, Jinhe Bi, Yunpu Ma, Yanfang Ye, and Chuxu Zhang. Graph is a substrate across data modalities. *arXiv preprint arXiv:2601.22384*, 2026.

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.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 1150–1160, 2021.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Jinhui Pang, Changqing Lin, Xiaoshuai Hao, Rong Yin, Zixuan Wang, Zhihui Zhang, Jinglin He, and Huang Tai Sheng. Ftf-er: Feature-topology fusion-based experience replay method for continual graph learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 8336–8344, 2024.

Ziyue Qiao, Junren Xiao, Qingqiang Sun, Meng Xiao, Xiao Luo, and Hui Xiong. Towards continuous reuse of graph models via holistic memory diversification. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pbz4i7B0B4.

Rihong Qiu, Xinke Jiang, Yuchen Fang, Hongbin Lai, Hao Miao, Xu Chu, Junfeng Zhao, and Yasha Wang. Efficient graph continual learning via lightweight graph neural tangent kernels-based dataset distillation. In *Forty-second International Conference on Machine Learning*, 2025.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pp. 593–607. Springer, 2018.

Chuan Shi, Binbin Hu, Wayne Xin Zhao, Philip S Yu, and Yanchi Liu. Heterogeneous information network embedding for recommendation. 2018. URL https://arxiv.org/pdf/1711.10730.pdf.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Lei Song, Jiaxing Li, Qinghua Si, Shihan Guan, and Youyong Kong. Exploring rationale learning for continual graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20540–20548, 2025.

Yizhou Sun and Jiawei Han. *Mining heterogeneous information networks: principles and methodologies*. Morgan & Claypool Publishers, 2012.

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4 (11):992–1003, 2011.

Qin Tian, Chen Zhao, Xintao Wu, Dong Li, Minglai Shao, Xujiang Zhao, and Wenjun Wang. Class-domain incremental learning on graphs via disentangled knowledge distillation. In *Proceedings of the ACM Web Conference 2026*, number 11, pp. 452–462, 2026.

Yijun Tian, Shichao Pei, Xiangliang Zhang, Chuxu Zhang, and Nitesh V Chawla. Knowledge distillation on graphs: A survey. *arXiv preprint arXiv:2302.00219*, 2023.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Altay Unal, Abdullah Akgül, Melih Kandemir, and Gozde Unal. Meta continual learning on graphs with experience replay. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=8tnrh56P5W.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pp. 2022–2032, 2019.

Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 2861–2868. ACM, October 2020.

Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4854–4873, 2020.

Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10816–10824, 2023.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 793–803, 2019.

Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. Continual learning on dynamic graphs via parameter isolation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 601–611, 2023a.

Qiannan Zhang, Xiaodong Wu, Qiang Yang, Chuxu Zhang, and Xiangliang Zhang. Hg-meta: Graph meta-learning over heterogeneous graphs. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 397–405. SIAM, 2022a.

Rui Zhang, Arthur Zimek, and Peter Schneider-Kamp. A simple meta-path-free framework for heterogeneous network embedding. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2600–2609, 2022b.

Xikun Zhang, Dongjin Song, and Dacheng Tao. CGLB: Benchmark tasks for continual graph learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022c. URL https://openreview.net/forum?id=5wNiiIDynDF.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Hierarchical prototype networks for continual graph representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4622–4636, 2022d.

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, 2022e.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Ricci curvature-based graph sparsification for continual graph representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023b.

Xikun Zhang, Dongjin Song, Yixin Chen, and Dacheng Tao. Topology-aware embedding memory for continual learning on expanding networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4326–4337, 2024a.

Xikun Zhang, Dongjin Song, and Dacheng Tao. Continual learning on graphs: Challenges, solutions, and opportunities. *arXiv preprint arXiv:2402.11565*, 2024b.

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.

Guanghui Zhu, Zhennan Zhu, Hongyang Chen, Chunfeng Yuan, and Yihua Huang. Hagnn: Hybrid aggregation for heterogeneous graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Shichao Zhu, Chuan Zhou, Shirui Pan, Xingquan Zhu, and Bin Wang. Relation structure-aware heterogeneous graph neural network. In *2019 IEEE international conference on data mining (ICDM)*, pp. 1534–1539. IEEE, 2019.

A METHODOLOGY

A.1 THEORETICAL ANALYSIS OF DISCO

Proof of Proposition 1 Since $F_{<t}$ is β-smooth, for any parameter displacement d,

$$F_{<t}(\theta + d) \leq F_{<t}(\theta) + \langle \nabla_\theta F_{<t}(\theta), d \rangle + \frac{\beta}{2}\|d\|_2^2.$$

For the replay-based update,

$$d = -\eta \left(g_t + \lambda_{\text{er}} g_{\mathcal{B}} \right).$$

Substituting d and using $g_o = \nabla_\theta F_{<t}(\theta)$ gives

$$\begin{aligned} F_{<t}(\theta^+) - F_{<t}(\theta) \leq & -\eta \langle g_o, g_t + \lambda_{\text{er}} g_{\mathcal{B}} \rangle \\ & + \frac{\beta\eta^2}{2} \|g_t + \lambda_{\text{er}} g_{\mathcal{B}}\|_2^2 \\ = & -\eta \langle g_o, g_t \rangle - \eta \lambda_{\text{er}} \langle g_o, g_{\mathcal{B}} \rangle \\ & + \frac{\beta\eta^2}{2} \|g_t + \lambda_{\text{er}} g_{\mathcal{B}}\|_2^2 . \end{aligned}$$

By the definition

$$\epsilon_{\mathcal{B}} = \|g_{\mathcal{B}} - g_o\|_2,$$

we have

$$\begin{aligned} \langle g_o, g_{\mathcal{B}} \rangle &= \langle g_o, g_o \rangle + \langle g_o, g_{\mathcal{B}} - g_o \rangle \\ &\geq \|g_o\|_2^2 - \|g_o\|_2 \|g_{\mathcal{B}} - g_o\|_2 \\ &= \|g_o\|_2^2 - \|g_o\|_2 \epsilon_{\mathcal{B}}, \end{aligned}$$

where the inequality follows from Cauchy–Schwarz. Substituting this lower bound into the previous expression proves the result.

