FORGET LESS, GENERALIZE MORE: UNIFYING TEMPORAL AND STRUCTURAL ADAPTATION FOR DYNAMIC GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representation learning on dynamic graphs requires capturing complex dependencies that evolve across both time and structure. Existing approaches typically adopt fixed temporal decay schemes or predetermined structural propagation depths, limiting their ability to generalize across graphs with diverse interaction frequencies and topological characteristics. We propose Dual-Scale Retentive Dynamics (DSRD), a unified framework that maintains a retentive representation state encoding both temporal memory and structural context. DSRD introduces two key components: (i) a retentive state with dual-scale adaptation that jointly models temporal dynamics and structural propagation within a single recurrent formulation, and (ii) adaptive decay kernels with learnable time-sensitivity parameters that automatically balance short-term responsiveness and long-term retention based on the underlying temporal and structural interaction patterns. We provide theoretical analysis that expands the recurrent state updates into weighted aggregation over time-respecting walks, together with boundedness guarantees for the learned dynamics. Extensive experiments on 14 real-world benchmarks demonstrate that DSRD achieves state-of-the-art performance on both link prediction and node classification tasks, with strong generalization across transductive and inductive settings. The implementation code is available at [Code].

1 INTRODUCTION

Dynamic graphs (also referred to as temporal graphs) offer a principled abstraction for modeling real-world systems whose entities and relations evolve over time (Holme & Saramäki, 2012; Feng et al., 2025; Martínez et al., 2016), enabling structured reasoning over temporal interactions in domains such as social networks (Deng et al., 2019), recommendation systems, traffic systems (Li et al., 2022), financial transaction networks, and knowledge graphs (Wang et al., 2024). However, learning effective representations from such data remains challenging due to the tight entanglement between temporal dynamics and evolving topological structure, which constrains both the expressiveness and robustness of existing dynamic graph learning models (Feng et al., 2025; Gravina & Bacciu, 2024).

Recent advances in dynamic graph learning span memory-based architectures (Rossi et al., 2020; Kumar et al., 2019), attention-driven temporal encoders (Yu et al., 2023; Xu et al., 2020), and temporal walk-based structural models (Wang et al., 2021b; Lu et al., 2024). While these approaches have achieved notable empirical success, they predominantly rely on a single temporal scale or a fixed structural propagation mechanism (Nguyen et al., 2018; Holme & Saramäki, 2012). As a result, they struggle to adapt to heterogeneous dynamic regimes, where both temporal patterns and structural dependencies vary substantially across datasets (e.g., sparse vs. dense graphs) and downstream tasks (e.g., node classification vs. link prediction) (Feng et al., 2025; Chang et al., 2026).

The first challenge is the **temporal trade-off**, arising from the coexistence of heterogeneous temporal regimes within and across dynamic graphs. For instance, as illustrated in Figure 1(a), gradual interaction patterns encode long-term relational dependencies (Yu et al., 2023), whereas bursty interactions reflect short-lived but informative events. Real-world dynamic graphs often exhibit both behaviors simultaneously, yet most existing models rely on fixed decay or uniform temporal encoding.

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.

This prevents adaptive balancing between short-term and long-term dependencies (Cong et al., 2023; Wang et al., 2021b; Chung et al., 2025). The second challenge is **structural conflation**, which concerns the loss of causal walks in temporal propagation. As shown in Figure 1(b), two walks with identical node sequences but different edge timestamps represent distinct information flow patterns (order of $A \to B$ and $A \to C$). However, conventional aggregation mechanisms treat them identically (Lu et al., 2024), collapsing high-order temporal dependencies that can be critical for downstream prediction (Jin et al., 2022; Ur Rahman et al., 2025).

