DYNAMIC GRAPH WORLD MODELLING: LEARNING FUTURE DYNAMICS FROM TEMPORAL CONTEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic graph learning models how entities and their relations evolve from historical observations to predict future graph states. Existing methods do not fully address dynamic graph world modelling, which requires learning transitions from temporal context to future latent graph states. This is difficult for three reasons: nodes and relations rely on different historical evidence, structural evolution is path-dependent, and future snapshots provide sparse and realization-specific targets. We propose **DY**namic **G**raph **W**orld **M**odelling (**DYG-WM**), the first framework to jointly predict future node and relation representations from temporal context by answering three questions: *what has evolved, how it evolved, and what will evolve next*. To capture (a) *what has evolved*, a historical node-relation context encoder summarizes node-neighborhood trajectories and pair-specific histories. To characterize (b) *how it evolved*, a node-relation path-signature encoder captures ordered structural changes. To predict (c) *what will evolve next*, a dual-level joint-embedding objective predicts target-snapshot node and relation representations without reconstructing the raw graph. Experiments on 12 link-prediction and two node-classification datasets demonstrate the effectiveness of DYG-WM across both tasks.

1 INTRODUCTION

Many real-world systems change through the evolution of both entities and their relationships. Dynamic graphs model these systems as sequences of graph-structured states, in which nodes represent entities and time-varying edges record their interactions (Gravina et al., 2024; Feng et al., 2025b; Skarding et al., 2021). They have been widely used to study social networks (Kumar et al., 2019; Zheng et al., 2026), recommender systems (Zhang et al., 2026a; Wang et al., 2026), traffic systems (Zou et al., 2026; Weng et al., 2023), financial transactions (Song et al., 2023; Yuan et al., 2025), and scientific discovery (Bai et al., 2024). By modelling structural dependencies within graph states and temporal dependencies across states, dynamic graph neural networks (DGNNs) can do temporal link prediction and node attribute prediction (Kumar et al., 2019; Rossi et al., 2020; Yu et al., 2023; Li et al., 2026). Although current models perform well on individual prediction tasks, real-world applications require more than forecasting a single edge or node label. Dynamic graph learning aims to model how the graph state evolves from historical observations so that future states can be predicted and used for planning and decision-making.

This motivates us to formulate the problem of **dynamic graph world modelling**, which extends the general world-modelling paradigm to evolving relational systems. As illustrated in Figure 1, historical graph observations are encoded into compact latent states, and the temporal transitions between these states are learned to predict future system dynamics. From this perspective, existing dynamic graph learning methods can be broadly divided into two groups, neither of which fully addresses this problem. *(1) Task-oriented temporal graph representation learning* trains DGNNs with node labels or observed interactions to predict specific outcomes, such as future links or node attributes (Kumar et al., 2019; Rossi et al., 2020; Yu et al., 2023; Li et al., 2026). Although effective on individual tasks, such supervision is often sparse, delayed, or costly to obtain and binds the learned representations to particular tasks and evolving environments. *(2) Self-supervised dynamic graph representation learning* reduces this dependence through contrastive, reconstruction-based, or latent-predictive objectives. Contrastive methods align temporal views, sampled subgraphs, or interacting nodes (Wang et al., 2021a; Xu et al., 2023). However, they mainly enforce invariance

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.

across constructed views, and their effectiveness depends on whether the selected positive-negative pairs preserve temporal semantics. Reconstruction and masking methods recover missing edges, attributes, or temporal observations (Gao et al., 2025; He et al., 2025), but encourage the model to focus on low-level details of the observed graph rather than its predictable dynamics. Dynamic graphs have a natural temporal prediction structure: past graph states provide the context, while future graph states serve as prediction targets. Predictive latent representation learning is therefore well suited to dynamic graph world modelling because temporal progression naturally separates context from target. The goal is to predict, from historical states, what the graph will become in representation space rather than reconstruct every observed interaction.

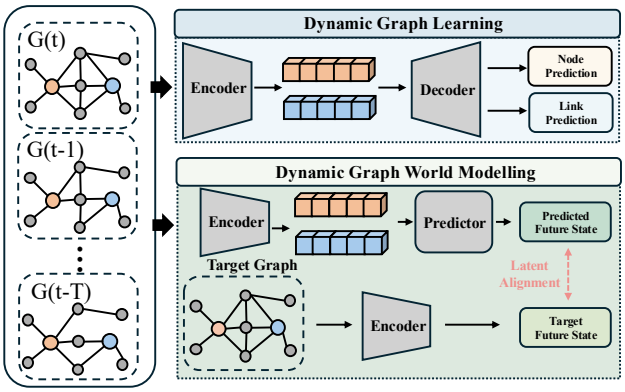


Figure 1: Comparison of dynamic graph learning and dynamic graph world modelling. Dynamic graph learning maps historical graphs to task-specific outputs, whereas dynamic graph world modelling predicts future latent state.

However, learning a dynamic graph world model involves three challenges. **C1: Multi-level historical dependency.** Nodes and relations evolve together but draw on different historical evidence. Node evolution depends on the trajectory of a node and its neighborhood, whereas relation evolution also depends on pair-specific structural histories that endpoint representations alone cannot recover. **C2: Path-dependent structural evolution.** Similar current graph states can result from different sequences of structural changes and may therefore lead to different future dynamics. A world model must retain both the resulting state and the ordered path that produced it. **C3: Sparse and realization-specific future targets.** A future graph snapshot records only the node states and interactions observed within a given time window, offering an incomplete view of the underlying dynamics. Predicting these outcomes separately or reconstructing the snapshot exactly may lead the model to fit a single observed realization rather than capture how nodes and relations evolve more generally. A dynamic graph world model therefore needs a compact future-state target that goes beyond individual node labels and edge observations.

To address these challenges, we introduce **DyG-WM**, a **Dy**namic **G**raph **W**orld **M**odelling method that learns future graph dynamics directly in the latent space through predictive representation learning guided by temporal context. Our proposed DyG-WM is designed and organized around three questions: *what has evolved, how it evolved, and what will evolve next*. To capture *what has evolved*, we develop **(a) a historical node-relation context encoder** that summarizes the trajectories of nodes and their neighborhoods and builds pair-specific structural contexts. This provides the different forms of historical evidence described in C1. To characterize *how the graph evolved*, we introduce **(b) a node-relation path-signature encoder** that captures the order of structural changes at both levels. It distinguishes trajectories that reach similar current states through different transition orders, addressing the path dependence described in C2. Finally, to predict *what will evolve next*, we design **(c) a dual-level joint-embedding predictive objective** that uses historical contexts and path signatures to predict node and relation representations from the target snapshot. By learning future dynamics in representation space instead of reconstructing a single future graph realization, the objective addresses C3. Our main contributions are as follows:

- To our knowledge, DyG-WM is the first framework to formulate dynamic graph world modelling as the joint prediction of future node and relation representations. It treats relations as temporally evolving latent states rather than only as message-passing channels or endpoint-based scores. Using strictly historical context, DyG-WM learns future dynamics directly in representation space.

- To account for path dependence in dynamic graph evolution, we combine a historical node-relation context encoder with a node-relation path-signature encoder. The two encoders represent accumulated states and ordered structural changes, respectively. A dual-level joint-embedding

predictive objective uses these encodings to predict future node and relation states without relying on augmented-view contrast or raw-graph reconstruction for self-supervision.

- Across 12 dynamic link-prediction datasets and two dynamic node-classification datasets, DyG-WM achieves the best average rank in every evaluated link-prediction setting. It also obtains the best Macro-F1 and Micro-F1 scores across all label ratios. Ablation and latent-alignment analyses further confirm the effectiveness of the individual components.

2 Related Works

Dynamic Graph Neural Networks. Dynamic graph neural networks (DGNNs) are commonly divided into discrete-time models, which operate on graph snapshots, and continuous-time models, which process timestamped interactions directly. Representative methods capture graph evolution through temporal attention (Sankar et al., 2020), recurrent updates (Pareja et al., 2020), memory mechanisms (Rossi et al., 2020), temporal walks (Wang et al., 2021b), MLP-based temporal mixing (Cong et al., 2023), Transformer-based sequence modeling (Yu et al., 2023), and state-space models (Li et al., 2026). Recent self-supervised DGNNs learn from temporal contrast, masked reconstruction, or future latent prediction (Chen et al., 2025; Gao et al., 2025; He et al., 2025; Zhang et al., 2026b). However, most remain node-centric, while pair-level information is mainly used for link prediction (Wang et al., 2021b;a; Yu et al., 2023; Tian et al., 2024). DyG-WM instead models and predicts future representations at both node and relation levels.

Graph World Model. World models learn latent states and transition functions to predict future dynamics (Ha & Schmidhuber, 2018). Graph world models extend this idea to structured environments by representing states as entities and relations, with a graph abstraction and a relational transition model (Battaglia et al., 2018; Liu et al., 2026). Existing object-centric models use graph neural networks to predict interactions and future object states, including C-SWM (Kipf et al., 2020), G-SWM (Lin et al., 2020), CWM (Li et al., 2020a), VGPL (Li et al., 2020b), and other graph-based world models (Feng et al., 2025a; Zhang et al., 2021; Zhu et al., 2022). These methods mainly target action-conditioned physical systems, where relations support object-state updates rather than being predicted themselves. DyG-WM instead models general dynamic graphs by constructing historical node and relation contexts and jointly predicting their future representations. It can therefore be viewed as an action-free relational transition model rather than a full planning system.

3 Methodology

3.1 Preliminary

Notation. Let $\mathcal{V}$ be a global node set with $N = |\mathcal{V}|$. A dynamic graph is a time-ordered interaction stream $\mathcal{E} = \big((u_i, v_i, \tau_i, \mathbf{e}_i)\big)_{i=1}^{M}$ with $\tau_i \le \tau_{i+1}$, where ordered node pairs may repeat. We partition $\mathcal{E}$ into T consecutive subsequences $\{\mathcal{E}_t\}_{t=1}^{T}$ with roughly equal event counts, keeping equal-timestamp interactions in the same bin. Each bin has two aligned views. The snapshot view is $\widetilde{G}_t = (\mathcal{V}, A_t, X_t, \mathbf{m}_t)$, where $A_t \in \{0,1\}^{N\times N}$ is the de-duplicated, symmetrized message-passing adjacency, $X_t \in \mathbb{R}^{N\times F}$ is the node-attribute matrix, and $\mathbf{m}_t \in \{0,1\}^N$ is the active-node mask. We define $\mathcal{V}_t = \{u \in \mathcal{V} : m_u^t = 1\}$ and write $\mathcal{N}_t(u)$ for the neighbors of u in A_t. Both A_t and X_t remain globally indexed. Time-varying components of X_t summarize events through bin t, with global preprocessing statistics computed from the training split only. The event view $\mathcal{E}_t$ preserves directed interactions, repetitions, exact timestamps, and link features. For query time τ, the strict history is $\mathcal{H}_{<\tau} = \big((u_i, v_i, \tau_i, \mathbf{e}_i) : \tau_i < \tau\big)$. Equal-timestamp events are simultaneous and excluded from one another's histories. Superscript t denotes bin-level indexing, whereas τ or $< \tau$ denotes exact-time dependence.

Problem Formalization. For target bin t, let $\mathcal{C}_t = (\widetilde{G}_{t-W+1}, \ldots, \widetilde{G}_{t-1})$ contain the preceding $W-1$ snapshot views and no event information from bin t. From $\mathcal{C}_t$, the model computes a node context $\mathbf{c}_u^t$, which, despite its index t, depends only on snapshots before the target bin. For query (u, v, τ) in bin t, the model combines $\mathcal{C}_t$ and $\mathcal{H}_{<\tau}$ to obtain the exact-time relation representation $\mathbf{z}_{uv}^{\tau}$. Dynamic link prediction scores an indexed candidate group $\mathcal{Q}(u, \tau)$ containing one observed

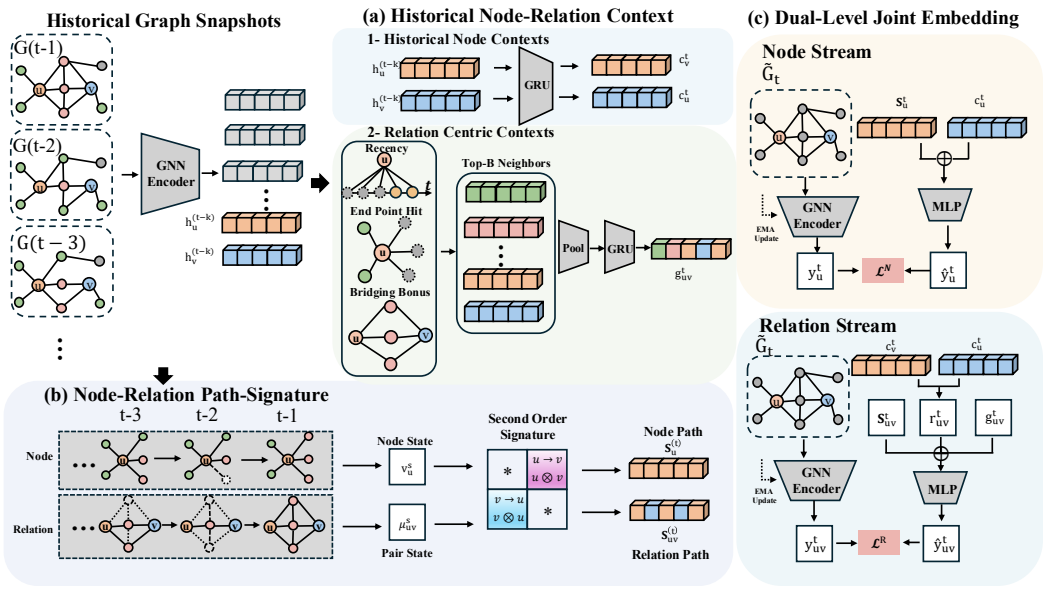


Figure 2: Overview of DyG-WM. (a) The historical node-relation context encoder summarizes the trajectories of nodes and their neighborhoods and constructs pair-specific relation contexts from historical snapshots. (b) The node-relation path-signature encoder uses second-order signatures to capture ordered structural changes at both levels. (c) The dual-level joint-embedding predictive objective predicts target-snapshot node and relation representations.

interaction and K sampled negatives; sampled destinations that coincide with the observed destination are retained rather than deduplicated.

Given the historical snapshot context $\mathcal{C}_t$ and strict event history $\mathcal{H}_{<\tau}$, we learn a shared predictive representation for node classification and dynamic link prediction, while keeping the target snapshot $\widetilde{G}_t$ hidden from the online branch.

The overall framework of DyG-WM is shown in Figure 2. DyG-WM consists of three main components. The historical node-relation context encoder summarizes node trajectories and constructs pair-specific relation contexts from preceding snapshots. The node-relation path-signature encoder models ordered structural changes at both the node and relation levels. The dual-level joint-embedding predictive objective predicts target-snapshot node and relation representations using an EMA target encoder. Given the historical snapshot context $\mathcal{C}_t$ and the strict event prefix $\mathcal{H}_{<\tau}$ for a query (u, v, τ), DyG-WM constructs a relation representation and produces the interaction score s^{τ}_{uv}. The following sections describe these components in detail.