Proof of Proposition 2: Target-Node Insufficiency Consider a target node v with feature x_v and label y_v. Let

$$b = W_0 x_v$$

denote its self-feature contribution.

Because the HGNN is relation-sensitive, there exist a relation r, a node feature x_u, and a nonzero aggregation coefficient a_{vu}^r such that the relation-specific message

$$m = a_{vu}^r W_r x_u$$

changes the target-node prediction:

$$f_\theta \left(\sigma(b + m) \right) \neq f_\theta \left(\sigma(b) \right).$$

Construct graph $\mathcal{G}$ by connecting v to node u through relation r. Construct graph $\mathcal{G}'$ using the same target node v, with the same feature x_v and label y_v, but without this relation-specific neighbor. Then

$$h_v(\mathcal{G}) = \sigma(b + m), \qquad h_v(\mathcal{G}') = \sigma(b),$$

and consequently

$$f_\theta(v, \mathcal{G}) \neq f_\theta(v, \mathcal{G}').$$

A memory that stores only (x_v, y_v) is identical for $\mathcal{G}$ and $\mathcal{G}'$, and therefore cannot determine which historical prediction function should be reconstructed.

Relation-Context Distortion under Subgraph Replay Let $\widetilde{\mathcal{G}} \subseteq \mathcal{G}$ be a replay subgraph. For relation r, define

$$\mathcal{C}_r(v) = \mathcal{N}_r^{\mathcal{G}}(v) \cap \mathcal{N}_r^{\widetilde{\mathcal{G}}}(v)$$

as the retained neighbors, and

$$\mathcal{O}_r(v) = \mathcal{N}_r^{\mathcal{G}}(v) \setminus \mathcal{N}_r^{\widetilde{\mathcal{G}}}(v)$$

as the omitted neighbors. We define the relation-context discrepancy as

$$\Delta_r(v; \mathcal{G}, \widetilde{\mathcal{G}}) = \sum_{u \in \mathcal{C}_r(v)} \left| a_{vu}^{r,\mathcal{G}} - a_{vu}^{r,\widetilde{\mathcal{G}}} \right| + \sum_{u \in \mathcal{O}_r(v)} \left| a_{vu}^{r,\mathcal{G}} \right|.$$

The first term accounts for changes in normalized aggregation weights on retained neighbors, while the second term accounts for omitted relation-specific messages.

Let the pre-activation representation be

$$s_v(\mathcal{G}) = W_0 x_v + \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{N}_r^{\mathcal{G}} \mathcal{G}(v)} a_{vu}^{r,\mathcal{G}} W_r x_u.$$

Because $\widetilde{\mathcal{G}}$ preserves the same target node, the self-feature terms cancel. Hence,

$$s_v(\mathcal{G}) - s_v(\widetilde{\mathcal{G}}) = \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{C}_r(v)} \left(a_{vu}^{r,\mathcal{G}} - a_{vu}^{r,\widetilde{\mathcal{G}}} \right) W_r x_u + \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{O}_r(v)} a_{vu}^{r,\mathcal{G}} W_r x_u.$$

Using the triangle inequality, the operator norm inequality, and $\|x_u\|_2 \le H$, we obtain

$$\|s_v(\mathcal{G}) - s_v(\widetilde{\mathcal{G}})\|_2 \le H \sum_{r \in \mathcal{R}} \|W_r\|_2 \Delta_r(v; \mathcal{G}, \widetilde{\mathcal{G}}).$$

Since σ is L_σ-Lipschitz,

$$\begin{aligned} \|h_v(\mathcal{G}) - h_v(\widetilde{\mathcal{G}})\|_2 &= \|\sigma(s_v(\mathcal{G})) - \sigma(s_v(\widetilde{\mathcal{G}}))\|_2 \\ &\le L_\sigma \|s_v(\mathcal{G}) - s_v(\widetilde{\mathcal{G}})\|_2 \\ &\le L_\sigma H \sum_{r \in \mathcal{R}} \|W_r\|_2 \Delta_r(v; \mathcal{G}, \widetilde{\mathcal{G}}). \end{aligned}$$

Corollary 1 (Context-induced replay-gradient error) *Suppose the per-example replay-gradient map $\psi(h, y)$ is L_g-Lipschitz in h. For a fixed set of replay targets $\mathcal{S}$,*

$$\left\| \frac{1}{|\mathcal{S}|} \sum_{v \in \mathcal{S}} \psi(h_v(\mathcal{G}), y_v) - \frac{1}{|\mathcal{S}|} \sum_{v \in \mathcal{S}} \psi(h_v(\widetilde{\mathcal{G}}), y_v) \right\|_2 \le \frac{L_g L_\sigma H}{|\mathcal{S}|} \sum_{v \in \mathcal{S}} \sum_{r \in \mathcal{R}} \|W_r\|_2 \Delta_r(v; \mathcal{G}, \widetilde{\mathcal{G}}).$$

By the triangle inequality and the Lipschitz continuity of ψ,

$$\left\| \frac{1}{|\mathcal{S}|} \sum_{v \in \mathcal{S}} \left[\psi(h_v(\mathcal{G}), y_v) - \psi(h_v(\widetilde{\mathcal{G}}), y_v) \right] \right\|_2 \le \frac{L_g}{|\mathcal{S}|} \sum_{v \in \mathcal{S}} \|h_v(\mathcal{G}) - h_v(\widetilde{\mathcal{G}})\|_2.$$

Substituting the relation-context distortion bound proves the result.