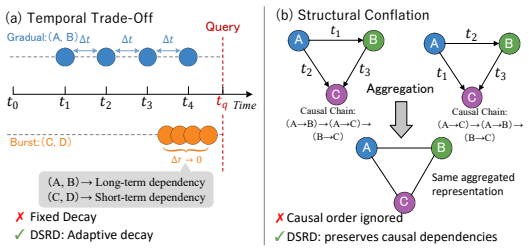


Figure 1: An illustration of the challenges in dynamic graph learning. (a) Temporal Trade-Off: Gradual (A, B) and burst (C, D) interaction patterns require both long-term and short-term dependencies, respectively, yet fixed decay cannot distinguish them. (b) Structural Conflation: Different causal orderings in temporal walks yield identical representations under standard aggregation.

To address these challenges, we propose **D**ual-**S**cale **R**etentive **D**ynamics (**DSRD**), a unified framework that jointly models temporal and structural evolution within a shared retentive state. Rather than treating temporal dependencies and structural propagation as independent modules, DSRD *couples them through a shared state representation, allowing the model to learn graph-level trade-offs between short-term and long-term dependencies*. Specifically, DSRD introduces a dynamic graph operator over the retentive state, coupling two mechanisms that target the two challenges above: (i) **adaptive temporal retention**, which equips the retentive state with learnable decay kernels and time-sensitivity parameters that automatically balance short-term responsiveness against long-term memory; and (ii) **structural diffusion**, which propagates signals along temporal walks, preserving the causal ordering collapsed by standard aggregation. Both mechanisms are fused into this operator through dual-scale adaptation (Section 3 and Appendix B). Temporal and structural adaptation are thus learned jointly within a single operator rather than in separate stages. We provide theoretical analysis showing that the retentive dynamics admit a state-space formulation and satisfy a geometric interlayer equivalence over temporal walks (Section 2.2 and Appendix A). Experiments on 14 benchmarks show that DSRD achieves state-of-the-art performance on dynamic link prediction and node classification under both transductive and inductive settings, attaining an average rank of 1.29 across the link-prediction benchmarks, while remaining efficient and scalable (Section 4 and Appendix C).

Our contributions are threefold:

- We propose DSRD, a unified retentive-state operator for dynamic graphs that fuses adaptive temporal retention and high-order structural diffusion within a *single* state, instead of treating temporal dynamics and structural propagation as separate modules.
- We design a dual-scale adaptive decay mechanism with learnable time-sensitivity parameters that jointly govern short-term responsiveness and long-term retention, aligning the model with each graph's temporal and structural interaction patterns.
- We conduct extensive experiments on 14 benchmarks, where DSRD attains state-of-the-art performance under both transductive and inductive settings across discrete- and continuous-time graphs, while remaining efficient.

2 Preliminaries

2.1 Problem Formulation

Dynamic Graphs. We consider a dynamic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, \Phi)$, where $\mathcal{V}$ denotes the node set and $\mathcal{E} = \{(u, v, t)\}$ is a set of interaction events with timestamps $t \in \mathbb{R}_{\geq 0}$. Each node $v \in \mathcal{V}$ is associated with a feature vector $x_v \in X$, and each event $e = (u, v, t_e)$ carries an edge attribute $\phi_e \in \Phi$. Depending on the temporal granularity, dynamic graphs can be modeled in two principal paradigms: *Discrete-Time Dynamic Graphs (DTDGs)*, which represent temporal evolution as a sequence of graph snapshots aggregated over fixed time intervals, and *Continuous-Time Dynamic Graphs (CTDGs)*, which capture fine-grained dynamics through asynchronous event streams where

each edge is associated with an exact timestamp. At any reference time t, the observable snapshot is the causal subgraph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, X, \Phi_t)$, where $\mathcal{E}_t = \{(u, v, t_e) \in \mathcal{E} \mid t_e \leq t\}$.

Temporal Walks. High-order structural context can be characterized through *temporal walks*, which are sequences of edges that respect the chronological ordering of interactions. To quantify the influence propagated along such walks, we introduce a diffusion score matrix $A_t \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where entry $A_t(i, j)$ measures the aggregated weight of all time-respecting walks from node i to node j up to time t. Rather than serving as explicit node representations, these scores act as structural routing coefficients that govern how temporally localized signals propagate across the graph.