3.2 Historical Node-Relation Context Encoder

Predicting future dynamics requires historical information at both the node and relation levels. Node prediction tracks how a node and its neighborhood evolve over time, while relation prediction also depends on pair-specific structural history that is not captured by the endpoint representations alone. We use a (a-1) historical node context encoder for node-level evolution and a (a-2) relation-centric context encoder for pair-specific structural history.

Historical Node Context Encoder. For target bin t, let $\mathcal{I}_t = \{t - W + 1, \ldots, t - 1\}$ index the historical snapshots in $\mathcal{C}_t$. Each node representation is obtained from the online GNN as

$$\mathbf{h}^s_u = \mathrm{GNN}_\theta(\widetilde{G}_s)[u], \qquad s \in \mathcal{I}_t. \tag{1}$$

The latest representation $\mathbf{h}_u^{t-1}$ alone does not capture the earlier evolution of node u or its neighborhood. We therefore compute the mean neighborhood representation

$$\overline{\mathbf{h}}_u^s = \frac{1}{\max\{1, |\mathcal{N}_s(u)|\}} \sum_{w \in \mathcal{N}_s(u)} \mathbf{h}_w^s, \tag{2}$$

with $\overline{\mathbf{h}}_u^s = \mathbf{0}$ when $\mathcal{N}_s(u) = \varnothing$.

The historical node context encoder processes the ordered node and neighborhood trajectories with separate recurrent encoders and combines them with the latest node representation through gated residual fusion:

$$\mathbf{c}_u^t = f_{\text{NodeC}}\left(\mathbf{h}_u^{t-1}, (\mathbf{h}_u^s)_{s \in \mathcal{I}_t}, (\overline{\mathbf{h}}_u^s)_{s \in \mathcal{I}_t}\right). \tag{3}$$

Thus, $\mathbf{c}_u^t$ summarizes the recent state and historical node-neighborhood evolution of u using only snapshots $s < t$.

Relation-Centric Context Encoder. For a queried pair (u, v), we collect the endpoints and their historical neighbors over $\mathcal{I}_t$. The endpoints are retained, while the remaining candidates are ranked using recency and common-neighbor bridging information. We keep the top B context nodes as $\mathcal{K}_{uv}^t$, where B is the relation-context budget. Their representations are pooled at each historical snapshot and summarized temporally:

$$\mathbf{g}_{uv}^t = \text{GRU}_{\text{rel}}\left(\left(\frac{1}{|\mathcal{K}_{uv}^t|} \sum_{w \in \mathcal{K}_{uv}^t} \mathbf{h}_w^s\right)_{s \in \mathcal{I}_t}\right). \tag{4}$$

The resulting $\mathbf{g}_{uv}^t$ captures pair-specific structural history using only $\mathcal{C}_t$.

3.3 Node-Relation Path-Signature Encoder

Dynamic graph evolution is path-dependent: similar current states may arise from different historical trajectories and lead to different future dynamics. Although $\mathbf{c}_u^t$ and $\mathbf{g}_{uv}^t$ summarize the states in $\mathcal{C}_t$, recurrent summaries do not explicitly preserve the order and interactions of structural changes. Node and relation prediction also require different trajectories. Node prediction depends on local neighborhood evolution, while relation prediction additionally depends on the joint evolution of the queried pair. We therefore use a node structural path-signature encoder and a relation structural path-signature encoder to model these ordered changes (Chevyrev & Kormilitzin, 2025).

Second-Order Path Signatures. Let $\mathbf{d}_1, \ldots, \mathbf{d}_L \in \mathbb{R}^{d_s}$ be a sequence of projected path increments, where $L = W - 1$. We represent the ordered path using the truncated second-order signature

$$\text{Sig}_2(\mathbf{d}_{1:L}) = \left[\sum_{i=1}^{L} \mathbf{d}_i \;\Big\|\; \text{vec}\left(\sum_{1 \le i < j \le L} \mathbf{d}_i \otimes \mathbf{d}_j + \frac{1}{2}\sum_{i=1}^{L} \mathbf{d}_i \otimes \mathbf{d}_i\right)\right] \tag{5}$$

The first component summarizes cumulative change, while the second captures ordered interactions between structural changes through the terms with $i < j$.

Node Structural Paths. For each $s \in \mathcal{I}_t$, we construct a structural descriptor $\boldsymbol{\nu}_u^s \in \mathbb{R}^{d_N}$ from the normalized degree of u, its signed degree change, the proportions of arriving, disappearing, and persistent neighbors, and an activity indicator. For the first historical snapshot $s_0 = t - W + 1$, transition-dependent entries are set to zero; subsequent transitions use snapshots $s - 1$ and s.

We append the normalized elapsed time $\overline{\Delta t}_s$, project each descriptor to the path space, and apply the second-order signature:

$$\mathbf{S}_u^t = \phi_N\left(\text{Sig}_2\left(\left(\text{Proj}_N[\overline{\Delta t}_s \| \boldsymbol{\nu}_u^s]\right)_{s \in \mathcal{I}_t}\right)\right). \tag{6}$$

Here, ϕ_N maps the truncated signature to the model dimension. While $\mathbf{c}_u^t$ summarizes the historical state of u, $\mathbf{S}_u^t$ captures its ordered structural evolution.

Relation Structural Paths. For each queried pair (u, v) and historical snapshot s, we construct a pair state $\boldsymbol{\mu}_{uv}^s \in \mathbb{R}^{d_R}$ from direct connectivity, normalized endpoint degrees, normalized common-neighbor count, neighborhood Jaccard overlap, endpoint activity, and coverage of the selected context nodes $\mathcal{K}_{uv}^t$. Endpoint-specific channels remain ordered to preserve the roles of u and v.

We describe the relation trajectory by changes in these pair states:

$$\Delta\boldsymbol{\mu}_{uv}^{s} = \begin{cases} \boldsymbol{\mu}_{uv}^{s}, & s = t - W + 1, \\ \boldsymbol{\mu}_{uv}^{s} - \boldsymbol{\mu}_{uv}^{s-1}, & s > t - W + 1. \end{cases} \tag{7}$$

After appending the normalized elapsed time and projecting each increment to the path space, we apply the second-order signature:

$$\mathbf{S}_{uv}^{t} = \phi_R\left(\mathrm{Sig}_2\left(\left(\mathrm{Proj}_R[\overline{\Delta t}_s \| \Delta\boldsymbol{\mu}_{uv}^{s}]\right)_{s\in\mathcal{I}_t}\right)\right). \tag{8}$$

Here, $\mathbf{S}_{uv}^{t}$ captures the ordered structural evolution of (u, v) using only $\mathcal{C}_t$. Together, $\mathbf{S}_{u}^{t}$ and $\mathbf{S}_{uv}^{t}$ describe ordered node-level and pair-level structural changes that complement $\mathbf{c}_{u}^{t}$ and $\mathbf{g}_{uv}^{t}$.

3.4 Dual-Level Joint-Embedding Predictive Objective

Future observations in dynamic graphs are sparse and variable, so directly predicting individual links or node labels can overfit to a particular realization of the future graph. We instead predict future latent states, which capture structural changes without reconstructing the exact target snapshot. Since these changes occur at both node and pair levels, we use a dual-level joint-embedding predictive objective for future node and relation states. We therefore predict future latent representations at both levels instead of reconstructing target adjacency or raw attributes.

Online and Target Branches. We use GNN_θ and g_θ as the online snapshot and relation encoders, with EMA target counterparts $\mathrm{GNN}_{\bar\theta}$ and $g_{\bar\theta}$. The online branch uses only historical information, while the target branch encodes $\widetilde{G}_t$ during training. The node target is $\mathbf{y}_{u}^{t} = \mathrm{sg}\left(\mathrm{GNN}_{\bar\theta}(\widetilde{G}_t)[u]\right), u \in \mathcal{V}_t$, where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The target parameters receive no gradients and are updated by EMA: $\bar\theta \leftarrow \eta_{\mathrm{ema}}\bar\theta + (1-\eta_{\mathrm{ema}})\theta$.

Node-Level Predictive Objective. The node predictor uses the historical context, path signature, and prediction horizon: $\widehat{\mathbf{y}}_{u}^{t} = p_N\left([\mathbf{c}_{u}^{t} \| \mathbf{S}_{u}^{t} \| \mathbf{e}_{\Delta_t}]\right)$, where $\mathbf{e}_{\Delta_t}$ encodes the prediction horizon. Prediction and target are compared after ℓ_2 normalization:

$$D(\widehat{\mathbf{y}}, \mathbf{y}) = \left\| \frac{\widehat{\mathbf{y}}}{\|\widehat{\mathbf{y}}\|_2} - \mathrm{sg}\left(\frac{\mathbf{y}}{\|\mathbf{y}\|_2}\right)\right\|_2^2. \tag{9}$$

The node loss is $\mathcal{L}^N = \frac{1}{|\mathcal{U}_t|}\sum_{u\in\mathcal{U}_t} D(\widehat{\mathbf{y}}_{u}^{t}, \mathbf{y}_{u}^{t})$, where $\mathcal{U}_t = \mathcal{V}_t$ for node prediction and contains the observed-interaction endpoints in the current batch for link prediction.

Relation-Level Predictive Objective. We use the ordered endpoint transformation $\chi(\mathbf{a}, \mathbf{b}) = [\mathbf{a} \| \mathbf{b} \| \mathbf{a} \odot \mathbf{b} \| \mathbf{a} - \mathbf{b}]$. The historical relation representation is

$$\mathbf{q}_{uv}^{t} = \phi_{\mathrm{fuse}}\left([g_\theta(\chi(\mathbf{c}_{u}^{t}, \mathbf{c}_{v}^{t})) \| \mathbf{g}_{uv}^{t} \| \mathbf{S}_{uv}^{t}]\right). \tag{10}$$

Its future latent is predicted as $\widehat{\mathbf{y}}_{uv}^{t} = p_R\left([\mathbf{q}_{uv}^{t} \| \mathbf{e}_{\Delta_t}]\right)$, with target $\mathbf{y}_{uv}^{t} = \mathrm{sg}\left(g_{\bar\theta}(\chi(\mathbf{y}_{u}^{t}, \mathbf{y}_{v}^{t}))\right)$. Explicit target-snapshot pair statistics, such as adjacency and common-neighbor counts, are not provided to $g_{\bar\theta}$. For observed relation pairs $\mathcal{R}_t^+$, $\mathcal{L}^R = \frac{1}{|\mathcal{R}_t^+|}\sum_{(u,v)\in\mathcal{R}_t^+} D(\widehat{\mathbf{y}}_{uv}^{t}, \mathbf{y}_{uv}^{t})$. Sampled negatives are used only by the downstream link objective. **Joint Objective.** We apply variance and covariance regularization to the predicted node and relation latents to reduce collapse. The resulting objective is

$$\mathcal{L}_{\mathrm{JEPA}} = \lambda_N\mathcal{L}^N + \lambda_R\mathcal{L}^R + \lambda_{\mathrm{var}}\left(\mathcal{L}_{\mathrm{var}}^N + \mathcal{L}_{\mathrm{var}}^R\right) + \lambda_{\mathrm{cov}}\left(\mathcal{L}_{\mathrm{cov}}^N + \mathcal{L}_{\mathrm{cov}}^R\right). \tag{11}$$

For node prediction, we additionally use in-batch node discrimination. For link prediction,

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{JEPA}} + \lambda_{\mathrm{link}}\mathcal{L}_{\mathrm{link}} + \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}},$$

where $\mathcal{L}_{\mathrm{link}}$ is the binary candidate loss and $\mathcal{L}_{\mathrm{rank}}$ ranks the observed destination against sampled candidates.

Strict Event-Prefix Encoder. The bin-level representation $\mathbf{q}_{uv}^{t}$ is fixed within target bin t and does not use events from that bin. To include information available before the exact query time, we

encode the strict prefix $\mathcal{H}_{<\tau}$. The event-history encoder uses statistics of the directed pair and its endpoints, together with historical link-feature profiles when available:

$$\mathbf{o}_{uv}^{<\tau} = \mathrm{EventHistory}(u, v; \mathcal{H}_{<\tau}). \tag{12}$$

All statistics are computed strictly before τ, so equal-timestamp events are excluded.

Training and Inference. The target snapshot $\widetilde{G}_t$ is used only to construct $\mathbf{y}_u^t$ and $\mathbf{y}_{uv}^t$. For query (u, v, τ), $\mathbf{z}_{uv}^\tau = \phi_\tau\left([\mathbf{q}_{uv}^t \| \mathbf{o}_{uv}^{<\tau}]\right), s_{uv}^\tau = h_\psi(\mathbf{z}_{uv}^\tau)$. Inference therefore uses only $\mathcal{C}_t$ and $\mathcal{H}_{<\tau}$; the target branch and predictive losses are used only during training.