Algorithm 1 The procedure of HERO

Input: Continual tasks $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$; buffer $\mathcal{B}$; number of representative nodes e; adaptation rates α; meta learning rate η; sampling algorithm $\mathrm{DiSCo}(\cdot, \cdot)$.
Output: Continually learned model f_θ.
Initialize model parameters θ, adaptation rates α, buffer $\mathcal{B} \leftarrow \emptyset$, and teacher set $\mathcal{F} \leftarrow \emptyset$
for $t = 1$ **to** T **do**
  Obtain the current training graph $\mathcal{G}_t$ and training nodes V_t^{tr}
  Select representative nodes $\mathcal{E}_t = \mathrm{DiSCo}(V_t^{tr}, e)$ and construct support graph $\mathcal{G}_{sup}^t$
  for $j = 1$ **to** E **do**
    Fast adaptation:
    $\theta_{\text{fast}}^j \leftarrow \theta^j - \alpha^j \odot \nabla_{\theta^j} \mathcal{L}_{sup}(\mathcal{G}_{sup}^t, \theta^j)$
    Compute replay loss $\mathcal{L}_{er}$ over historical subgraphs in $\mathcal{B}$
    Compute logit- and semantic-level distillation losses $\mathcal{L}_{logit}$ and $\mathcal{L}_{sem}$ using teacher models in $\mathcal{F}$
    Construct the joint meta-objective:
    $\mathcal{L}_{meta} \leftarrow \mathcal{L}_{\mathcal{T}_t} + \lambda_{er}\mathcal{L}_{er} + \lambda_{kd}\mathcal{L}_{kd}$
    Meta update:
    $\alpha^{j+1} \leftarrow \alpha^j - \eta \nabla_{\alpha^j} \mathcal{L}_{meta}(\mathcal{G}_t, \theta_{\text{fast}}^j)$
    $\theta^{j+1} \leftarrow \theta_{\text{fast}}^j - \max(0, \alpha^{j+1}) \odot \nabla_{\theta_{\text{fast}}^j} \mathcal{L}_{meta}(\mathcal{G}_t, \theta_{\text{fast}}^j)$
  end for
  Sampling target-type nodes and heterogeneous neighbors via $\mathrm{DiSCo}(\cdot, \cdot)$
  Construct the replay subgraph and update buffer $\mathcal{B}$
  Add the current model f_θ to teacher set $\mathcal{F}$ and retain the latest K teachers
end for
Return f_θ

A.2 Detailed Algorithm

To better understand the proposed method, we provide the detailed algorithm 1 to illustrate the learning process.

A.3 Extension to General HGNNs

Previously, we have discussed how to align semantic-level information via knowledge distillation in meta-path-guided attention networks. In this section, we further demonstrate that the proposed method can be easily extended to HGNNs without attention mechanisms (such as RGCN), enabling the preservation of semantic information from previous tasks.

For each pair of target-type nodes $v_i, v_j \in V_{\tau_t}$, we define the attention score e_{ij} based on their hidden representations from the penultimate layer of the network $h_i^{(L-1)}$ and $h_j^{(L-1)}$, using the weight matrix $W^{(L)}$ from the last layer:

$$e_{ij} = \left(h_i^{(L-1)} W^{(L)}\right)^\top \tanh\left(h_j^{(L-1)} W^{(L)}\right), \tag{15}$$

We then apply softmax normalization to the attention scores for each node v_i to obtain an attention vector:

$$\boldsymbol{\alpha}_i = \mathrm{softmax}(e_{i1}, e_{i2}, \ldots, e_{i|V_{\tau_t}|}) \tag{16}$$

We compute attention vectors for both the teacher model $\boldsymbol{\alpha}_i^T$ and the student model $\boldsymbol{\alpha}_i^S$, and define the semantic alignment loss as the L_2 distance between the two:

$$\mathcal{L}_{\text{sem}} = \sum_{i \in V_{\tau_t}} \left\| \boldsymbol{\alpha}_i^T - \boldsymbol{\alpha}_i^S \right\|_2 \tag{17}$$

In addition to the node similarity-based attention defined above, for edge-type-aware attention models such as HGT, attention is computed for each relation type $r \in \mathcal{R}$. Let $\boldsymbol{\beta}_r^{(T)}$ and $\boldsymbol{\beta}_r^{(S)}$ denote the

attention vectors for relation r from the teacher and student models, respectively. The semantic alignment loss for such edge-type-based attention mechanisms is formulated as:

$$\mathcal{L}_{\text{sem}} = \sum_{r \in \mathcal{R}} \left\| \boldsymbol{\beta}_r^T - \boldsymbol{\beta}_r^S \right\|_2 \tag{18}$$

This formulation ensures that the student model preserves relation-aware structural semantics by aligning its attention distribution with that of the teacher model on each edge type.