Learning Objective. Given a node v at time t, the goal is to learn a representation $Z_v(t) = f_\Theta(v, \mathcal{G}_t)$ with all trainable parameters Θ that encodes all relevant structural and temporal information up to time t. The learned representation can be used for tasks such as *link prediction*, where the model estimates the likelihood of a future interaction (u, v, t^+), and *node classification*, where the model predicts the ground-truth label of v at time t^+. Both transductive and inductive evaluation settings are considered.

2.2 Retentive State

Given the causal subgraph $\mathcal{G}_t$, each node j at reference time t is associated with a feature vector $x_j(t) \in \mathbb{R}^{d_x}$. We apply shared linear projections

$$Q_t^j = x_j(t)W_Q,\ K_t^j = x_j(t)W_K,\ V_t^j = x_j(t)W_V, \tag{1}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_x\times d}$ are learnable parameters.

To any interaction event $(i, j, t_e) \in \mathcal{E}_t$, we associate the rank-one second-order increment $(K_t^i)^\top V_t^i \in \mathbb{R}^{d\times d}$. This outer-product form encodes pairwise feature interactions emitted by node i at t for both CTDGs and DTDGs, serving as the atomic unit of information injection. A straightforward approach aggregates all historical increments uniformly into the matrix $\mathcal{H}_t^j = \sum_{(i,j,t_e)\in\mathcal{E}_t}(K_{t_e}^i)^\top V_{t_e}^i$, from which the node-level output is computed via query contraction $O_t^j = Q_t^j\mathcal{H}_t^j$ (Zhai et al., 2021; Peng et al., 2023; Sun et al., 2023; Gu & Dao, 2024). This formulation treats all past interactions in a flat and parallel manner, without modeling (i) how information is temporally retained and (ii) how it diffuses structurally. To disentangle these mechanisms, we first introduce weighted instantaneous injections and layer-wise temporal walks.

Instantaneous Injection. For the target node j, we define its instantaneous injection from an arbitrary source node i, at time t, as a weighted aggregation of source messages:

$$\Delta_t^i = \sum_{(i,j,t_e)\in\mathcal{E}_t} \omega(t, (i, j, t_e))\,(K_{t_e}^i)^\top V_{t_e}^i, \tag{2}$$

where $\omega(t, (i, j, t_e)) \geq 0$ is a causal weight that modulates the contribution of each historical interaction based on temporal proximity and semantic relevance. This quantity represents the localized signal emitted by source node i.

Temporal Walk Diffusion. Let $T_t \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ be the instantaneous transition matrix induced by $\mathcal{E}_t$. Given a maximum diffusion depth K, we formalize recursive walk diffusion kernels $\{A_t^{(\ell)}\}_{\ell=0}^K$ as

$$A_t^{(\ell)} = A_{t^-}^{(\ell)} + A_{t^-}^{(\ell-1)}T_t, \quad \text{for } t, \ell > 0, \tag{3}$$

with $A_t^{(0)} = I$ for all t, and $A_0^{(\ell)} = 0$ for $\ell > 0$. Here t^- denotes the most recent event time right before t. Recall that $A_t^{(\ell)}(i, j)$ denotes the aggregated weight of length-ℓ time-respecting walks from i to j up to time t. To gain more insight into the walk matrix, we provide a closed-form expansion of $A_t^{(\ell)}$ in Appendix A (see Lemma A.1).

Retentive State Dynamics. To couple short-term injections with long-term structural diffusion, we define a latent retentive state $S_t^{j,(\ell)} \in \mathbb{R}^{d\times d}$ for each node j, diffusion depth $\ell > 0$, and time $t > 0$:

$$S_t^{j,(\ell)} = a_t S_{t^-}^{j,(\ell)} + b_t \bar{\Delta}_t^{j,(\ell)} \qquad S_0^{j,(\ell)} = 0, \tag{4}$$

where $a_t, b_t \in [0, 1]$ control long-term retention and short-term injection strength, respectively. Note that $\bar{\Delta}_t^{j,(\ell)} = \sum_{i\in\mathcal{V}} A_t^{(\ell)}(i, j)\Delta_t^i$. Although the summation is written over all nodes, only temporally reachable nodes contribute non-zero terms, since $A_t^{(\ell)}(i, j) = 0$ whenever no temporal walk of length ℓ exists from i to j. Since Equation (4) defines this state only implicitly, we unroll the recurrence into the following closed form to expose how each past injection contributes.