Table 1: **Transductive** AUC-ROC for dynamic link prediction with random (*rnd*), historical (*hist*), and inductive (*ind*) negative edge sampling. **Bold red** and underlined blue emphasize best and 2nd-best results.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer	CLDG	MaskDGNN	DVGMAE	DyG-WM
rnd	Wikipedia	96.55± 0.19	94.93± 0.50	92.66± 0.43	98.46± 0.05	98.60± 0.01	97.51± 0.09	97.12± 0.06	**98.98± 0.01**	89.50± 0.18	86.71± 0.53	83.96± 0.55	98.94± 0.02
	Reddit	98.04± 0.04	97.69± 0.51	97.97± 0.02	97.75± 0.38	99.10± 0.01	97.91± 0.04	98.03± 0.02	99.09± 0.00	96.75± 0.01	96.73± 0.02	95.72± 0.07	**99.20± 0.01**
	MOOC	83.16± 2.19	79.33± 4.02	86.01± 0.06	**94.19± 0.73**	82.05± 0.28	85.58± 0.39	84.98± 0.13	89.80± 0.23	68.17± 1.30	74.97± 0.20	71.02± 0.89	82.50± 0.39
	LastFM	70.94± 0.72	71.92± 0.84	71.54± 0.35	62.38± 2.16	85.93± 0.11	71.10± 0.32	73.59± 0.11	**93.15± 0.02**	64.49± 0.69	65.48± 0.66	63.00± 0.21	92.61± 0.06
	Enron	87.29± 1.84	85.60± 1.98	74.41± 0.73	84.87± 1.60	90.49± 0.04	83.90± 0.28	84.56± 0.24	**93.46± 0.14**	87.87± 0.47	85.82± 0.86	74.84± 1.19	92.55± 0.25
	UCI	90.35± 0.50	64.90± 9.03	82.54± 0.41	94.05± 0.37	93.93± 0.04	89.71± 2.88	91.98± 0.43	**94.59± 0.22**	86.18± 0.30	88.04± 0.08	88.31± 0.17	94.15± 0.17
	Flights	91.92± 1.17	92.01± 1.02	94.50± 0.07	94.34± 0.73	98.44± 0.03	91.31± 0.04	91.21± 0.01	**98.95± 0.01**	94.66± 0.12	93.89± 0.06	91.81± 0.16	98.03± 0.15
	Can. Parl.	80.76± 1.34	71.50± 0.68	75.48± 1.40	80.12± 2.27	75.24± 2.82	74.99± 1.52	83.83± 0.39	**97.24± 0.33**	77.81± 0.96	82.96± 0.99	82.13± 0.78	82.88± 0.76
	US Legis.	**84.68± 0.17**	78.09± 2.08	74.89± 0.34	83.15± 0.54	77.54± 0.08	75.90± 0.51	76.15± 0.42	76.95± 0.32	68.83± 2.94	82.34± 0.12	82.24± 0.13	83.53± 0.14
	UN Trade	69.53± 0.49	66.55± 0.30	66.85± 0.02	69.47± 0.77	69.05± 0.05	64.87± 0.03	65.02± 1.30	62.34± 6.07	69.73± 0.10	77.28± 0.30	78.01± 0.20	**80.65± 0.68**
	UN Vote	69.66± 0.73	68.88± 0.67	53.63± 0.31	**71.99± 1.30**	53.41± 0.17	51.41± 0.27	52.39± 0.15	59.38± 0.35	50.00± 0.00	51.87± 0.98	53.98± 1.00	66.95± 0.26
	Contact	96.65± 0.58	96.44± 0.16	97.19± 0.03	96.68± 0.65	90.38± 0.12	95.52± 0.15	93.84± 0.06	**98.57± 0.01**	72.65± 0.21	75.08± 0.52	71.30± 0.45	98.21± 0.04
	Avg. Rank	5.33	7.83	7.50	5.25	5.50	8.42	7.00	3.08	8.92	7.75	8.75	**2.67**
hist	Wikipedia	81.62± 0.51	77.80± 0.54	69.91± 0.42	82.98± 0.24	67.18± 3.87	85.93± 1.29	**87.85± 0.42**	80.37± 1.55	73.86± 0.31	71.65± 0.84	70.38± 1.04	79.35± 0.80
	Reddit	79.94± 0.52	78.80± 1.64	77.71± 0.06	78.48± 1.05	80.35± 0.10	80.57± 0.23	78.35± 0.12	81.15± 0.27	76.99± 0.13	78.06± 0.07	75.77± 0.33	**81.82± 0.33**
	MOOC	81.34± 2.48	72.45± 4.05	81.97± 0.36	**90.48± 1.77**	72.78± 0.62	77.44± 0.95	84.68± 1.47	89.12± 0.88	60.42± 3.86	73.13± 1.49	70.77± 1.83	82.50± 1.82
	LastFM	75.93± 2.22	68.70± 3.12	62.99± 0.43	54.92± 9.88	67.69± 0.12	61.47± 3.00	64.21± 0.35	**78.54± 0.20**	61.08± 1.52	60.20± 1.50	61.88± 0.97	77.36± 0.73
	Enron	74.41± 1.61	76.64± 0.71	65.70± 0.37	74.75± 2.46	66.52± 0.14	72.16± 0.61	**77.50± 0.59**	77.47± 0.36	73.49± 1.32	69.98± 1.64	62.82± 3.52	74.99± 1.93
	UCI	**80.48± 2.67**	55.49± 8.21	61.00± 0.63	79.10± 1.36	57.84± 0.40	74.64± 7.05	78.21± 0.53	76.18± 0.55	72.87± 0.38	75.01± 0.69	77.04± 0.59	67.61± 0.82
	Flights	67.44± 0.80	63.01± 1.89	71.96± 0.18	68.73± 1.48	66.38± 0.89	70.40± 0.22	70.51± 0.13	67.40± 1.12	72.82± 0.08	**73.05± 0.24**	70.60± 0.48	62.34± 2.16
	Can. Parl.	77.22± 10.36	69.29± 1.29	70.46± 2.72	76.92± 4.29	70.64± 2.13	71.49± 1.87	81.58± 0.96	**97.75± 0.66**	84.90± 1.53	89.30± 1.49	87.71± 0.62	78.19± 1.97
	US Legis.	79.40± 4.47	84.25± 2.48	64.27± 3.26	78.80± 4.12	78.22± 4.57	78.27± 5.61	73.38± 8.17	77.50± 6.41	68.79± 5.21	88.85± 4.53	**92.47± 0.93**	87.56± 1.65
	UN Trade	63.84± 0.87	62.48± 0.44	63.62± 0.39	64.69± 1.41	66.19± 0.37	61.02± 0.39	64.01± 1.37	62.09± 5.46	71.35± 1.19	79.72± 0.97	79.72± 1.43	**81.23± 1.62**
	UN Vote	78.79± 0.92	75.89± 1.22	58.80± 1.20	75.91± 1.64	51.46± 0.68	51.29± 1.16	53.65± 1.44	67.62± 0.64	50.00± 0.00	55.86± 2.62	58.93± 1.11	**84.05± 2.79**
	Contact	96.42± 0.43	95.87± 0.22	95.76± 0.19	94.81± 1.47	83.73± 0.33	94.70± 0.27	93.25± 0.29	**97.15± 0.10**	64.50± 0.69	68.73± 0.46	64.65± 1.00	95.98± 0.15
	Avg. Rank	4.75	7.25	8.42	5.50	8.58	7.33	5.50	4.42	8.42	6.50	7.00	**4.33**
ind	Wikipedia	72.75± 0.63	70.45± 0.77	72.35± 0.45	81.45± 0.33	69.43± 4.24	74.14± 3.26	**84.76± 0.38**	78.68± 2.75	57.84± 0.47	56.98± 0.45	56.61± 0.44	77.62± 2.29
	Reddit	83.23± 0.29	82.23± 1.46	85.55± 0.07	82.19± 1.54	**88.32± 0.12**	73.02± 1.12	85.06± 0.08	85.68± 0.41	74.38± 0.51	78.23± 0.33	78.12± 0.20	87.60± 0.30
	MOOC	67.88± 1.79	59.54± 4.35	75.56± 0.70	**83.13± 3.07**	71.58± 0.64	74.63± 0.79	77.51± 1.11	82.55± 0.68	51.71± 2.49	56.38± 0.66	57.15± 0.62	74.66± 3.07
	LastFM	63.34± 1.75	61.19± 1.55	62.99± 0.41	53.18± 4.97	67.77± 0.15	55.40± 2.81	60.77± 0.29	68.50± 0.59	52.21± 0.79	52.23± 0.78	51.77± 0.31	**69.65± 2.74**
	Enron	72.78± 1.45	72.65± 1.36	64.91± 0.42	71.51± 3.26	76.72± 0.28	70.77± 0.70	72.83± 0.49	75.45± 0.71	66.21± 1.20	67.32± 1.92	58.12± 3.21	**79.55± 0.71**
	UCI	63.20± 1.39	52.22± 1.92	60.51± 0.43	64.22± 0.94	57.67± 0.43	70.11± 3.98	**74.34± 0.58**	65.51± 0.74	50.54± 0.35	54.48± 0.39	56.16± 0.08	57.52± 0.44
	Flights	68.28± 1.44	64.04± 2.14	72.87± 0.28	71.91± 1.68	70.12± 0.89	72.48± 0.20	72.36± 0.19	68.36± 1.58	73.15± 0.30	**75.33± 0.54**	72.68± 0.46	66.71± 1.01
	Can. Parl.	67.81± 12.02	63.83± 0.82	69.54± 1.78	74.82± 4.48	73.30± 1.08	72.13± 0.80	72.14± 0.98	**97.53± 0.44**	77.81± 0.75	83.69± 1.26	82.15± 0.38	76.29± 3.18
	US Legis.	81.63± 3.74	83.66± 2.33	62.35± 3.80	73.73± 4.72	74.50± 4.20	76.55± 8.69	71.15± 9.06	75.45± 5.69	71.18± 3.09	89.21± 1.90	**92.17± 0.19**	88.67± 1.06
	UN Trade	65.83± 0.96	64.43± 0.20	70.71± 0.31	66.95± 1.90	73.89± 0.25	67.14± 0.41	67.93± 1.41	62.85± 5.23	71.25± 0.64	79.40± 0.62	80.30± 0.58	**84.53± 0.74**
	UN Vote	75.70± 1.40	74.74± 1.21	55.59± 1.03	74.87± 1.56	54.20± 1.68	53.29± 1.28	51.44± 0.64	57.19± 0.29	50.00± 0.00	51.19± 2.24	57.90± 2.11	**84.54± 0.71**
	Contact	94.57± 0.73	94.65± 0.30	94.81± 0.22	92.87± 1.87	88.95± 0.28	92.69± 0.32	91.29± 0.30	95.30± 0.35	55.37± 0.55	59.52± 0.70	53.94± 0.57	**95.79± 0.40**
	Avg. Rank	6.42	7.83	6.50	5.92	6.08	7.08	5.92	4.42	9.17	7.42	7.67	**3.58**

4 Experiments

Datasets and Baselines. We evaluate on 12 dynamic graph datasets from the DyGLib benchmark (Yu et al., 2023): Wikipedia, Reddit, MOOC, LastFM, Enron, UCI, Can. Parl., US Legis., UN Trade, UN Vote, Contact and Flights, each split chronologically into 70%/15%/15% for training, validation and testing. Details are in Appendix A.1. We compare against eleven baselines in two families. The first consists of eight supervised temporal graph learning methods: three memory-based methods, JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019) and TGN (Rossi et al., 2020); a random-walk-based method, CAWN (Wang et al., 2021b); a GNN-based method, TGAT (Xu et al., 2020); an MLP-based method, GraphMixer (Cong et al., 2023); and two Transformer-based methods, TCL (Wang et al., 2021a) and DyGFormer (Yu et al., 2023). The second consists of three self-supervised dynamic graph representation learning methods: CLDG (Xu et al., 2023), DVGMAE (Gao et al., 2025) and MaskDGNN (He et al., 2025). For dynamic node classification, we additionally include two discrete-time DGNNs, EvolveGCN-H (Pareja et al., 2020) and ROLAND (You et al., 2022), and compare against SG-JEPA (Zhang et al., 2026b), the most closely related joint-embedding predictive architecture for dynamic graphs.

Implementation Details. For a fair comparison, we reproduce all baselines in a unified pipeline built on DyGLib (Yu et al., 2023), with the same event-bin discretization, the same chronological split and the same negative sampling protocol. We set the same history window for DyG-WM and the snapshot-based baselines to fairly compare their dynamic graph modeling ability. We train each model for up to 100 epochs with early stopping and select the best-performing checkpoint for testing.

Table 2: **Inductive** AUC-ROC for dynamic link prediction in the inductive (new-node) setting with random negative edge sampling. **Bold red** and underlined blue indicate the best and second-best results, respectively.

Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer	CLDG	MaskDGNN	DVGMAE	DyG-WM
Wikipedia	94.06± 0.23	91.46± 0.24	78.29± 0.67	97.22± 0.07	96.93± 0.02	96.74± 0.09	95.89± 0.06	97.42± 0.02	86.63± 0.42	84.35± 0.20	80.43± 1.29	**97.52± 0.03**
Reddit	96.01± 0.10	95.25± 0.29	67.96± 1.62	96.12± 0.61	96.48± 0.03	95.95± 0.09	95.36± 0.11	96.56± 0.02	94.33± 0.08	94.66± 0.06	93.26± 0.21	**97.44± 0.10**
MOOC	81.86± 0.78	79.41± 2.36	80.43± 0.30	**91.44± 0.49**	82.43± 0.28	82.33± 0.65	83.09± 0.18	87.92± 0.29	63.33± 0.58	72.03± 0.39	66.51± 0.95	79.63± 0.44
LastFM	81.89± 0.69	81.36± 1.62	76.83± 0.14	62.44± 5.05	87.88± 0.05	76.36± 1.04	80.10± 0.16	93.05± 0.05	67.08± 1.27	66.47± 1.02	63.90± 0.57	**93.50± 0.22**
Enron	82.81± 1.85	74.13± 2.31	69.84± 0.47	74.17± 2.79	87.60± 0.25	81.23± 0.33	77.88± 0.15	**90.73± 0.10**	80.10± 1.15	76.05± 1.65	62.54± 0.60	87.45± 0.26
UCI	77.47± 1.78	58.26± 6.40	77.93± 0.49	86.21± 2.79	89.65± 0.16	79.89± 0.98	88.84± 0.15	92.03± 0.09	75.80± 0.37	76.76± 0.18	77.16± 0.11	**92.32± 0.13**
Flights	90.02± 1.59	86.79± 1.09	87.50± 0.11	87.28± 0.79	97.20± 0.03	84.18± 0.05	83.85± 0.12	**98.06± 0.02**	90.27± 0.28	89.06± 0.08	84.55± 0.44	93.98± 0.46
Can. Parl.	54.34± 2.88	56.87± 1.73	59.57± 1.46	61.61± 1.38	63.27± 0.54	61.04± 0.27	60.06± 0.70	**96.51± 0.55**	61.78± 0.68	60.35± 1.78	63.45± 0.58	62.63± 1.08
US Legis.	66.20± 0.44	60.56± 0.98	47.87± 1.00	63.91± 0.42	54.53± 1.45	51.35± 0.55	51.97± 1.39	54.20± 0.41	59.01± 4.64	65.25± 0.28	65.29± 0.30	**66.75± 1.29**
UN Trade	61.11± 0.34	58.65± 1.10	56.94± 1.45	61.87± 0.65	67.06± 0.20	63.59± 0.06	62.96± 0.85	53.59± 3.14	67.45± 0.21	74.92± 0.37	**76.31± 0.31**	67.42± 1.82
UN Vote	58.19± 1.79	56.75± 0.91	51.31± 0.51	**61.08± 0.50**	49.23± 0.63	48.95± 0.33	50.77± 0.25	56.37± 0.25	49.86± 0.27	48.42± 0.46	49.75± 0.75	54.84± 1.17
Contact	95.87± 0.65	92.36± 0.38	96.40± 0.05	94.39± 0.85	89.51± 0.27	94.73± 0.13	92.63± 0.05	**98.30± 0.01**	69.24± 0.89	70.23± 0.54	65.79± 0.61	96.75± 0.11
Avg. Rank	5.67	8.25	8.75	5.67	4.67	7.17	7.08	3.25	7.92	8.25	8.50	**2.83**

We repeat each experiment 5 times with different random seeds and report the mean and standard deviation. Details in Appendix A.2.

Evaluation Details. We evaluate dynamic link prediction in both transductive and inductive settings. The transductive setting considers future links between nodes observed during training, while the inductive setting evaluates links involving unseen nodes. Following DyGLib (Yu et al., 2023), we use random (*rnd*), historical (*hist*), and inductive (*ind*) negative sampling, and report Average Precision (AP) and AUC-ROC. For dynamic node classification, we use labelled-node training ratios of 40%, 60%, and 80%, and report Macro-F1 and Micro-F1 on the held-out test nodes. More details are given in Appendix A.3.

Table 3: **Node classification on dynamic graphs.** Macro-F1 and Micro-F1 (%) on the held-out test nodes, reported as mean ± std over 5 seeds at labelled-node training ratios of 40%, 60%, and 80%. **Bold red** and underlined blue indicate the best and second-best results, respectively.