B Experiments Setup

B.1 Datasets

Table 3: Key characteristics of datasets

Node Classification	#Nodes	#Node Types	#Edges	#Edge Types	#Classes
DBLP (Lv et al., 2021)	26,128	4	239,566	6	4
IMDB (Lv et al., 2021)	21,420	4	86,642	6	5
Freebase (Lv et al., 2021)	180,098	8	1,057,688	36	7
Yelp (Yang et al., 2020)	82,465	4	32,548,358	7	16

DBLP is a bibliographic dataset in the field of computer science. A widely adopted subset is used, containing four research areas, where nodes represent authors, papers, terms, and venues.

IMDB is a dataset derived from a online movie information platform. A subset including classes such as Action, Comedy, Drama, Romance, and Thriller is selected for use.

Freebase is a large-scale knowledge graph originally constructed for web search and semantic applications. A subgraph comprising eight genres of entities and approximately one million edges is sampled following procedures outlined in prior studies.

Yelp is a review network constructed from businesses, users, locations, and reviews. Although node features are not available, many business nodes are annotated with one or more labels from sixteen predefined categories. As a prototypical web application graph, Yelp embodies real-world challenges in recommendation and user–business interaction modeling.

B.2 Baselines

We introduce the baseline models that we choose to compare in the following section:

Elastic weight consolidation (EWC) Kirkpatrick et al. (2017) is a technique that regularizes the loss function, such that the model is encouraged to only modify the weights that are less important for the previous tasks. This is achieved by penalizing changes to weights that have large importance for the previously learned tasks, thereby mitigating catastrophic forgetting when learning new tasks.

Memory aware synapses (MAS) Aljundi et al. (2018) is a regularization-based method that measures the importance of parameters based on the sensitivity of predictions to these parameters. When learning a new task, the model adjusts the weights according to the network's activations to update the important parameters relevant to the new task data.

Topology-aware weight preserving(TWP) Liu et al. (2021) is a method introduced for graph continual learning that incorporates weight preservation mechanisms based on graph topology. By considering the local structure of the graph, TWP aims to overcome catastrophic forgetting. After computing the loss, the importance scores of the model weights are calculated according to the topology of the given graph, and the loss is regularized accordingly. This method leverages the properties of the graph.

ER-GNN Zhou & Cao (2021) stores samples from previous tasks as experiences and replays them when learning new tasks. After learning task T_i, sample nodes are saved to the buffer using a selection function. When learning the next task T_{i+1}, separate graphs are constructed for each learned

task $\{T_t\}_{t=1,\ldots,i}$. Then, the GNN is trained using these graphs. The overall loss is calculated by regularizing the current task's loss with the loss from the separately constructed graphs through experience replay.

MetaCLGraph Unal et al. (2023) combines experience replay and meta-learning. In this method, the initial parameters of the model are calculated using the current task data. When learning a new task, stored samples from previous tasks are merged with the current task data to form a new graph, which is then used to update the model parameters.

FTF-ER Pang et al. (2024) combines feature and global topological information by normalizing and weightedly integrating two types of node importance scores to evaluate node importance. Specifically, it introduces the Hodge Potential Score (HPS) module to capture global topological information. When learning a new task, a subgraph is induced using all the experience nodes stored in the buffer, the overall loss is calculated as the sum of the loss for the current task and the loss of the subgraph.

DMSG Qiao et al. (2025) selects representative nodes using a diversity-aware buffer strategy that accounts for both intra-class and inter-class variations, and further augments replay through variationally generated synthetic embeddings. Adversarial and reconstruction-based objectives are used to preserve embedding quality and improve generalization during replay.

B.3 Task Incremental HCGL

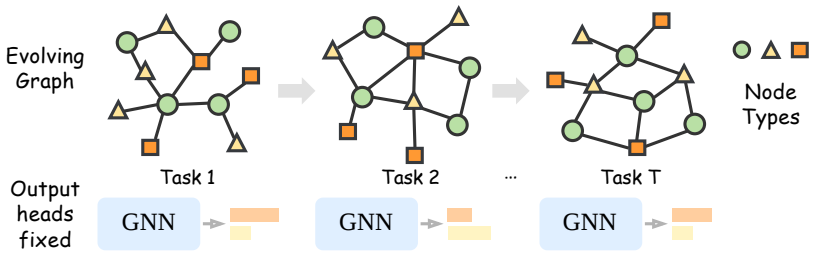


Figure 5: Illustration of the setting used in this paper.

In task-IL setting, a model requires the task indicator to perform inference on a given task, and the model only needs to distinguish different classes within each task without considering the classes from existing tasks. When the total number of classes is not divisible by K, the remaining class(es) are excluded from the continual task label space and are not used as target classes. Nodes belonging to the excluded target classes are removed from the constructed task sequence. Accordingly, DBLP uses all 4 classes to form two 2-way tasks; IMDB uses 4 of its 5 classes to form two 2-way tasks; Freebase uses 6 of its 7 classes to form three 2-way tasks; and Yelp uses 15 of its 16 classes to form five 3-way tasks. For the 8-task/2-way Yelp setting, all 16 classes are used.

B.4 Experimental Setting

Adam optimizer is used and the initial learning rate is set to 0.005 for all datasets. For HAN, each task is trained for 200 epochs with an early stopping patience of 100. For MAGNN, each task is trained for 100 epochs, with the early stopping patience set to 5 on DBLP and 10 on IMDB. For HGT, all tasks are trained for 300 epochs with a patience of 30. For datasets trained with mini-batches, the batch size is set to 8. The buffer size is uniformly set to 50 for the target node type in memory replay baselines, and 20 under mini-batch settings. For non-target node types, the buffer size is set to 200 for all memory replay methods. The hyperparameters for regularization-based baselines (e.g., EWC, MAS, and TWP) are set following the TWP official repository and the benchmark study (Zhang et al., 2022c). The hyper-parameters of HERO include some weight of loss $\lambda_{er}, \lambda_{kd}$. We chose them from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Different combinations are adjusted for different datasets to achieve optimal performance. The hyper-parameters of the Heterogeneity-aware knowledge distillation module (HKD) include λ_{logit} and λ_{sem}. We chose λ_{logit} from $\{0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$ and set λ_{sem} to 10.0 globally.