Proposition 2.1. *Let $\mathcal{T}_t$ denote the ordered set of event timestamps up to t. For each node j and diffusion depth ℓ, the retentive state admits the closed-form expansion*

$$S_t^{j,(\ell)} = \sum_{\tau\in\mathcal{T}_t} \left(\prod_{r\in\mathcal{T}_{(\tau,t]}} a_r \right) b_\tau \bar{\Delta}_\tau^{j,(\ell)} \tag{5}$$

where $\mathcal{T}_{(\tau,t]} = \{r \in \mathcal{T}_t | \tau < r \le t\}$.

The proof of Proposition 2.1 is provided in Appendix A.1. Note that Equation (5) unifies temporal retention and high-order structural diffusion within a single recurrent state. For each diffusion depth ℓ, the resulting readout $O_t^{j,(\ell)} = Q_t^j S_t^{j,(\ell)}$ captures a temporally filtered and structurally diffused summary of all historical second-order injections.

3 Dual-Scale Retentive Dynamics

In this section, we instantiate the DSRD framework with two core mechanisms: adaptive temporal retention and structural diffusion (see Section 3.1). We additionally describe auxiliary components including time encoding, feature embedding, and the overall block architecture (see Section 3.2).

3.1 Adaptive Dual-Scale Retentive Diffusion

We model dynamic graph representation learning as a retentive diffusion process in which long-term temporal memory and high-order structural propagation are jointly maintained in a unified latent state, as illustrated in Figure 5 (Appendix B). For each target node j and diffusion layer ℓ, the model maintains a second-order latent state $S_t^{j,(\ell)} \in \mathbb{R}^{d\times d}$, which is updated at any time t when an interaction occurs. Following the unified retentive formulation established in Proposition 2.1, we introduce our strategies:

Short-Term Injection with Adaptive Decay. To separate instantaneous event responses from long-term structural retention, we instantiate the short-term injection Δ_t^i from Equation (2) by specifying the causal weight as

$$\omega\big(t, (i, j, t_e)\big) = \underbrace{\sigma\big(\langle Q_t^j, K_{t_e}^i\rangle\big)}_{\text{query–key attention}} \cdot \underbrace{\exp\big(-\lambda(\Delta t_e)^\alpha\big)}_{\text{adaptive temporal decay}}. \tag{6}$$

Here, the attention factor computes the dot product $\langle Q_t^j, K_{t_e}^i\rangle$ between query and key vectors and applies sigmoid $\sigma(\cdot)$ to obtain a context-aware gating coefficient in $(0, 1)$. The second factor introduces adaptive temporal decay over the elapsed time $\Delta t_e = t - t_e$, where $\lambda > 0$ is a learnable per-layer decay rate (parameterized via softplus) and $\alpha \in (0, 1)$ is a learnable per-layer sensitivity exponent (parameterized via sigmoid). Although λ and α are shared within each layer, they shape a weight ω that stays event-specific through the query–key gate and each event's interval Δt_e, allowing bursty and gradual interactions to be balanced within a single graph.

Gated Long-term and Short-term Fusion. The short-term injection $\bar{\Delta}_t^{j,(\ell)} = \sum_{i\in\mathcal{V}} A_t^{(\ell)}(i, j)\Delta_t^i$ captures instantaneous event signals, while the accumulated state $S_{t^-}^{j,(\ell)}$ preserves long-range historical dependencies. To balance these two components, we introduce a learnable gating mechanism.