Metric	Dataset	Ratio	TGAT	CAWN	TCL	GraphMixer	DyGFormer	EvolveGCN-H	ROLAND	CLDG	MaskDGNN	DVGMAE	SG-JEPA	DyG-WM
Macro-F1	DBLP	40%	68.54± 0.60	64.71± 0.92	69.76± 1.09	71.69± 0.79	67.88± 0.72	26.10± 11.19	47.14± 20.87	60.70± 1.34	70.98± 0.84	60.36± 0.83	73.96± 0.59	**75.45± 0.35**
		60%	71.60± 0.61	66.32± 0.60	72.33± 0.68	73.80± 0.52	69.99± 0.62	32.22± 19.39	58.32± 13.49	59.70± 1.95	71.91± 0.24	60.36± 0.81	75.88± 0.36	**77.13± 0.51**
		80%	73.12± 0.56	67.66± 1.11	73.59± 0.77	75.41± 0.77	71.46± 0.80	37.47± 9.69	52.91± 12.28	61.77± 2.20	73.88± 1.21	60.38± 1.00	77.55± 0.33	**78.13± 0.23**
	Tmall	40%	47.36± 0.78	45.49± 0.35	49.12± 0.49	49.20± 0.94	33.31± 6.17	35.80± 9.81	41.46± 8.38	32.06± 1.83	48.35± 1.60	36.23± 1.02	54.16± 0.36	**57.37± 0.37**
		60%	49.05± 0.26	47.28± 0.56	50.10± 0.58	51.50± 0.75	32.65± 2.84	36.13± 6.82	40.87± 6.29	30.10± 1.44	50.83± 1.03	35.47± 1.19	55.81± 0.34	**59.17± 0.49**
		80%	49.86± 0.44	48.23± 0.82	50.56± 0.78	52.92± 0.41	34.73± 2.49	39.41± 3.33	40.74± 6.82	31.00± 1.70	51.39± 1.52	35.40± 1.30	57.24± 0.39	**60.52± 0.39**
Micro-F1	DBLP	40%	69.59± 0.48	66.24± 0.41	70.35± 1.55	72.15± 0.67	68.31± 0.69	43.97± 7.17	59.19± 14.41	62.97± 0.85	71.01± 0.67	62.41± 0.58	74.20± 0.55	**75.79± 0.24**
		60%	71.72± 0.50	67.94± 0.56	72.84± 0.80	74.15± 0.23	70.25± 0.31	45.96± 14.30	66.06± 4.61	62.40± 1.45	72.07± 0.32	62.42± 0.58	76.17± 0.18	**77.52± 0.39**
		80%	73.43± 0.50	68.67± 0.83	73.78± 0.88	75.60± 0.26	70.99± 0.97	51.36± 6.31	64.18± 4.32	63.64± 1.33	73.41± 1.00	62.46± 0.91	77.57± 0.21	**78.42± 0.23**
	Tmall	40%	53.28± 0.80	50.82± 0.80	54.56± 0.50	54.31± 0.66	45.61± 2.26	50.60± 7.74	54.41± 4.10	45.23± 0.66	55.43± 1.39	47.94± 0.64	58.18± 0.76	**61.88± 0.30**
		60%	54.88± 0.41	53.03± 0.49	56.19± 0.75	56.69± 1.17	45.25± 2.39	52.33± 4.71	54.91± 3.32	44.34± 0.64	57.51± 0.48	47.89± 0.52	59.93± 0.83	**63.72± 0.43**
		80%	55.97± 0.44	54.28± 0.61	57.05± 0.57	57.90± 0.57	47.22± 0.81	54.13± 1.68	54.68± 3.83	45.15± 0.80	57.68± 1.45	47.43± 1.35	61.71± 0.53	**65.08± 0.37**

4.1 Main Results

Overall Performance. Tables 1 and 2 show that DyG-WM achieves the best average rank under all three transductive negative-sampling strategies, with ranks of 2.67, 4.33, and 3.58, and also ranks first in the new-node setting with 2.83. The advantage is more apparent under inductive negatives, where DyG-WM performs best on several datasets, including UN Trade and UN Vote. For dynamic node classification, Table 3 shows that DyG-WM gives the best Macro-F1 and Micro-F1 on both DBLP and Tmall across all three training ratios. Overall, the results show that DyG-WM performs well on both link prediction and node classification. More details in Appendix A.4.

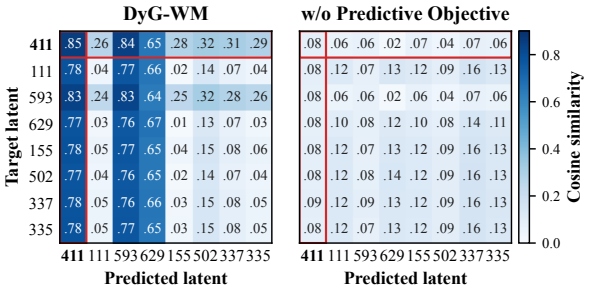


Figure 3: Latent alignment on Can. Parl. with and without the predictive objective.

4.2 Ablation Study

Component Contributions. Table 4 shows that removing any component lowers AUC-ROC on all three datasets. The largest drop comes from removing the path-signature encoder, from 74.99 to 59.09 on Enron, 78.19 to 60.31 on Can. Parl., and 77.36 to 70.17 on LastFM. Historical context and the predictive objective also provide consistent gains.

Table 4: Results (AUC-ROC) of ablations.

Settings	Enron	Can. Parl.	LastFM
w/o Historical Context	71.53± 3.02	76.08± 2.77	75.11± 0.52
w/o Path Signature	59.09± 5.52	60.31± 4.65	70.17± 0.87
w/o Predictive Objective	71.69± 3.94	75.58± 2.11	75.93± 1.32
DyG-WM	**74.99± 1.93**	**78.19± 1.97**	**77.36± 0.73**

Predictive Alignment. Figure 3 shows that the JEPA objective substantially improves alignment between predicted and target latents. With the full model, many cosine similarities are around 0.64-0.85, while removing the objective reduces most values to near zero. This matches the performance drop observed in the ablation table. More details in Appendix A.4.

4.3 Efficiency

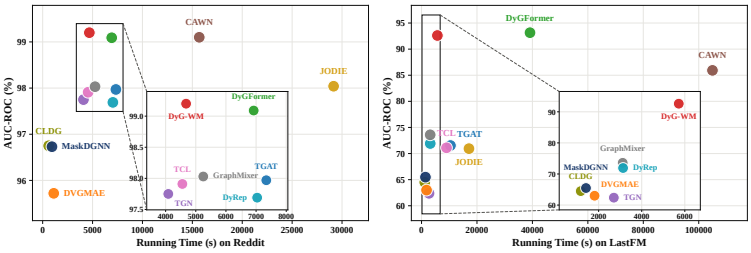


Figure 4: AUC-ROC-efficiency trade-off on Reddit and LastFM.

As shown in Figure 4, Reddit and LastFM exhibit different accuracy–efficiency trade-offs. On Reddit, several strong baselines already approach 99 AUC-ROC, so the main difference lies in the computation needed to reach this accuracy range. DyG-WM attains the highest AUC-ROC while training faster than other high-accuracy models such as DyGFormer and CAWN. On LastFM, the trade-off is more pronounced: DyGFormer remains slightly more accurate, but requires substantially more training time, whereas faster baselines incur large accuracy losses. DyG-WM stays close to the best accuracy while using a moderate training budget. Across both datasets, no compared method is simultaneously faster and more accurate than DyG-WM, placing it on the accuracy-efficiency frontier rather than optimizing either accuracy or speed alone. More details are in Appendix A.4.

4.4 Robustness

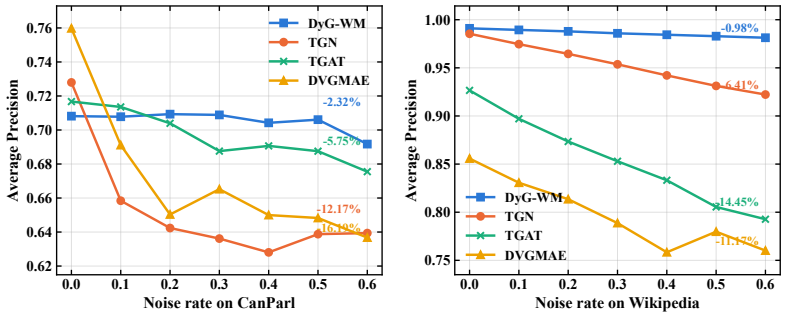


Figure 5: Robustness to noise on Can. Parl. and Wikipedia.

We train all models on clean data in the transductive setting with random negative sampling and add feature noise only at test time. For a noise rate σ, we randomly replace a fraction σ of the input features with resampled values, with σ ranging from 0.1 to 0.6. Figure 5 shows that DyG-WM is less affected by increasing noise than the compared methods. On Wikipedia, its AP drops by only 0.98% at $\sigma = 0.6$, compared with 6.41%, 14.45%, and 11.17% for TGN, TGAT, and DVGMAE. On Can. Parl., DyG-WM becomes the best-performing method from $\sigma = 0.2$ onward and drops by only 2.32% at $\sigma = 0.6$.

5 Conclusion

We introduced DyG-WM, a dynamic graph world modelling framework for jointly modelling the evolution of nodes and relations. DyG-WM uses historical node-relation contexts and their ordered structural changes to predict future representations through a dual-level joint-embedding objective, without reconstructing the original graph observations. Experiments on 12 dynamic link prediction datasets and two dynamic node classification datasets show that the model performs well across both tasks. The ablation and latent-alignment results also show the contributions of the main components. Future work will consider more complex evolving relational systems and a broader range of downstream tasks.

AI USE STATEMENT

Generative AI tools were used solely for language polishing, grammar checking, and improving the clarity and readability of the manuscript. They were not used to develop the proposed methodology, design or conduct experiments, generate experimental results, or analyze and interpret the results. All AI-assisted revisions were carefully reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Jiaru Bai, Sebastian Mosbach, Connor J Taylor, Dogancan Karan, Kok Foong Lee, Simon D Rihm, Jethro Akroyd, Alexei A Lapkin, and Markus Kraft. A dynamic knowledge graph approach to distributed self-driving laboratories. *Nature Communications*, 15(1):462, 2024.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Dong Chen, Shuai Zheng, Muhao Xu, Zhenfeng Zhu, and Yao Zhao. Signn: A spike-induced graph neural network for dynamic graph representation learning. *Pattern Recognition*, 158:111026, 2025.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. In *Signature Methods in Finance: An Introduction with Computational Applications*, pp. 3–64. Springer Nature Switzerland Cham, 2025.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? *arXiv preprint arXiv:2302.11636*, 2023.

Tao Feng, Yexin Wu, Guanyu Lin, and Jiaxuan You. Graph world model. *arXiv preprint arXiv:2507.10539*, 2025a.

ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2025b.

Mengzhou Gao, Xinxun Zhang, Pengfei Jiao, Tianpeng Li, and Zhidong Zhao. Dvgmae: Self-supervised dynamic variational graph masked autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17901–17913, 2025.

Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. Long range propagation on continuous-time dynamic graphs. *arXiv preprint arXiv:2406.02740*, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Yiming He, Xiang Li, Zhongying Zhao, Haobing Liu, Peilan He, and Yanwei Yu. Maskdgnn: Self-supervised dynamic graph neural networks with activeness-aware temporal masking. In *IJCAI*, pp. 2892–2900, 2025.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *Proc. of ICLR*, 2020.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Dongyuan Li, Shiyin Tan, Ying Zhang, Ming Jin, Shirui Pan, Manabu Okumura, and Renhe Jiang. Dyg-mamba: Continuous state space modeling on dynamic graphs. *Advances in Neural Information Processing Systems*, 38:129101–129130, 2026.

Minne Li, Mengyue Yang, Furui Liu, Xu Chen, et al. Causal world models by unsupervised deconfounding of physical dynamics. *arXiv preprint arXiv:2012.14228*, 2020a.

Yunzhu Li, Toru Lin, Kexin Yi, et al. Visual grounding of learned physical models. In *Proc. of ICML*, pp. 5927–5936. PMLR, 2020b.

Zhixuan Lin, Yi-Fu Wu, et al. Improving generative imagination in object-centric world models. In *Proc. of ICML*, pp. 6140–6149. PMLR, 2020.

Jiawei Liu, Senqiao Yang, Mingjun Wang, Yu Wang, and Bei Yu. Graph world models: Concepts, taxonomy, and future directions. *arXiv preprint arXiv:2604.27895*, 2026.

Yuanfu Lu, Xiao Wang, Chuan Shi, Philip S Yu, and Yanfang Ye. Temporal network embedding with micro-and macro-dynamics. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 469–478, 2019.

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

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35: 32928–32941, 2022.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 519–527, 2020.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9:79143–79168, 2021.

Zixing Song, Yuji Zhang, and Irwin King. Towards fair financial services for all: A temporal gnn approach for individual fairness on transaction networks. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 2331–2341, 2023.

Yuxing Tian, Yiyan Qi, and Fan Guo. Freedyg: Frequency enhanced continuous-time dynamic graph model for link prediction. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=82Mc5ilInM.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

Lu Wang, Xiaofu Chang, Shuang Li, Yunfei Chu, Hui Li, Wei Zhang, Xiaofeng He, Le Song, Jingren Zhou, and Hongxia Yang. Tcl: Transformer-based dynamic graph modelling via contrastive learning. *arXiv preprint arXiv:2105.07944*, 2021a.

Maolin Wang, Xinjian Zhao, Wanyu Wang, Sheng Zhang, Jiansheng Li, Bowen Yu, Binhao Wang, Shucheng Zhou, Dawei Yin, Qing Li, et al. Embedding in recommender systems: A survey. *ACM Transactions on Information Systems*, 44(5):1–55, 2026.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. *arXiv preprint arXiv:2101.05974*, 2021b.

Wenchao Weng, Jin Fan, Huifeng Wu, Yujie Hu, Hao Tian, Fu Zhu, and Jia Wu. A decomposition dynamic graph convolutional recurrent network for traffic forecasting. *Pattern Recognition*, 142: 109670, 2023.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

Yiming Xu, Bin Shi, Teng Ma, Bo Dong, Haoyi Zhou, and Qinghua Zheng. Cldg: Contrastive learning on dynamic graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 696–707. IEEE, 2023.

Jiaxuan You, Tianyu Du, and Jure Leskovec. Roland: Graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 2358–2366, 2022.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

Qi Yuan, Yang Liu, Yateng Tang, Xinhuan Chen, Xuehao Zheng, Qing He, and Xiang Ao. Dynamic graph learning with static relations for credit risk assessment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 13133–13141, 2025.

Dong Zhang, Lin Li, Ming Li, Amran Bhuiyan, Meng Sun, Xiaohui Tao, and Jimmy Huang. Modeling item-level dynamic variability with residual diffusion for bundle recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16253–16261, 2026a.

Huizhe Zhang, Yuchang Zhu, Huazhen Zhong, Liang Chen, and Zibin Zheng. Scalable and efficient joint spiking embedding predictive architecture for large-scale dynamic graphs. *arXiv preprint arXiv:2607.18412*, 2026b.