Table 4: GPU memory usage on Freebase. The first group compares HERO with replay-based methods, while the second group provides diagnostic controls for the additional overhead of HKD. Lower is better.

Method	Avg. GPU/task (MB) ↓	Peak GPU (MB) ↓
Replay-based methods		
HERO (Ours)	**7,046**	**8,107**
DMSG	7,126	8,131
ER-GNN	7,065	8,138
FTF-ER	7,097	8,122
MetaCLGraph	7,102	8,125
Diagnostic controls		
HERO w/o HKD	7,013	8,124
Finetune	6,973	8,121

Table 5: Effect of the teacher-window size on Yelp. $k = T - 1$ retains all historical teachers, while fixed k bounds teacher storage to $O(k)$.

	Yelp 5-task		Yelp 8-task	
Teachers	AP ↑	AF ↓	AP ↑	AF ↓
$k = 1$	53.51	13.41	55.55	11.31
$\mathbf{k = 3}$	$\mathbf{56.83}$	$\mathbf{9.29}$	$\mathbf{57.43}$	$\mathbf{7.98}$
$k = T - 1$	57.44	8.15	57.81	7.43

C Scalability Analysis

Memory and Long-Sequence Scalability. We further examine the scalability of HERO from both empirical memory usage and long-sequence model storage. We first measure GPU memory consumption on Freebase and compare HERO with representative replay-based methods. As shown in Table 4, HERO requires 7,046 MB of GPU memory per task on average and reaches a peak of 8,107 MB, both slightly lower than the compared replay baselines. We further remove HKD to isolate the overhead introduced by teacher-based distillation. The average memory decreases from 7,046 MB to 7,013 MB, while Finetune requires 6,973 MB, indicating that HKD introduces only modest runtime memory overhead in the evaluated setting.

Nevertheless, retaining all historical teachers leads to a model storage cost of $O(T)$ as the number of tasks T grows, even when the size of each teacher is constant. To avoid this linear growth, we maintain only the most recent k teachers using a sliding window, reducing teacher storage to $O(k)$. Table 5 shows the results on Yelp under 5-task and 8-task settings. Keeping three teachers provides a favorable efficiency–performance trade-off: relative to retaining all teachers, $k = 3$ reduces AP by only 0.61 and 0.38 points in the 5-task and 8-task settings, while AF increases by 1.14 and 0.55 points. In exchange, teacher storage changes from $O(T)$ to $O(1)$ for fixed k. These results suggest that a small fixed teacher window can retain most of the benefit of multi-teacher distillation while keeping the additional model storage bounded. Overall, HERO incurs little additional runtime memory in the evaluated setting, while the fixed teacher window prevents its teacher-storage cost from increasing with the number of tasks.

Training Time. Based on the HAN backbone, we evaluate recent continual graph learning methods and the proposed HERO framework on four datasets using a single RTX 3080Ti GPU. Table 6 reports the average training time per task. Overall, regularization-based methods such as TWP are more efficient than replay-based approaches, highlighting a known efficiency gap in replay-driven continual learning. Addressing this gap in heterogeneous settings remains an open problem. Despite relying on replay, HERO is consistently faster than other replay-based methods, showing clear gains in training efficiency. Although TWP is slightly more efficient in terms of runtime, it underperforms HERO across all benchmarks with respect to accuracy and forgetting. These measurements show manageable overhead in the evaluated settings.

Table 6: Training time (s) comparison on four datasets. Lower is better. Numbers in parentheses denote the ratio relative to HERO.

Method	DBLP	IMDB	Freebase	Yelp
TWP	0.157 (x0.42)	0.032 (x0.78)	0.068 (x0.59)	3.020 (x0.50)
FTF-ER	1.619 (x4.39)	0.238 (x5.78)	0.351 (x3.06)	8.424 (x1.39)
ER-GNN	0.411 (x1.11)	0.053 (x1.29)	0.078 (x0.68)	6.652 (x1.10)
MetaCLGraph	1.770 (x4.80)	0.310 (x7.53)	0.614 (x5.34)	9.752 (x1.61)
HERO	0.369 (x1.00)	0.041 (x1.00)	0.115 (x1.00)	6.043 (x1.00)

D Additional Results

D.1 Results on Different Backbones

Additional experiment results on other two backbones: HGT (Hu et al., 2020) and MAGNN (Fu et al., 2020) as shown in Table 7 and 8.