Formally, we take $a_t = \gamma^{(\ell)}$ and $b_t = 1 - \gamma^{(\ell)}$ in Equation (4), where $\gamma^{(\ell)} \in (0, 1)$ is a per-layer learnable scalar gate (parameterized via sigmoid), shared across all nodes and events within layer ℓ, yielding the convex combination

$$S_t^{j,(\ell)} = \gamma^{(\ell)} S_{t^-}^{j,(\ell)} + (1 - \gamma^{(\ell)}) \bar{\Delta}_t^{j,(\ell)}. \tag{7}$$

A higher $\gamma^{(\ell)}$ retains more historical information, favoring long-term dependencies, while a lower $\gamma^{(\ell)}$ allows the model to respond more rapidly to recent events.

Topological Diffusion via Temporal Walks. We note that the gated update in Equation (7) requires computing $\bar{\Delta}_t^{j,(\ell)} = \sum_{i \in \mathcal{V}} A_t^{(\ell)}(i, j)\Delta_t^i$, which involves the ℓ-hop walk matrix $A_t^{(\ell)} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. Explicitly maintaining this matrix for each layer ℓ incurs $\mathcal{O}(K|\mathcal{V}|^2)$ memory overhead, which is prohibitive for large-scale graphs. To circumvent this, we introduce layer-specific projection matrices $W_K^{(\ell)}, W_V^{(\ell)}$ and define the layer-ℓ short-term injection as

$$\tilde{\Delta}_t^{j,(\ell)} = \sum_{(i,j,t_e) \in \mathcal{E}_t} \omega\big(t, (i, j, t_e)\big) \left(\tilde{K}_{t_e}^{i,(\ell)}\right)^\top \tilde{V}_{t_e}^{i,(\ell)}, \tag{8}$$

where $\tilde{K}_{t_e}^{i,(\ell)}$ and $\tilde{V}_{t_e}^{i,(\ell)}$ follow the same augmented form as Equation (1) but with layer-specific parameters (see Section 3.2). This directly aggregates contributions from all incident edges at time t, without requiring the walk matrix $A_t^{(\ell)}$. Substituting into Equation (7), the state update becomes:

$$S_t^{j,(\ell)} = \gamma^{(\ell)} S_{t^-}^{j,(\ell)} + (1 - \gamma^{(\ell)}) \tilde{\Delta}_t^{j,(\ell)}. \tag{9}$$

Since Equation (9) shares the recurrence form of Equation (7), the same gated dynamics govern both, now with $\tilde{\Delta}_t^{j,(\ell)}$ in place of $\bar{\Delta}_t^{j,(\ell)}$. Its short-term injection stays uniformly bounded under the neighborhood normalization of Section 3.2, as established in Lemma A.3.

Finally, the readout for each layer is obtained via query contraction $O_t^{j,(\ell)} = Q_t^{j,(\ell)} S_t^{j,(\ell)}$, where we extend the projection defined in Equation (1) to each layer ℓ by applying layer-specific parameters. The state is further updated by observing that the temporal walk matrix satisfies a recursive structure: when a new interaction (i, j, t_e) arrives, existing $(\ell-1)$-step walks ending at i extend to ℓ-step walks ending at j. This motivates a layer-wise state propagation scheme that implicitly encodes the walk matrix without explicit materialization (an illustrative example is provided in Appendix B.2). Proposition A.4 in Appendix A.3 shows that the resulting state admits a closed-form expansion over the same time-respecting walks that define $A_t^{(\ell)}$, with learned per-hop attenuation and retention weights in place of uniform walk counts. Specifically, we propagate information across walk lengths:

$$S_t^{j,(\ell-1)} \leftarrow S_t^{j,(\ell-1)} + \sum_{(i,j,t_e) \in \mathcal{E}_t} \psi_{ij}^{(\ell)}(t_e)\, S_t^{i,(\ell)},\ \ell = K, \ldots, 2, \tag{10}$$

where $\psi_{ij}^{(\ell)}(t_e) = \exp(-\ell\delta^{(\ell)}\Delta t_e)$ is a trainable attenuation with a learnable parameter $\delta^{(\ell)} > 0$. The readout precedes topological diffusion to preserve causality, while the short-term injection $\tilde{\Delta}_t^{j,(\ell)}$, depending only on immediate neighbors, remains unaffected, as shown in Algorithm 1 in Appendix B.1. With normalized structural weights $\tilde{\psi}_{ij}^{(\ell)}$ (see Section 3.2), the state after the full update remains uniformly bounded over time, which we now state formally.