Lunjun Zhang, Ge Yang, and Bradly C Stadie. World model as a graph: Learning latent landmarks for planning. In *International conference on machine learning*, pp. 12611–12620. PMLR, 2021.

Jialun Zheng, Jie Liu, Jiannong Cao, Xiao Wang, Hanchen Yang, and Yankai Chen. Dp-dgad: A generalist dynamic graph anomaly detector with dynamic prototypes. In *Proceedings of the ACM Web Conference 2026*, pp. 857–868, 2026.

Deyao Zhu, Li Erran Li, and Mohamed Elhoseiny. Value memory graph: A graph-structured world model for offline reinforcement learning. *arXiv preprint arXiv:2206.04384*, 2022.

Yiqing Zou, Hanning Yuan, Qianyu Yang, Ziqiang Yuan, Shuliang Wang, and Sijie Ruan. Meta dynamic graph for traffic flow prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16584–16592, 2026.

APPENDIX

This is the appendix of the work: "**Dynamic Graph World Modelling: Learning Future Dynamics from Temporal Context**". In this appendix, we provide additional methodological details, experimental settings, extended results, and further analyses of the proposed DYG-WM.

A EXPERIMENTAL DETAILS

A.1 DATASETS

We evaluate link prediction on the twelve public dynamic graph datasets collected by Poursafaei et al. (2022) and redistributed with DyGLib (Yu et al., 2023); their statistics are summarized in Table 5. Following SG-JEPA (Zhang et al., 2026b), we use the two large-scale datasets released by Lu et al. (2019), summarized in Table 6.

Table 5: Statistics of the datasets. N/A denotes the absence of node or edge features. #Nodes denotes the number of nodes.

Datasets	Domains	#Nodes	#Links	#N&L Feature	Bipartite	Duration	Unique Steps	Time Granularity
Wikipedia	Social	9,227	157,474	N/A & 172	True	1 month	152,757	Unix timestamps
Reddit	Social	10,984	672,447	N/A & 172	True	1 month	669,065	Unix timestamps
MOOC	Interaction	7,144	411,749	N/A & 4	True	17 months	345,600	Unix timestamps
LastFM	Interaction	1,980	1,293,103	N/A & N/A	True	1 month	1,283,614	Unix timestamps
Enron	Social	184	125,235	N/A & N/A	False	3 years	22,632	Unix timestamps
UCI	Social	1,899	59,835	N/A & N/A	False	196 days	58,911	Unix timestamps
Can. Parl.	Politics	734	74,478	N/A & 1	False	14 years	14	years
US Legis.	Politics	225	60,396	N/A & 1	False	12 congresses	12	congresses
UN Trade	Economics	255	507,497	N/A & 1	False	32 years	32	years
UN Vote	Politics	201	1,035,742	N/A & 1	False	72 years	72	years
Contact	Proximity	692	2,426,279	N/A & 1	False	1 month	8,064	5 minutes
Flights	Transport	13,169	1,927,145	N/A & 1	False	4 months	122	days

Table 6: Statistics of the dynamic node classification datasets.

Datasets	#Nodes	#Links	#Classes	#Snapshots	#Features
DBLP	28,085	236,894	10	27	80
Tmall	577,314	4,807,545	5	186	80

A.2 IMPLEMENTATION DETAILS

Environment and Reproducibility. We implement all methods in Python using PyTorch. The code requires Python 3.10 or later and PyTorch 2.1 or later. All experiments are conducted on NVIDIA A100 GPUs.

Shared Architecture Configuration. Unless specified otherwise, both DYG-WM and the snapshot-based baselines use a hidden dimension of 128 and two message-passing layers. For dynamic link prediction, we set $W = 4$, corresponding to three historical snapshots in $\mathcal{C}_t$, with a 32-dimensional temporal encoding and a 256-dimensional predictor. For dynamic node classification, we set $W = 3$, corresponding to two historical snapshots, with a 32-dimensional temporal encoding and a 256-dimensional predictor.

For DYG-WM, we use second-order path signatures throughout. In link prediction, the event-history representation and semantic projection have dimension 16, the relation-context budget is $B = 32$, the temporal decay factor is $\lambda = 0.8$, and the common-neighbor bonus is $\beta = 1.0$. The target encoder uses $\eta_{\text{ema}} = 0.999$. The default loss weights are $\lambda_N = 0.02, \lambda_R = 0.02, \lambda_{\text{link}} = 1.0, \lambda_{\text{rank}} = 1.0, \lambda_{\text{var}} = 0.005, \lambda_{\text{cov}} = 0.0005$. For node classification, the relation and link objectives are disabled, and we use $\lambda_N = 1.0, \lambda_{\text{var}} = 0.02, \lambda_{\text{cov}} = 0.001$, with $\eta_{\text{ema}} = 0.995$ for the node target encoder.

Dynamic Link Prediction. We follow the unified DyGLib evaluation pipeline (Yu et al., 2023). Each dataset is split chronologically into 70%, 15%, and 15% for training, validation, and testing.

Table 7: Default implementation settings for DyG-WM. Dataset-specific overrides are selected using validation data only.

Configuration	Link prediction	Node classification
Hidden dimension	128	128
Message-passing layers	2	2
Window parameter W	4	3
Temporal dimension	32	32
Predictor hidden dimension	256	256
Signature order	2	2
Relation-context budget B	32	N/A
EMA coefficient η_{ema}	0.999	0.995
Maximum epochs	100	100
Learning rate	3×10^{-4}	10^{-3}
Weight decay	10^{-5}	10^{-5}
Gradient-norm threshold	1.0	1.0

Under the default random-negative protocol, each observed interaction is paired with one uniformly corrupted destination during evaluation. Candidate collisions produced by the benchmark sampler are retained rather than deduplicated. Evaluation batches contain 200 positive interactions and their corresponding negatives. We use fixed negative-sampling seeds of 0 and 2 for validation and testing, respectively, and present the same sampled candidates to all methods.

We optimize DyG-WM with AdamW using $\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-10}$. The default mini-batch contains 200 positive candidate groups and their one-to-one negatives, giving 400 candidate rows. We reduce the learning rate by a factor of 0.5 after five validation evaluations without improvement, down to a minimum of 10^{-5}. Models are evaluated after every epoch, and training stops after 20 evaluations without checkpoint improvement. Checkpoints are selected using validation AP and AUC only, and the test set is evaluated once using the selected checkpoint.

Dynamic Node Classification. We train the node representation learner for at most 100 epochs using AdamW with $\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-10}$, and gradient clipping at 1.0. Node mini-batches contain 1,024 nodes for DBLP and Tmall. After representation learning, all encoders are frozen and an MLP probe with a hidden dimension of 512 is fitted to the final-snapshot representations. We use 40%, 60%, 80% of the labelled nodes for probe training and reserve 10% of this training portion for validation. Checkpoints are selected according to validation Macro-F1 after a minimum of 25 representation-learning epochs.

For DyG-WM, separate MLP probes are fitted to the predicted node representation and the online-encoder representation, and their logits are averaged for final classification. The target encoder is used only to construct representation targets during training and is not used by the downstream classifier.

A.3 Detailed Evaluation Settings

Detailed Settings for Effectiveness Evaluation. To comprehensively evaluate the performance of all methods, we adopt three negative edge sampling strategies for dynamic link prediction following DyGLib (Yu et al., 2023). Let $\mathcal{E}_{\text{train}}$ and $\mathcal{E}_{\text{test}}$ denote the training and test edge sets, respectively. The edges can be divided into three categories: (a) edges appearing only during training, $\mathcal{E}_{\text{train}} \setminus \mathcal{E}_{\text{test}}$; (b) edges appearing in both training and testing, $\mathcal{E}_{\text{train}} \cap \mathcal{E}_{\text{test}}$; and (c) edges appearing only during testing, $\mathcal{E}_{\text{test}} \setminus \mathcal{E}_{\text{train}}$. We emphasize that inductive negative sampling is different from the new-node inductive setting. The three negative sampling strategies are described as follows:

- **Random Negative Sampling (*rnd*).** For each positive interaction, we retain its source node and timestamp while uniformly sampling a destination node from the complete destination set. The resulting source-destination pair is treated as a negative edge.
- **Historical Negative Sampling (*hist*).** Historical negative sampling selects edges that appeared at earlier timestamps but are absent from the current evaluation batch. This setting evaluates whether a model can identify the specific time at which a previously observed

Table 8: **Transductive** Average Precision (AP) for dynamic link prediction with random (*rnd*), historical (*hist*), and inductive (*ind*) negative edge sampling. **bold red** and underlined blue emphasize best and 2nd-best results.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer	CLDG	MaskDGNN	DVGMAE	DyG-WM
rnd	Wikipedia	96.69± 0.21	95.20± 0.54	92.68± 0.47	98.54± 0.05	98.78± 0.01	97.93± 0.06	97.39± 0.05	99.06± 0.02	91.55± 0.09	88.76± 0.43	85.68± 0.62	**99.10± 0.02**
	Reddit	98.07± 0.02	97.71± 0.56	98.03± 0.02	97.74± 0.39	99.18± 0.01	98.09± 0.04	98.09± 0.01	99.18± 0.00	97.01± 0.02	96.97± 0.02	95.91± 0.07	**99.28± 0.01**
	MOOC	79.55± 2.76	75.48± 4.64	84.54± 0.05	**92.81± 0.85**	81.50± 0.22	85.17± 0.49	83.79± 0.16	89.26± 0.26	64.90± 1.88	72.35± 0.17	68.41± 1.18	81.94± 0.38
	LastFM	71.67± 0.90	72.38± 0.75	73.39± 0.27	62.64± 2.26	86.99± 0.04	76.82± 0.38	75.81± 0.09	**93.12± 0.05**	63.91± 0.63	65.76± 0.57	63.67± 0.19	92.20± 0.04
	Enron	84.06± 2.82	83.50± 2.64	76.08± 0.56	81.68± 2.21	89.42± 0.05	85.53± 0.30	81.62± 0.29	**92.27± 0.09**	86.13± 0.35	84.57± 0.68	71.82± 0.58	91.40± 0.22
	UCI	89.16± 0.88	60.57± 8.46	82.60± 0.44	94.19± 0.38	95.17± 0.04	91.31± 3.18	93.51± 0.28	**95.88± 0.14**	86.68± 0.14	87.71± 0.06	88.05± 0.10	95.26± 0.09
	Flights	90.89± 1.28	90.77± 0.97	94.44± 0.06	94.06± 0.59	98.51± 0.02	91.40± 0.04	91.11± 0.02	**98.94± 0.01**	95.00± 0.12	94.19± 0.06	92.04± 0.15	98.24± 0.10
	Can. Parl.	72.25± 1.93	64.92± 0.94	70.45± 1.24	74.33± 1.48	69.91± 2.04	70.13± 0.96	77.46± 0.52	**97.18± 0.31**	70.14± 0.97	74.10± 1.50	76.06± 0.63	73.86± 0.68
	US Legis.	**77.97± 0.28**	70.93± 2.77	66.87± 0.53	75.92± 0.88	70.97± 0.16	69.24± 0.73	69.62± 0.69	70.40± 0.47	67.09± 1.68	74.97± 0.12	74.59± 0.28	76.81± 0.29
	UN Trade	64.76± 0.24	63.34± 0.32	64.16± 0.06	65.43± 0.62	65.77± 0.05	62.44± 0.02	63.51± 1.04	68.30± 2.28	65.85± 0.08	72.70± 0.29	73.46± 0.25	**76.75± 0.62**
	UN Vote	65.45± 0.66	64.84± 0.49	52.89± 0.27	**67.60± 1.04**	53.06± 0.08	51.57± 0.23	52.08± 0.13	57.33± 0.24	50.86± 0.00	51.20± 0.81	53.03± 0.84	62.21± 0.17
	Contact	95.27± 0.84	96.02± 0.17	96.62± 0.04	95.60± 0.97	90.66± 0.13	94.67± 0.19	91.81± 0.08	**98.34± 0.01**	70.93± 0.38	74.53± 0.43	69.61± 0.32	98.06± 0.04
	Avg. Rank	6.50	8.50	7.75	5.42	5.17	7.17	6.92	**2.42**	9.00	7.92	8.58	2.67
hist	Wikipedia	84.19± 0.52	79.75± 0.61	71.19± 0.87	86.98± 0.27	68.24± 7.35	86.77± 3.35	**90.99± 0.33**	84.55± 2.14	77.21± 0.14	75.87± 0.66	74.48± 0.69	84.06± 0.93
	Reddit	79.23± 0.63	77.96± 2.11	77.20± 0.12	77.92± 1.17	80.85± 0.25	80.81± 0.54	78.77± 0.24	81.12± 0.90	76.12± 0.26	78.35± 0.21	75.93± 0.38	**83.84± 0.62**
	MOOC	76.75± 4.57	67.32± 4.48	82.40± 0.61	**89.38± 2.13**	74.63± 0.85	80.86± 0.99	83.33± 1.67	88.27± 1.02	59.34± 2.39	71.43± 1.40	69.74± 2.05	81.52± 2.35
	LastFM	76.15± 2.39	71.65± 2.46	70.70± 0.34	59.74± 6.41	69.68± 0.11	71.57± 1.41	72.08± 0.39	81.55± 0.33	59.78± 0.86	61.99± 0.76	61.39± 1.17	**81.95± 0.61**
	Enron	67.04± 2.26	73.25± 1.48	64.87± 0.37	70.50± 2.50	65.56± 0.18	71.66± 0.81	**78.25± 0.29**	75.92± 0.49	68.89± 1.24	65.65± 1.40	61.39± 2.23	76.82± 1.87
	UCI	79.81± 2.74	52.95± 6.43	69.36± 0.43	81.47± 1.33	65.50± 0.45	81.54± 5.46	**84.70± 0.45**	81.39± 0.43	72.67± 0.34	74.60± 0.91	76.21± 0.54	74.38± 0.84
	Flights	63.22± 0.77	59.23± 2.13	71.87± 0.23	67.80± 2.30	64.82± 0.96	70.89± 0.23	71.76± 0.22	64.93± 1.39	72.90± 0.35	**73.48± 0.40**	70.91± 0.78	60.98± 2.16
	Can. Parl.	67.77± 12.02	62.85± 1.28	66.12± 2.02	72.39± 3.46	66.24± 2.27	67.23± 1.65	77.43± 1.52	**97.46± 0.63**	77.24± 1.44	82.21± 2.40	82.26± 0.71	73.28± 1.94
	US Legis.	64.99± 4.62	76.62± 3.77	52.92± 1.97	66.33± 5.57	68.96± 8.28	69.98± 7.32	62.89± 10.19	66.57± 8.52	73.81± 4.42	83.93± 6.04	**87.88± 2.05**	84.55± 2.56
	UN Trade	55.13± 0.64	57.06± 0.86	57.55± 0.42	58.89± 1.04	58.32± 0.36	55.64± 0.40	59.34± 3.41	65.91± 4.09	61.87± 1.19	72.20± 1.41	72.02± 1.91	**73.20± 2.07**
	UN Vote	72.85± 0.94	70.41± 0.89	55.99± 0.87	70.57± 1.12	51.45± 0.34	51.43± 1.04	52.07± 1.16	61.66± 0.66	50.57± 0.00	53.57± 3.00	56.37± 0.93	**76.68± 4.03**
	Contact	95.21± 1.09	96.22± 0.29	96.53± 0.24	93.81± 2.71	84.67± 0.39	95.33± 0.23	93.55± 0.28	**97.52± 0.18**	63.62± 0.77	69.80± 0.30	64.20± 0.82	97.16± 0.11
	Avg. Rank	6.75	7.75	8.17	6.08	8.75	6.17	4.83	**3.67**	8.25	6.42	7.33	3.83
ind	Wikipedia	77.53± 0.77	73.50± 1.05	74.27± 0.84	85.92± 0.56	70.11± 8.39	74.74± 6.48	**88.87± 0.32**	83.09± 4.38	64.95± 0.54	63.25± 0.79	62.64± 0.94	82.28± 3.02
	Reddit	86.44± 0.25	85.14± 2.15	87.77± 0.11	84.88± 1.65	**91.98± 0.13**	79.43± 1.21	87.55± 0.07	91.14± 0.38	75.45± 0.97	80.31± 0.62	80.97± 0.23	91.92± 0.30
	MOOC	67.62± 2.01	58.75± 3.92	77.44± 1.01	82.50± 3.35	74.29± 0.65	77.91± 0.90	77.34± 1.25	**83.00± 0.66**	53.96± 2.63	56.99± 0.87	57.56± 0.29	77.10± 3.38
	LastFM	65.37± 2.09	63.86± 1.77	70.46± 0.31	57.04± 2.80	67.21± 0.18	63.12± 1.54	68.11± 0.14	73.21± 1.20	52.23± 0.48	52.83± 0.51	51.97± 0.12	**75.73± 2.12**
	Enron	70.43± 2.10	70.56± 2.41	65.76± 0.32	70.24± 3.19	76.63± 0.27	71.52± 0.39	74.79± 0.27	77.77± 0.99	66.57± 1.18	66.84± 1.38	60.25± 2.38	**83.28± 0.94**
	UCI	64.41± 1.65	52.57± 1.43	68.44± 0.34	70.27± 0.77	64.61± 0.63	76.11± 2.59	**80.05± 0.39**	72.41± 0.55	52.39± 0.50	55.07± 0.31	55.84± 0.16	65.38± 1.11
	Flights	65.43± 1.59	60.92± 2.63	74.35± 0.50	72.31± 2.95	69.33± 0.91	75.07± 0.23	75.14± 0.26	68.16± 1.79	75.85± 0.44	**78.06± 0.96**	74.99± 0.72	66.84± 0.95
	Can. Parl.	62.31± 11.75	59.72± 0.67	65.90± 1.24	70.18± 4.04	68.44± 1.10	68.13± 1.14	71.20± 1.10	**96.96± 0.39**	74.72± 1.21	79.03± 0.86	80.30± 0.30	71.60± 2.66
	US Legis.	67.60± 4.48	75.69± 3.52	51.80± 2.26	61.28± 5.31	64.51± 6.78	69.09± 10.16	61.38± 10.93	63.99± 7.41	73.48± 2.09	83.60± 2.11	**87.49± 0.11**	85.47± 0.93
	UN Trade	58.21± 0.90	58.41± 0.62	63.50± 0.45	60.83± 1.84	65.28± 0.48	60.72± 0.49	62.56± 3.46	66.78± 5.38	62.42± 0.81	72.69± 1.17	72.91± 0.72	**78.71± 0.99**
	UN Vote	70.10± 1.17	69.01± 1.08	54.47± 0.68	69.11± 1.16	53.97± 1.11	51.70± 1.37	50.98± 0.39	53.46± 0.22	50.57± 0.00	51.39± 1.62	55.86± 1.43	**78.42± 0.85**
	Contact	93.46± 1.18	94.69± 0.45	95.37± 0.28	91.31± 3.33	90.62± 0.37	92.76± 0.33	91.33± 0.28	95.13± 0.60	56.90± 0.83	61.76± 1.33	55.20± 0.73	**96.99± 0.31**
	Avg. Rank	7.33	7.92	6.33	6.50	6.42	6.58	5.33	4.00	9.08	7.58	7.58	**3.33**