Table 7: Performance comparison on three benchmark datasets using HGT as the backbone. The symbol ↑ (↓) indicates that higher (lower) is better. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	DBLP		IMDB		Freebase	
	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓
Finetune	86.2 ± 1.5	12.7 ± 2.2	75.1 ± 7.5	17.0 ± 9.4	67.8 ± 1.8	16.5 ± 2.5
Joint Train	93.9 ± 0.6	1.7 ± 1.0	80.0 ± 0.7	2.0 ± 0.9	72.5 ± 0.6	8.5 ± 1.2
EWC (Kirkpatrick et al., 2017)	89.1 ± 1.2	10.7 ± 1.8	77.3 ± 0.4	12.5 ± 0.2	69.6 ± 3.0	14.7 ± 2.9
MAS (Aljundi et al., 2018)	89.8 ± 2.5	9.0 ± 3.0	77.5 ± 1.2	11.8 ± 2.6	70.4 ± 3.1	14.8 ± 2.0
TWP (Liu et al., 2021)	89.1 ± 2.1	10.3 ± 1.1	77.7 ± 0.6	11.1 ± 1.6	69.6 ± 2.0	15.1 ± 0.9
ER-GNN (Zhou & Cao, 2021)	90.7 ± 1.5	3.5 ± 0.4	77.3 ± 2.2	9.0 ± 3.6	69.5 ± 1.4	13.6 ± 1.7
MetaCLGraph (Unal et al., 2023)	90.0 ± 0.7	10.9 ± 1.4	76.1 ± 0.3	12.2 ± 1.8	69.1 ± 2.2	16.3 ± 2.5
FTF-ER (Pang et al., 2024)	89.4 ± 2.2	5.5 ± 1.9	76.1 ± 1.4	8.9 ± 3.5	69.7 ± 2.8	13.0 ± 2.5
DMSG (Qiao et al., 2025)	88.2 ± 1.9	11.3 ± 2.4	76.1 ± 0.4	12.1 ± 1.5	64.6 ± 0.6	16.4 ± 2.7
HERO	**92.8 ± 0.8**	**3.3 ± 1.2**	**78.4 ± 0.4**	**6.6 ± 1.7**	**70.6 ± 1.2**	**9.1 ± 2.7**

Table 8: Performance comparison on DBLP and IMDB using MAGNN as the backbone. MAGNN runs out of memory on Freebase and is therefore not reported. The symbol ↑ (↓) indicates that higher (lower) is better. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	DBLP		IMDB	
	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓
Finetune	79.1 ± 5.1	34.8 ± 10.1	71.3 ± 0.6	21.7 ± 3.0
Joint Train	94.1 ± 0.1	1.9 ± 1.0	77.4 ± 0.7	4.7 ± 1.5
EWC (Kirkpatrick et al., 2017)	91.0 ± 2.3	10.9 ± 4.5	73.4 ± 1.6	16.7 ± 2.5
MAS (Aljundi et al., 2018)	92.1 ± 2.1	7.8 ± 5.3	74.3 ± 2.0	14.2 ± 3.7
TWP (Liu et al., 2021)	91.6 ± 2.1	7.4 ± 5.0	74.0 ± 0.8	13.7 ± 3.3
ER-GNN (Zhou & Cao, 2021)	90.0 ± 2.1	12.4 ± 4.3	75.3 ± 0.9	10.3 ± 1.5
MetaCLGraph (Unal et al., 2023)	92.3 ± 0.4	2.1 ± 0.7	75.6 ± 1.0	11.8 ± 1.3
FTF-ER (Pang et al., 2024)	92.8 ± 0.8	**1.9 ± 2.3**	73.0 ± 0.6	15.8 ± 1.4
DMSG (Qiao et al., 2025)	90.6 ± 1.1	8.6 ± 2.2	74.5 ± 0.5	8.8 ± 2.7
HERO	**93.2 ± 0.5**	3.1 ± 0.8	**76.8 ± 1.1**	**7.5 ± 1.4**

D.2 Reordered Task Sequence

We further evaluate robustness under task-order variations by training with randomly permuted task sequences on DBLP, IMDB, and Freebase. Results is reported in Table 9. Under the tested random task orders, HERO improves AP over the strongest baseline by 0.9 points on DBLP, 2.1 points on IMDB, and 5.1 points on Freebase. HERO also obtains the lowest AF on IMDB and Freebase, while FTF-ER gives a lower AF on DBLP. These results show that HERO's AP advantage persists under the evaluated task permutations.

Table 9: Performance comparison under reordered task sequences on three benchmark datasets using HAN as the backbone. The symbol ↑ (↓) indicates that higher (lower) is better. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	DBLP		IMDB		Freebase	
	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓	AP / % ↑	AF / % ↓
Finetune	83.5 ± 0.9	24.3 ± 2.1	56.7 ± 0.7	49.8 ± 1.3	49.2 ± 2.3	31.8 ± 5.7
EWC (Kirkpatrick et al., 2017)	90.7 ± 1.1	10.5 ± 2.5	67.8 ± 2.5	24.0 ± 2.6	54.0 ± 11.0	24.2 ± 20.1
MAS (Aljundi et al., 2018)	92.6 ± 2.5	4.8 ± 5.3	72.0 ± 4.9	13.0 ± 10.7	47.1 ± 0.6	34.6 ± 2.3
TWP (Liu et al., 2021)	93.6 ± 1.1	3.5 ± 1.5	73.4 ± 0.5	11.2 ± 4.9	52.0 ± 6.8	26.4 ± 9.9
ER-GNN (Zhou & Cao, 2021)	93.2 ± 2.7	6.1 ± 5.4	69.7 ± 2.7	13.6 ± 3.2	56.0 ± 3.9	17.7 ± 7.5
FTF-ER (Pang et al., 2024)	94.6 ± 0.5	**1.1 ± 0.5**	67.2 ± 4.7	30.4 ± 3.9	45.1 ± 1.1	35.6 ± 0.7
MetaCLGraph (Unal et al., 2023)	94.7 ± 0.5	3.4 ± 0.8	65.6 ± 3.6	26.9 ± 4.8	50.1 ± 2.1	26.2 ± 4.5
HERO	**95.6 ± 0.6**	2.1 ± 2.6	**75.5 ± 0.1**	**8.0 ± 0.3**	**61.1 ± 1.0**	**5.1 ± 1.5**