edge reappears, rather than predicting that the edge remains active after its first occurrence. For an evaluation batch spanning timestamps $[t_s, t_e]$, negatives are sampled from edges observed strictly before t_s after excluding edges appearing within $[t_s, t_e]$. If the historical candidate pool contains fewer edges than the number of positive examples, the remaining negatives are generated using random sampling.

- **Inductive Negative Sampling (*ind*).** In contrast to historical sampling, inductive negative sampling focuses on edges that first appear during the evaluation period. After such an edge has appeared, the model is required to determine whether it will reoccur at subsequent timestamps. For each evaluation batch, we sample edges that were not observed before the current evaluation split, have appeared before the current batch, and are absent from the current batch. If the number of eligible inductive edges is insufficient, the remaining negatives are generated using random sampling.

Detailed Settings for Robustness Evaluation. As shown in Figure 5, we evaluate model robustness to feature corruption on Wikipedia and Can. Parl. All models are trained on clean data under the transductive setting with random negative sampling, while noise is introduced only during evaluation. For a noise rate σ, we randomly select a fraction σ of the input features and replace them with resampled values. We vary σ from 0.1 to 0.6, with $\sigma = 0$ denoting the clean setting, and use the same corruption seed for all compared methods at each noise level.

A.4 Additional Experimental Results

This subsection provides additional results that complement the main experiments. Tables 8 and 9 report Average Precision (AP) for dynamic link prediction under transductive and new-node inductive settings. We also provide additional latent-alignment visualizations on Wikipedia and UCI, together with the full accuracy-efficiency comparison across all 12 link prediction datasets in Figure 8.

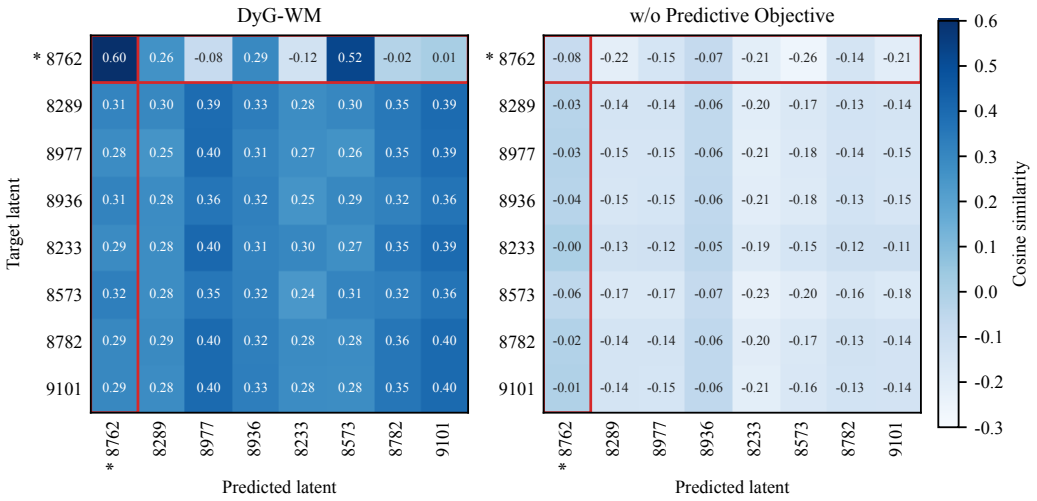


Figure 6: Latent alignment with and without the predictive objective in the Wikipedia dataset.

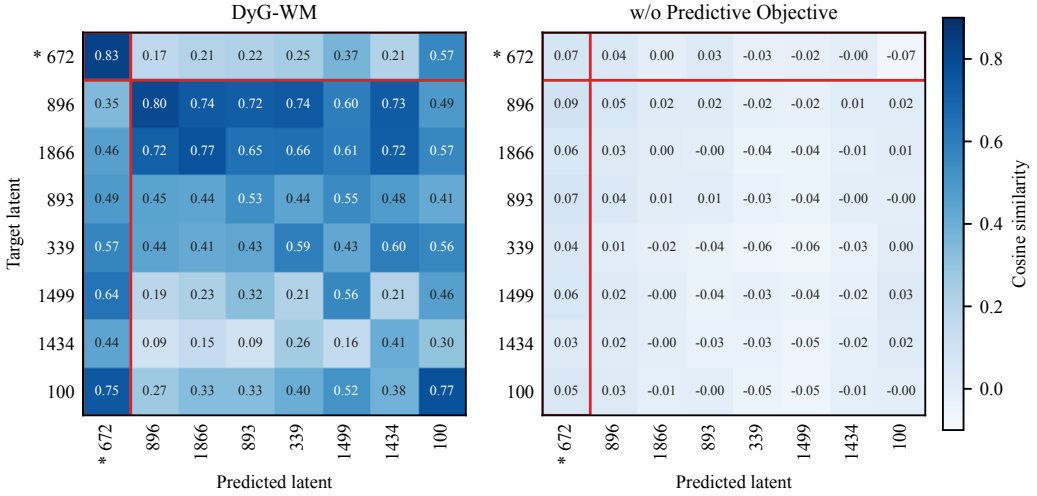
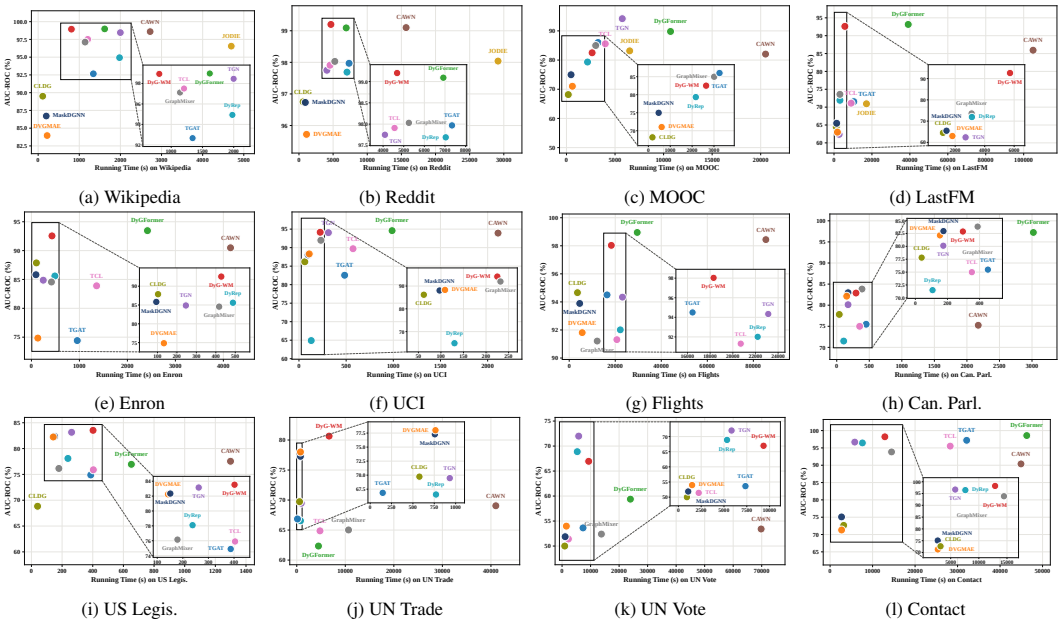


Figure 7: Latent alignment with and without the predictive objective in the UCI dataset.

Table 9: **Inductive** Average Precision (AP) for dynamic link prediction in the inductive (new-node) setting with random negative edge sampling. **Bold red** and underlined blue indicate the best and second-best results, respectively.

Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer	CLDG	MaskDGNN	DVGMAE	DyG-WM
Wikipedia	94.75± 0.20	92.25± 0.22	80.33± 0.69	97.45± 0.06	97.22± 0.03	97.26± 0.08	96.33± 0.05	97.56± 0.03	89.39± 0.28	86.59± 0.20	82.19± 1.04	**97.94± 0.02**
Reddit	95.87± 0.13	95.19± 0.23	66.20± 1.64	96.24± 0.66	97.66± 0.03	96.45± 0.06	95.61± 0.08	97.74± 0.02	94.68± 0.06	95.02± 0.04	93.54± 0.18	**97.90± 0.06**
MOOC	78.67± 1.16	75.82± 3.14	79.46± 0.43	**89.71± 0.64**	81.86± 0.21	81.88± 1.08	81.71± 0.20	87.24± 0.30	61.43± 0.51	69.48± 0.39	65.55± 0.93	79.74± 0.48
LastFM	82.84± 0.84	82.01± 1.45	78.54± 0.16	62.93± 3.73	89.44± 0.04	80.93± 0.78	81.98± 0.11	92.22± 0.09	66.60± 1.18	67.66± 0.63	65.99± 0.41	**92.61± 0.17**
Enron	80.62± 2.01	70.73± 1.73	71.10± 0.35	72.08± 2.32	86.31± 0.21	82.54± 0.40	76.70± 0.16	**89.89± 0.13**	76.55± 1.20	74.24± 1.44	59.32± 0.79	86.75± 0.56
UCI	78.46± 3.21	58.76± 6.47	79.51± 0.30	87.78± 2.28	92.10± 0.09	82.22± 1.31	90.91± 0.13	92.99± 0.06	76.78± 0.32	77.92± 0.24	77.93± 0.15	**93.38± 0.18**
Flights	90.27± 1.51	86.50± 1.20	87.76± 0.12	86.35± 0.98	97.32± 0.02	84.89± 0.04	84.37± 0.19	**98.02± 0.01**	90.94± 0.24	89.79± 0.09	85.67± 0.39	94.14± 0.64
Can. Parl.	54.06± 2.10	55.36± 1.35	57.28± 0.95	58.74± 1.04	59.72± 0.20	58.41± 0.30	57.82± 0.58	**95.58± 0.45**	61.62± 0.70	56.60± 1.22	60.20± 0.83	61.77± 0.63
US Legis.	61.14± 0.31	57.16± 0.76	49.86± 0.87	59.52± 0.49	55.37± 0.98	52.15± 0.77	53.46± 1.42	55.03± 0.11	56.90± 3.39	60.28± 0.47	60.47± 0.41	**61.66± 1.99**
UN Trade	58.60± 0.49	56.85± 1.42	55.40± 1.77	60.02± 0.86	65.12± 0.20	62.09± 0.10	63.49± 0.89	70.05± 3.61	65.72± 0.25	72.18± 0.13	**73.33± 0.27**	62.74± 1.48
UN Vote	57.33± 0.92	57.13± 0.83	51.28± 0.46	**59.57± 0.37**	50.30± 0.32	50.64± 0.88	51.08± 0.34	55.22± 0.34	50.98± 0.12	49.18± 0.53	50.62± 0.48	55.14± 1.17
Contact	95.11± 1.04	92.58± 0.38	95.61± 0.09	92.97± 1.26	89.88± 0.18	93.87± 0.19	90.36± 0.05	96.03± 0.01	68.19± 0.95	70.30± 0.66	65.54± 0.46	**96.89± 0.05**
Avg. Rank	6.08	8.25	8.75	5.92	5.08	6.58	7.08	**2.58**	8.00	8.33	8.75	**2.58**



(a) Wikipedia (b) Reddit (c) MOOC (d) LastFM

(e) Enron (f) UCI (g) Flights (h) Can. Parl.

(i) US Legis. (j) UN Trade (k) UN Vote (l) Contact

Figure 8: Accuracy-efficiency trade-off across all dynamic link prediction datasets. Each panel reports AUC-ROC against training time.

A.5 Algorithm Details

This section summarizes DyG-WM's representation construction, training, and inference. Algorithm 1 constructs representations, Algorithm 2 gives dual-level predictive training, and Algorithm 3 gives link inference without the target branch.

A.6 Historical Representation Construction

For target bin t, the online branch uses $\mathcal{C}_t = (\widetilde{G}_{t-W+1}, \ldots, \widetilde{G}_{t-1})$. Each query (u, v, τ) additionally uses the strict event prefix $\mathcal{H}_{<\tau}$. Bin-level representations depend only on $\mathcal{C}_t$, while exact-time representations also depend on $\mathcal{H}_{<\tau}$.

The quantities $\mathbf{c}_x^t$, $\mathbf{g}_{uv}^t$, $\mathbf{S}_x^t$, $\mathbf{S}_{uv}^t$, and $\mathbf{q}_{uv}^t$ depend only on $\mathcal{C}_t$ and are fixed within bin t, whereas $\mathbf{o}_{uv}^{<\tau}$ and $\mathbf{z}_{uv}^{\tau}$ vary with τ. Events at timestamp τ are excluded from $\mathcal{H}_{<\tau}$.

A.7 Dual-Level Predictive Training

Let Θ contain all trainable online parameters, while $\bar{\theta}$ denotes the EMA target-encoder parameters. Candidate groups remain intact within minibatches so each observed interaction is trained with all sampled negatives.

Algorithm 1 Historical node-relation representation construction

Require: Historical snapshots $\mathcal{C}_t$; queries $\mathfrak{P} = \{(u_j, v_j, \tau_j)\}_{j=1}^{|\mathfrak{P}|}$; strict prefixes $\{\mathcal{H}_{<\tau_j}\}$; online parameters Θ; context budget B

Ensure: Historical and exact-time relation representations

for $s \in \mathcal{I}_t$ **do**
  $\mathbf{H}^s \leftarrow \mathrm{GNN}_\theta(\widetilde{G}_s)$ and $\mathbf{h}_x^s \leftarrow \mathbf{H}^s[x]$ for $x \in \mathcal{V}$.
end for
for all $x \in \mathcal{V}$ **do**
  Encode $\{\mathbf{h}_x^s\}_{s\in\mathcal{I}_t}$ and neighborhood trajectories as $\mathbf{c}_x^t$.
  Construct $\{\mathbf{d}_{x,s}^N\}_{s\in\mathcal{I}_t}$ and compute $\mathbf{S}_x^t \leftarrow \phi_N\left(\mathrm{Sig}_2((\mathbf{d}_{x,s}^N)_{s\in\mathcal{I}_t})\right)$.
end for
for $j = 1, \ldots, |\mathfrak{P}|$ **do**
  $(u, v, \tau) \leftarrow (u_j, v_j, \tau_j)$.
  Construct $\mathcal{P}_{uv}^t$ and select $\mathcal{K}_{uv}^t \leftarrow \mathrm{TopB}_{w\in\mathcal{P}_{uv}^t}\, \rho_{uv}^t(w)$.
  for $s \in \mathcal{I}_t$ **do**
    Pool selected embeddings into $\mathbf{p}_{uv}^s$.
  end for
  $\mathbf{g}_{uv}^t \leftarrow \mathrm{GRU}_{\mathrm{rel}}((\mathbf{p}_{uv}^s)_{s\in\mathcal{I}_t})$.
  Construct $\{\mathbf{d}_{uv,s}^R\}_{s\in\mathcal{I}_t}$ and compute $\mathbf{S}_{uv}^t \leftarrow \phi_R\left(\mathrm{Sig}_2((\mathbf{d}_{uv,s}^R)_{s\in\mathcal{I}_t})\right)$.
  $\mathbf{r}_{uv}^t \leftarrow g_\theta\left(\chi(\mathbf{c}_u^t, \mathbf{c}_v^t)\right)$.
  $\mathbf{q}_{uv}^t \leftarrow \phi_{\mathrm{fuse}}\left([\mathbf{r}_{uv}^t \| \mathbf{g}_{uv}^t \| \mathbf{S}_{uv}^t]\right)$.
  $\mathbf{o}_{uv}^{<\tau} \leftarrow \mathrm{EventHistory}(u, v; \mathcal{H}_{<\tau})$.
  $\mathbf{z}_{uv}^\tau \leftarrow \phi_\tau\left([\mathbf{q}_{uv}^t \| \mathbf{o}_{uv}^{<\tau}]\right)$.
end for
return $\{\mathbf{c}_x^t, \mathbf{S}_x^t\}_{x\in\mathcal{V}}$ and $\{\mathbf{g}_{u_jv_j}^t, \mathbf{S}_{u_jv_j}^t, \mathbf{q}_{u_jv_j}^t, \mathbf{o}_{u_jv_j}^{<\tau_j}, \mathbf{z}_{u_jv_j}^{\tau_j}\}_{j=1}^{|\mathfrak{P}|}$

Node and relation predictive targets use only observed interactions; sampled negatives enter only $\mathcal{L}_{\mathrm{link}}$ and $\mathcal{L}_{\mathrm{rank}}$. Relation targets use target endpoint latents without explicit target-snapshot pair statistics.

For the node benchmark, the relation predictor, event-history branch, link scorer, and relation loss are disabled. The node predictor maps $[\mathbf{c}_x^t \| \mathbf{S}_x^t \| \mathbf{e}_{\Delta_t}]$ to $\widehat{\mathbf{y}}_x^t$ and is trained against the target latent $\mathbf{y}_x^t$ together with node discrimination and anti-collapse terms. Node and link prediction are trained separately.

A.8 Link Inference

At inference, target encoders and predictive losses are removed. For query (u, v, τ) in bin t, only $\mathcal{C}_t$ and $\mathcal{H}_{<\tau}$ are used.

Thus, $s_{uv}^\tau = \mathrm{GNN}_\theta(u, v, \tau; \mathcal{C}_t, \mathcal{H}_{<\tau})$ and neither $\widetilde{G}_t$ nor the target latents $\mathbf{y}_u^t, \mathbf{y}_{uv}^t$ enter inference.

B Computational Complexity

We analyze one target bin t with $\mathcal{I}_t = \{t - W + 1, \ldots, t - 1\}$ and $L = |\mathcal{I}_t| = W - 1$. Let $m_s = \|A_s\|_0, M_{\mathcal{C}} = \sum_{s\in\mathcal{I}_t} m_s$, where m_s and $M_{\mathcal{C}}$ are the numbers of stored message-passing edges in snapshot s and the historical context, respectively. Let $N = |\mathcal{V}|$, d be the hidden dimension, d_s the projected path dimension, B the relation-context budget, and $P = |\mathfrak{P}|$ the number of candidate rows in a group-preserving minibatch. With G observed interactions and K negatives per interaction, $P = G(K + 1)$.

Historical snapshot encodings and node contexts are shared across candidates, while relation contexts, relation signatures, and event-history features are pair-conditioned.

Algorithm 2 Dual-level joint-embedding predictive training

Require: Training windows $\{(\mathcal{C}_t, \widetilde{G}_t)\}$; event stream $\mathcal{E}$; negative count K; online parameters Θ; loss weights $\lambda_N, \lambda_R, \lambda_{\text{var}}, \lambda_{\text{cov}}, \lambda_{\text{link}}, \lambda_{\text{rank}}$; EMA coefficient η_{ema}
Ensure: Trained online parameters Θ

Initialize $\bar{\theta} \leftarrow \theta$ and construct the event-history index.
for each training epoch **do**
  for each target bin t **do**
    Construct candidate groups $\{\mathcal{Q}(u,\tau)\}$ and partition complete groups into minibatches.
    for each group-preserving minibatch $\mathfrak{P}$ **do**
      Apply Algorithm 1.
      $\mathbf{y}_x^t \leftarrow \text{sg}\left(\text{GNN}_{\bar{\theta}}(\widetilde{G}_t)[x]\right)$ for $x \in \mathcal{V}_t$.
      Let $\mathcal{R}_{t,\mathfrak{P}}^+$ be the observed relation pairs in $\mathfrak{P}$ and $\mathcal{U}_t$ their unique endpoints.
      for all $x \in \mathcal{U}_t$ **do**
        $\widehat{\mathbf{y}}_x^t \leftarrow p_N\left([\mathbf{c}_x^t \| \mathbf{S}_x^t \| \mathbf{e}_{\Delta_t}]\right)$.
      end for
      for all $(u,v) \in \mathcal{R}_{t,\mathfrak{P}}^+$ **do**
        $\widehat{\mathbf{y}}_{uv}^t \leftarrow p_R\left([\mathbf{q}_{uv}^t \| \mathbf{e}_{\Delta_t}]\right)$.
        $\mathbf{y}_{uv}^t \leftarrow \text{sg}\left(g_{\bar{\theta}}\left(\chi(\mathbf{y}_u^t, \mathbf{y}_v^t)\right)\right)$.
      end for
      for all $(u_j, v_j, \tau_j) \in \mathfrak{P}$ **do**
        $s_{u_j v_j}^{\tau_j} \leftarrow h_\psi(\mathbf{z}_{u_j v_j}^{\tau_j})$.
      end for
      $\mathcal{L}^N = \frac{1}{|\mathcal{U}_t|} \sum_{x \in \mathcal{U}_t} D(\widehat{\mathbf{y}}_x^t, \mathbf{y}_x^t)$.
      $\mathcal{L}^R = \frac{1}{|\mathcal{R}_{t,\mathfrak{P}}^+|} \sum_{(u,v) \in \mathcal{R}_{t,\mathfrak{P}}^+} D(\widehat{\mathbf{y}}_{uv}^t, \mathbf{y}_{uv}^t)$.
      Compute node- and relation-level variance and covariance penalties.
      Compute $\mathcal{L}_{\text{link}}$ over all candidate scores and $\mathcal{L}_{\text{rank}}$ within each candidate group.

$$\begin{aligned}\mathcal{L}_{\text{JEPA}} = \lambda_N \mathcal{L}^N + \lambda_R \mathcal{L}^R \\ + \lambda_{\text{var}}\left(\mathcal{L}_{\text{var}}^N + \mathcal{L}_{\text{var}}^R\right) + \lambda_{\text{cov}}\left(\mathcal{L}_{\text{cov}}^N + \mathcal{L}_{\text{cov}}^R\right).\end{aligned} \tag{13}$$

      $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{JEPA}} + \lambda_{\text{link}}\mathcal{L}_{\text{link}} + \lambda_{\text{rank}}\mathcal{L}_{\text{rank}}$.
      Update Θ, then $\bar{\theta} \leftarrow \eta_{\text{ema}}\bar{\theta} + (1 - \eta_{\text{ema}})\theta$.
    end for
  end for
end for
return Θ

B.1 Historical Snapshot and Node Encoding

For a fixed-depth sparse message-passing implementation of GNN_θ, encoding all historical snapshots costs

$$\mathcal{T}_{\text{snap}} = \mathcal{O}\left(M_{\mathcal{C}} d + LNd^2\right). \tag{14}$$

No dense $N \times N$ adjacency matrix is materialized.

Sparse neighborhood pooling costs $\mathcal{O}(M_{\mathcal{C}} d)$, node-wise recurrent updates cost $\mathcal{O}(LNd^2)$, and node-context fusion costs $\mathcal{O}(Nd^2)$. Hence,

$$\mathcal{T}_{\text{node}} = \mathcal{O}\left(M_{\mathcal{C}} d + LNd^2\right). \tag{15}$$

Both terms are shared across all P candidate rows.

B.2 Relation-Centric Context Selection

For candidate pair (u_j, v_j), define

$$D_j = \sum_{s \in \mathcal{I}_t} \left(|\mathcal{N}_s(u_j)| + |\mathcal{N}_s(v_j)|\right). \tag{16}$$

Algorithm 3 Dynamic link inference

Require: Historical snapshots $\mathcal{C}_t$; candidate queries $\mathfrak{P} = \{(u_j, v_j, \tau_j)\}$; strict event-history index; trained online parameters Θ

Ensure: Interaction scores $\{s_{u_j v_j}^{\tau_j}\}_{j=1}^{|\mathfrak{P}|}$

1: Encode only $\mathcal{C}_t$; do not invoke $\mathrm{GNN}_{\bar{\theta}}$, $g_{\bar{\theta}}$, or encode $\widetilde{G}_t$.
2: **for** $j = 1, \ldots, |\mathfrak{P}|$ **do**
3: $(u, v, \tau) \leftarrow (u_j, v_j, \tau_j)$.
4: Construct $\mathbf{c}_u^t, \mathbf{c}_v^t, \mathbf{g}_{uv}^t, \mathbf{S}_{uv}^t, \mathbf{q}_{uv}^t$ from $\mathcal{C}_t$.
5: Query the event-history index for events satisfying $\tau_i < \tau$ to obtain $\mathbf{o}_{uv}^{<\tau}$.
6: $\mathbf{z}_{uv}^{\tau} \leftarrow \phi_\tau\left([\mathbf{q}_{uv}^t \| \mathbf{o}_{uv}^{<\tau}]\right)$.
7: $s_{uv}^{\tau} \leftarrow h_\psi(\mathbf{z}_{uv}^{\tau})$.
8: **end for**
9: **return** $\{s_{u_j v_j}^{\tau_j}\}_{j=1}^{|\mathfrak{P}|}$

The number of distinct candidate context nodes is

$$C_j = \left| \{u_j, v_j\} \cup \bigcup_{s \in \mathcal{I}_t} \left(\mathcal{N}_s(u_j) \cup \mathcal{N}_s(v_j)\right) \right|. \tag{17}$$

Constructing the candidate union and scores costs $\mathcal{O}(D_j)$, while sorting costs $\mathcal{O}(C_j \log C_j)$. Thus,

$$\mathcal{T}_{\text{select}} = \mathcal{O}\left(D_{\mathfrak{P}} + \sum_{j=1}^{P} C_j \log C_j \right), \qquad D_{\mathfrak{P}} = \sum_{j=1}^{P} D_j. \tag{18}$$

Partial top-B selection can reduce the sorting term to $\mathcal{O}(C_j \log B)$.

Pooling at most B embeddings over L snapshots costs $\mathcal{O}(PLBd)$, and the relation GRU costs $\mathcal{O}(PLd^2)$. Therefore,

$$\mathcal{T}_{\text{relctx}} = \mathcal{O}\left(D_{\mathfrak{P}} + \sum_{j=1}^{P} C_j \log C_j + PLBd + PLd^2 \right). \tag{19}$$

B.3 Node-Relation Path Signatures

For L increments in $\mathbb{R}^{d_s}$, first-order accumulation costs $\mathcal{O}(Ld_s)$ and the second-order signature costs $\mathcal{O}(Ld_s^2)$ time and $\mathcal{O}(d_s^2)$ working memory.