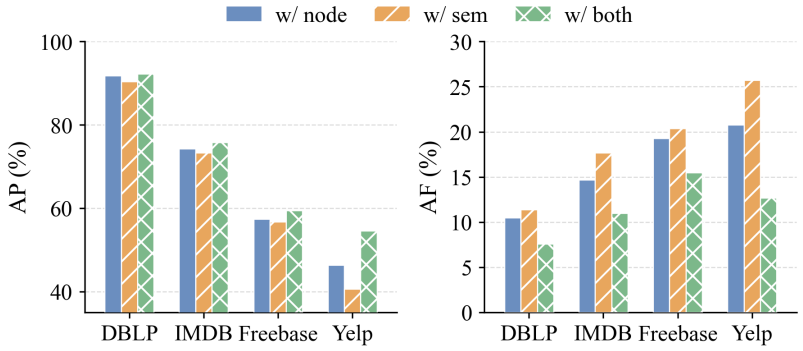


Figure 6: Ablation Study on Heterogeneity-aware Knowledge Distillation Module on all datasets.

D.3 Further Ablation Study on HKD

We further analyze the contribution of the two submodules in the HKD module: logit-level distillation and semantic-level distillation to overall model performance, and assess the effectiveness of our proposed design. Figure 6 illustrates the impact of different distillation strategies—node-level only ("with node"), semantic-level only ("with sem"), and the combination of both ("with both")—on the model's Average Performance (AP) and Average Forgetting (AF) across four datasets (DBLP, IMDB, Freebase, Yelp). The results show that the combined strategy ("with both") consistently achieves the highest AP and lowest AF across all datasets, demonstrating that logit-level and semantic-level distillation complement each other. Semantic-level distillation alone ("with sem") exhibits worse forgetting suppression on Freebase and Yelp, suggesting it may struggle to independently capture task-level semantic changes. On complex heterogeneous graphs such as Yelp, the synergy of both distillation levels yields especially significant performance improvements, indicating that addressing both structural and semantic forgetting is crucial.

D.4 Forgetting-aware Gap

AP (Average Performance) measures the mean test accuracy across all previously learned tasks, while AF (Average Forgetting) quantifies the performance drop on a specific task after learning subsequent ones. However, we observe that in order to retain knowledge from earlier tasks, the model often compromises its performance on the final task. To assess this trade-off, we introduce a new metric, **Forgetting-aware Gap** (FaG), defined as the difference between the test accuracy obtained by training the model solely on the final task and the test accuracy on the same task after completing all tasks under the continual learning framework:

$$\text{FaG} = a_T^{FT} - a_T^{CL} \tag{19}$$

where a_T denotes the test accuracy on the final task $\mathcal{T}_T$, with "FT" referring to the finetuning baseline (i.e., training only on task $\mathcal{T}_T$), and "CL" referring to the accuracy obtained after training with a continual learning method. This gap reflects the performance degradation caused by the inherent

plasticity-stability trade-off in continual learning. Due to limited space, we provide experiment results in Appendix D.4.

Table 10: Forgetting-aware Gap (FaG, %) on four benchmark datasets, averaged over five runs. Lower values indicate better performance. The best results are highlighted in **bold**, while the second-best results are underlined.

Methods	DBLP	IMDB	Freebase	Yelp
Other				
Joint Train	0.69	2.47	11.59	11.58
TWP	1.03	2.27	10.44	**5.58**
Memory Replay				
ER-GNN	**0.22**	2.62	**8.44**	8.46
FTF-ER	0.83	3.43	10.59	7.19
MetaCLGraph	1.36	2.87	13.68	13.61
DMSG	0.89	3.76	11.87	15.57
HERO	0.63	**1.31**	9.90	6.88

As shown in Table 10, HERO achieves a consistently competitive forgetting-aware gap across different datasets. It obtains the lowest FaG on IMDB and the second-lowest results on DBLP, Freebase, and Yelp. These results suggest that HERO maintains a favorable balance between retaining previously learned knowledge and adapting to new tasks, rather than improving average performance at the cost of excessive forgetting.

D.5 Additional Parameter Analysis and Visualizations

We provide additional sensitivity analyses for two hyperparameters: the distillation loss weight λ_{kd} and the shot number as shown in Figure 7 and 8.

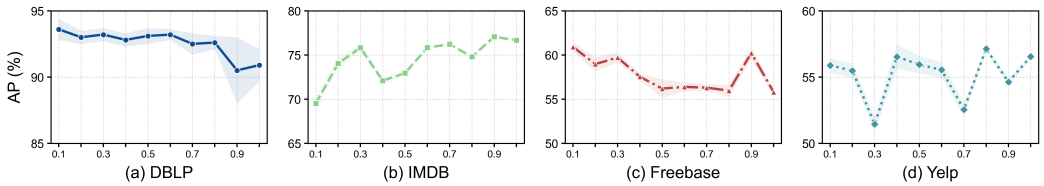


Figure 7: Effect of the distillation loss weight λ_{kd} across four datasets.

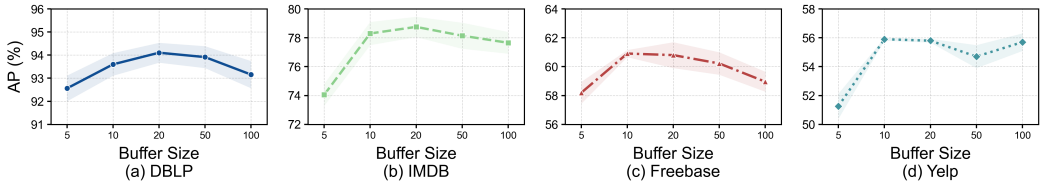


Figure 8: Effect of the buffer size across four datasets.

E Extended Related Work

Heterogeneous Graph Neural Networks. Heterogeneous Graph Neural Networks (HGNNs) are designed to model graphs containing multiple node and edge types by encoding type-dependent structures and semantics. Representative HGNNs include metapath-guided methods, which aggregate information along predefined or learned semantic paths (Fu et al., 2020; Zhang et al., 2019; Wang et al., 2019; Schlichtkrull et al., 2018; Dong et al., 2017; Sun et al., 2011; Yun et al., 2019; Yang et al., 2023; Zhu et al., 2024), and metapath-free methods, which directly model interactions between node and relation types (Hu et al., 2020; Lv et al., 2021; Zhu et al., 2019; Hong et al., 2020; Zhang et al., 2022b; Fu et al., 2023). These methods have shown strong performance in extracting heterogeneous structural and semantic information. Recent work such as G-Substrate Li et al. (2026) further explores reusable graph representations across different data modalities and learning contexts, highlighting the value of preserving transferable structural knowledge.

Most existing HGNN studies adopt an offline learning protocol in which the training graph is jointly accessible. Under continual learning, however, an HGNN must update its parameters using the current task and a bounded historical memory. This creates a different preservation problem because the prediction of a target node depends not only on its own feature and label, but also on typed neighbors, relation-specific transformations, and semantic aggregation channels. Our work does not introduce a new HGNN encoder. Instead, it studies how representative HGNN backbones can retain these heterogeneous computation patterns when learned over a sequence of class-disjoint tasks.

Continual Graph Learning. Continual Graph Learning (CGL) studies how graph models learn from sequentially presented data while reducing catastrophic forgetting (Zhang et al., 2022c; 2024b). Existing methods can be broadly grouped into parameter regularization or isolation, experience replay, and knowledge distillation. Regularization- and parameter-isolation-based methods protect knowledge by restricting updates to parameters estimated to be important for previous tasks (Zhang et al., 2022d; 2023a; Kirkpatrick et al., 2017; Aljundi et al., 2018; Liu et al., 2021). Some graph-specific methods further incorporate topological information when estimating parameter importance. However, in an HGNN, the same parameter may play different roles for different node types, relation types, or semantic channels. A globally aggregated importance score does not directly represent such type- or relation-conditioned roles. It may therefore over-constrain shared parameters or provide insufficient protection for low-frequency heterogeneous patterns. Experience-replay methods retain historical nodes, embeddings, or subgraphs and revisit them when learning subsequent tasks (Zhang et al., 2023b; Ahrabian et al., 2021; Zhou & Cao, 2021; Zhang et al., 2022e; Choi et al., 2024; Zhang et al., 2024a; Hoang et al., 2023; Han et al., 2024; Song et al., 2025). These methods have shown that data selection and topology preservation are important for graph continual learning. Nevertheless, replay in heterogeneous graphs introduces an additional context-reconstruction problem. A target node with the same feature and label can produce different representations when its typed neighbors or relation-specific connections change. Storing isolated target nodes may therefore fail to reproduce the heterogeneous computation that generated the historical training signal. On the other hand, retaining complete heterogeneous neighborhoods can lead to substantial memory and computation costs. DiSCo addresses this trade-off by combining representative target-node selection with bounded multi-type neighbor expansion. Distillation-based graph continual learning methods preserve historical knowledge by aligning model predictions or representations across learning stages (Xu et al., 2020; Dong et al., 2021; Qiu et al., 2025). Such methods provide useful functional constraints, but output or embedding alignment does not explicitly determine which relation- or metapath-specific semantic channels produce the prediction. Two HGNNs may generate similar outputs while assigning different importance to heterogeneous semantic contexts. HERO therefore combines logit-level distillation with semantic-level alignment to constrain both historical predictions and heterogeneous semantic responses. Different from settings that explicitly introduce new node or relation types, our protocol retains a fixed heterogeneous schema and partitions target classes into a sequence of disjoint tasks. The main challenge studied here is thus the preservation of type-dependent and relation-dependent computation under sequential parameter updates and bounded historical access.

Meta-Learning. Meta-learning aims to improve adaptation to new tasks by transferring knowledge across related learning episodes. Existing approaches are commonly divided into optimization-based methods, such as MAML (Finn et al., 2017) and Reptile (Nichol & Schulman, 2018), and metric-based methods, such as Prototypical Networks (Snell et al., 2017) and Matching Networks (Vinyals et al., 2016). Optimization-based methods typically learn an initialization or update mechanism from which a model can adapt using a small number of gradient steps. Meta-learning has also been studied in continual learning. La-MAML (Gupta et al., 2020) uses look-ahead optimization to account for the influence of current updates on historical examples, while representation-based approaches learn features that remain adaptable across sequential tasks (Javed & White, 2019). In graph learning, MetaCLGraph (Unal et al., 2023) combines meta-learning with experience replay for continual graph learning, whereas HG-Meta (Zhang et al., 2022a) studies few-shot adaptation on heterogeneous graphs. HERO follows the look-ahead optimization principle but focuses on heterogeneous continual node classification. Its adaptation module uses a small representative support graph to obtain a fast update, and evaluates this update using a joint objective containing the current-task, replay, and distillation constraints. Thus, meta-adaptation primarily improves plasticity on the incoming task, while DiSCo and heterogeneity-aware distillation are responsible for historical knowledge retention.