For all N nodes,

$$\mathcal{T}_{\text{sig}}^{N} = \mathcal{O}\left(M_{\mathcal{C}} + NLd_s^2 + Nd_s^2 d\right). \tag{20}$$

The terms correspond to structural-transition extraction, second-order signature updates, and projection to dimension d.

For P candidate relations,

$$\mathcal{T}_{\text{sig}}^{R} = \mathcal{O}\left(D_{\mathfrak{P}} + PLd_s^2 + Pd_s^2 d\right). \tag{21}$$

Node signatures are shared, whereas relation signatures are pair-conditioned.

B.4 Exact-Time Event-History Queries

The event stream contains M raw interactions. Timestamp, directed-pair, and endpoint indices are constructed once in $\mathcal{O}(M \log M)$ time. Let U denote the number of distinct directed pairs. Without semantic profiles, the persistent index requires

$$\mathcal{O}\left(M + T(U + N)\right) \tag{22}$$

memory. Enabling d_m-dimensional semantic profiles adds $\mathcal{O}(T(U+N)d_m)$.

For query (u, v, τ), binary search retrieves the required strict-prefix statistics in $\mathcal{O}(\log M)$ time. Across one minibatch,

$$\mathcal{T}_{\text{hist}} = \mathcal{O}\left(P \log M + Pd^2\right). \tag{23}$$

The second term covers event-history projection and fusion.

B.5 Prediction Heads and Training Objective

The endpoint transformation, relation encoder, fusion modules, predictors, and link scorer cost $\mathcal{O}(Pd^2)$. Pointwise losses cost $\mathcal{O}(Pd)$ and groupwise ranking at most $\mathcal{O}(P \log P)$.

Let $P_+ = |\mathcal{R}^+_{t,\mathfrak{P}}|, U_+ = |\mathcal{U}_t|$, where P_+ is the number of observed relation pairs and U_+ their unique endpoints. Since $U_+ \leq 2P_+$, the node- and relation-level variance terms cost $\mathcal{O}(P_+ d)$, while the covariance terms cost

$$\mathcal{T}_{\text{cov}} = \mathcal{O}\left((U_+ + P_+)d^2\right) = \mathcal{O}(P_+ d^2). \tag{24}$$

The covariance term is omitted at a level with fewer than two prediction vectors.

B.6 Training-Time Overhead

Training additionally encodes the target snapshot $\widetilde{G}_t$. Let $m_t = \|A_t\|_0$. Target snapshot encoding and positive relation-target construction cost

$$\mathcal{T}_{\text{target}} = \mathcal{O}\left(m_t d + Nd^2 + P_+ d^2\right). \tag{25}$$

Including covariance regularization and the EMA update,

$$\mathcal{T}_{\text{train}} = \mathcal{O}\left(\mathcal{T}_{\text{infer}} + m_t d + Nd^2 + P_+ d^2 + |\Theta_{\text{EMA}}|\right). \tag{26}$$

This excludes the constant-factor backward pass through the online branch. Since $U_+ \leq 2P_+$, the $P_+ d^2$ term covers positive relation-target encoding and both covariance computations. The EMA update is independent of graph size and candidate count.

Each group-preserving minibatch performs one parameter update, so historical and target snapshots are re-encoded after each update. At inference, historical representations for a target bin can be reused across candidate minibatches.

B.7 Memory Complexity

Excluding model parameters and the persistent event-history index, inference requires

$$\mathcal{S}_{\text{infer}} = \mathcal{O}\left(M_{\mathcal{C}} + LNd + PB + PLd + (N+P)d_s^2\right). \tag{27}$$

Training additionally stores target embeddings, online activations, covariance matrices, gradients, and optimizer states:

$$\mathcal{S}_{\text{train}} = \mathcal{O}\left(\mathcal{S}_{\text{infer}} + Nd + P_+ d + d^2 + |\Theta|\right). \tag{28}$$

Relation-centric modeling is bounded by B, while the second-order signature remains linear in L. The target branch is training-only, so inference uses only $\mathcal{C}_t$ and $\mathcal{H}_{<\tau}$.

C Proofs and Technical Derivations

We establish three properties of DyG-WM: de-duplicated snapshots do not generally identify pairwise interaction multiplicity, second-order signatures capture temporal order beyond first-order summaries, and the deployed scorer depends only on the historical snapshot context and strict event prefix.

C.1 Non-Identifiability of Pairwise Interaction Multiplicity

For a directed pair (u, v), define its multiplicity in bin t as

$$n^t_{uv} = \left|\{(u_i, v_i, \tau_i, \mathbf{e}_i) \in \mathcal{E}_t : u_i = u,\ v_i = v\}\right|. \tag{29}$$

For the following proposition, assume that X_t contains fixed node attributes and nodewise event aggregates, such as incident interaction counts, nodewise link-feature aggregates, and last-activity times, but no pair-resolved statistic such as n^t_{uv}.

Proposition 1 (Snapshot non-identifiability).

Under this snapshot construction, the mapping from the raw event stream to the snapshot sequence is not injective. For every integer $k \geq 1$, there exist two streams $\mathcal{S}_k$ and $\mathcal{S}'_k$ with identical snapshot views $\widetilde{G}_s = (\mathcal{V}, A_s, X_s, \mathbf{m}_s)$ for every bin s, but for some directed pair, $|n^t_{uv} - n'^t_{uv}| = k$.

Proof. Consider four distinct nodes a, b, c, d, with a, b as sources and c, d as destinations. The two streams are identical outside bin t. Within bin t, all interactions share timestamp τ_0 and feature $\mathbf{e}_0$, with multiplicities

$\mathcal{S}_k$	c	d
a	$k+1$	1
b	1	$k+1$

$\mathcal{S}'_k$	c	d
a	1	$k+1$
b	$k+1$	1

. (30)

Both streams contain the same directed edge support $\{(a, c), (a, d), (b, c), (b, d)\}$, so de-duplication and symmetrization give identical A_t. Each source has $k+2$ outgoing interactions and each destination has $k+2$ incoming interactions in both streams. Since all interactions share $\mathbf{e}_0$ and τ_0, nodewise link-feature aggregates and last-activity statistics are also identical. The active-node masks coincide, and the streams are identical outside bin t. Hence, $\widetilde{G}_s(\mathcal{S}_k) = \widetilde{G}_s(\mathcal{S}'_k)$.

For pair (a, c), $n^t_{ac} = k+1, n'^t_{ac} = 1$, so $|n^t_{ac} - n'^t_{ac}| = k$. Since k is arbitrary, the snapshot sequence does not determine pairwise interaction multiplicity.

Consequence. For any deterministic encoder F whose graph input is only $\mathcal{C}_t$,

$$F\big(\mathcal{C}_t(\mathcal{S}_k)\big) = F\big(\mathcal{C}_t(\mathcal{S}'_k)\big), \tag{31}$$

although the underlying pairwise recurrence patterns differ. We therefore retain $\mathcal{H}_{<\tau}$ separately to preserve direction and repeated interactions. The proposition does not apply to snapshot representations that explicitly store pairwise multiplicities.

C.2 Order Sensitivity of the Second-Order Signature

For projected increments $\mathbf{d}_1, \ldots, \mathbf{d}_L \in \mathbb{R}^{d_s}$, define

$$\mathrm{Sig}^{(1)} = \sum_{i=1}^{L} \mathbf{d}_i, \tag{32}$$

and

$$\mathrm{Sig}^{(2)} = \frac{1}{2} \sum_{i=1}^{L} \mathbf{d}_i \otimes \mathbf{d}_i + \sum_{i<j} \mathbf{d}_i \otimes \mathbf{d}_j. \tag{33}$$

Proposition 2 (Two-step order sensitivity).

For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{d_s}$, the paths $(\mathbf{a}, \mathbf{b})$ and $(\mathbf{b}, \mathbf{a})$ have identical first-order signatures, while

$$\mathrm{Sig}^{(2)}(\mathbf{a}, \mathbf{b}) - \mathrm{Sig}^{(2)}(\mathbf{b}, \mathbf{a}) = \mathbf{a} \otimes \mathbf{b} - \mathbf{b} \otimes \mathbf{a}. \tag{34}$$

If both increments are nonzero and ϑ is their angle,

$$\Big\| \mathrm{Sig}^{(2)}(\mathbf{a}, \mathbf{b}) - \mathrm{Sig}^{(2)}(\mathbf{b}, \mathbf{a}) \Big\|_F = \sqrt{2}\, \|\mathbf{a}\|_2 \|\mathbf{b}\|_2 |\sin \vartheta|. \tag{35}$$

Thus, the second-order signature distinguishes the two orderings exactly when the increments are not collinear.

Proof. The first-order signatures satisfy $\mathrm{Sig}^{(1)}(\mathbf{a}, \mathbf{b}) = \mathbf{a} + \mathbf{b} = \mathrm{Sig}^{(1)}(\mathbf{b}, \mathbf{a})$. From Eq. 33,

$$\begin{aligned} \mathrm{Sig}^{(2)}(\mathbf{a}, \mathbf{b}) &= \frac{1}{2}\mathbf{a} \otimes \mathbf{a} + \mathbf{a} \otimes \mathbf{b} + \frac{1}{2}\mathbf{b} \otimes \mathbf{b}, \\ \mathrm{Sig}^{(2)}(\mathbf{b}, \mathbf{a}) &= \frac{1}{2}\mathbf{b} \otimes \mathbf{b} + \mathbf{b} \otimes \mathbf{a} + \frac{1}{2}\mathbf{a} \otimes \mathbf{a}. \end{aligned} \tag{36}$$

Subtracting gives Eq. 34. Let $\mathbf{M} = \mathbf{a}\mathbf{b}^\top - \mathbf{b}\mathbf{a}^\top$. Then

$$\begin{aligned} \|\mathbf{M}\|_F^2 &= 2\|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2 - 2(\mathbf{a}^\top \mathbf{b})^2 \\ &= 2\|\mathbf{a}\|_2^2 \|\mathbf{b}\|_2^2 \sin^2 \vartheta, \end{aligned} \tag{37}$$

which gives Eq. 35.

General Decomposition. For arbitrary L, define

$$\mathbf{A} = \frac{1}{2}\left(\mathrm{Sig}^{(2)} - \mathrm{Sig}^{(2)\top}\right). \tag{38}$$

Then

$$\mathbf{A} = \frac{1}{2}\sum_{i<j}\left(\mathbf{d}_i \otimes \mathbf{d}_j - \mathbf{d}_j \otimes \mathbf{d}_i\right), \tag{39}$$

while

$$\frac{1}{2}\left(\mathrm{Sig}^{(2)} + \mathrm{Sig}^{(2)\top}\right) = \frac{1}{2}\,\mathrm{Sig}^{(1)} \otimes \mathrm{Sig}^{(1)}. \tag{40}$$

Hence,

$$\mathrm{Sig}^{(2)} = \frac{1}{2}\,\mathrm{Sig}^{(1)} \otimes \mathrm{Sig}^{(1)} + \mathbf{A}. \tag{41}$$

The symmetric component is determined by the first-order signature, whereas $\mathbf{A}$ contains additional ordered cross-step information. This establishes order sensitivity, not full injectivity: distinct longer paths may still share the same truncated second-order signature.

C.3 Prefix of the Deployed Scorer

The following claim concerns the deployed scorer with fixed learned parameters and an offline snapshot partition; it does not require the learned parameters to be independent of target snapshots used during training.

Proposition 3 (Prefix scoring invariance).

Fix learned online parameters Θ, node identities, and an offline snapshot partition. Consider two streams $\mathcal{S}$ and $\mathcal{S}'$ with the same query (u, v, τ) in target bin t. If

$$\mathcal{C}_t(\mathcal{S}) = \mathcal{C}_t(\mathcal{S}'), \qquad \mathcal{H}_{<\tau}(\mathcal{S}) = \mathcal{H}_{<\tau}(\mathcal{S}'), \tag{42}$$

then

$$s^{\tau}_{uv}(\mathcal{S};\Theta) = s^{\tau}_{uv}(\mathcal{S}';\Theta). \tag{43}$$

Proof. At inference, $\widetilde{G}_t$, $\mathrm{GNN}_{\bar{\theta}}$, and $g_{\bar{\theta}}$ are not used, while the online encoders remain part of the scorer. Equality of $\mathcal{C}_t$ gives identical historical representations:

$$\begin{aligned} \mathbf{c}^t_u(\mathcal{S}) &= \mathbf{c}^t_u(\mathcal{S}'), \qquad \mathbf{c}^t_v(\mathcal{S}) = \mathbf{c}^t_v(\mathcal{S}'), \\ \mathbf{g}^t_{uv}(\mathcal{S}) &= \mathbf{g}^t_{uv}(\mathcal{S}'), \quad \mathbf{S}^t_{uv}(\mathcal{S}) = \mathbf{S}^t_{uv}(\mathcal{S}'). \end{aligned} \tag{44}$$

Since

$$\mathbf{q}^t_{uv} = \phi_{\mathrm{fuse}}\left([g_\theta(\chi(\mathbf{c}^t_u, \mathbf{c}^t_v)) \| \mathbf{g}^t_{uv} \| \mathbf{S}^t_{uv}]\right), \tag{45}$$

we have $\mathbf{q}^t_{uv}(\mathcal{S}) = \mathbf{q}^t_{uv}(\mathcal{S}')$. Equality of the strict prefixes similarly gives $\mathbf{o}^{<\tau}_{uv}(\mathcal{S}) = \mathbf{o}^{<\tau}_{uv}(\mathcal{S}')$. Therefore,

$$\begin{aligned} \mathbf{z}^{\tau}_{uv}(\mathcal{S}) &= \phi_\tau\left([\mathbf{q}^t_{uv}(\mathcal{S}) \| \mathbf{o}^{<\tau}_{uv}(\mathcal{S})]\right) \\ &= \phi_\tau\left([\mathbf{q}^t_{uv}(\mathcal{S}') \| \mathbf{o}^{<\tau}_{uv}(\mathcal{S}')]\right) \\ &= \mathbf{z}^{\tau}_{uv}(\mathcal{S}'). \end{aligned} \tag{46}$$

Since $s^{\tau}_{uv} = h_\psi\left(\mathbf{z}^{\tau}_{uv}\right)$, the scores are identical.

Scope of the Claim. Proposition 3 assumes a fixed offline partition. Given this partition, deployed scoring uses only $\mathcal{C}_t$ and $\mathcal{H}_{<\tau}$.

The claim does not require $\mathbf{y}^t_{uv}$ to be independent of the queried interaction. Although $g_{\bar{\theta}}$ receives no explicit target-bin pair counts or pair-resolved event features, its endpoint latents are computed from $\widetilde{G}_t$ and may depend indirectly on its adjacency. Target-bin information may therefore affect learned parameters during training, but $\widetilde{G}_t$, $\mathbf{y}^t_u$, and $\mathbf{y}^t_{uv}$ are not scoring inputs at inference.