Theorem 3.1 (Uniform boundedness of the retentive state)**.** *Suppose the augmented inputs $x_v + \tilde{\phi}_e$ and projection matrices of Section 3.2 are bounded, $\|x_v + \tilde{\phi}_e\| \le B_x$ and $\|W_K\|, \|W_V\| \le L$, with the temporal and structural weights normalized over each target neighborhood. Set $M = L^2 B_x^2$. Then, for every node j, layer ℓ, and time t, the state after the full update (Equations* (9) *and* (10)*) and its readout satisfy*

$$\textit{(i)}\ \|S_t^{j,(\ell)}\| \le B_\ell, \qquad \textit{(ii)}\ \|O_t^{j,(\ell)}\| = \|Q_t^{j,(\ell)} S_t^{j,(\ell)}\| \le L B_x B_\ell, \tag{11}$$

both independent of the number of events, where the finite constants B_ℓ satisfy $B_K = M$ and $B_{\ell-1} = M + B_\ell/(1 - \gamma^{(\ell-1)})$ for $\ell = K, \ldots, 2$.

We defer the proof to Appendix A.2. The bound B_ℓ grows with the retention strength through the geometric factor $1/(1 - \gamma^{(\ell-1)})$ but stays finite for any fixed K and $\gamma^{(\ell)} \in (0, 1)$, so the retentive state cannot grow without bound as the event stream lengthens.

3.2 Network Composition and Optimization Objective

Auxiliary Augmentation. To model both transient interactions and long-term dependencies in dynamic graphs, DSRD is organized as a hierarchical composition of model layers. Specifically, for each event (u, v, t_e), we compute a time encoding (Cong et al., 2023) $\phi_t = \cos(\Delta t_e \, W_{t_e})$ and fuse it with the edge attribute to obtain $\tilde{\phi}_e = (\phi_e + \phi_t)W_e$, where W_{t_e} and W_e are trainable parameters. The temporal-aware edge features are injected into node representations by augmenting the projections:

$$\tilde{K}^i_{t_e} = K^i_{t_e} + \tilde{\phi}_e W_K, \quad \tilde{V}^i_{t_e} = V^i_{t_e} + \tilde{\phi}_e W_V. \tag{12}$$

These augmented projections replace $(K^i_{t_e}, V^i_{t_e})$ in Equation (2). As a result, the increment $(\tilde{K}^i_{t_e})^\top \tilde{V}^i_{t_e}$ encodes not only the node features but also edge-specific temporal signals, enabling the model to differentiate between interactions that share the same target nodes but occur at different times or carry distinct attributes.

Model Architecture. Within the ℓ-th layer, we extend the retentive state to H parallel attention heads, where each head $h \in \{1, \ldots, H\}$ maintains independent parameters $(\omega^{h,(\ell)}, \gamma^{h,(\ell)}, \psi^{h,(\ell)})$ as defined in Section 3.1. The node representation $Z^{(\ell)}$ for each node j is updated as

$$Z^{(\ell)} = X^{(\ell)} + \mathrm{LN}\Big(\mathrm{Concat}(\, g^h_\ell(X^{(\ell)}, S^{j,h,(\ell)}))\Big), \tag{13}$$

where $g^h_\ell(\cdot)$ denotes the DSRD operator for head h, Concat denotes concatenation across heads, and LN represents layer normalization (Ba et al., 2016) applied across the concatenated feature channels. Following the core DSRD operation, a feed-forward sublayer (Vaswani et al., 2017) refines the representation as

$$X^{(\ell+1)} = Z^{(\ell)} + \mathrm{GELU}\big(\mathrm{LN}(Z^{(\ell)})\, W_1^{(\ell)}\big)\, W_2^{(\ell)}, \tag{14}$$

where GELU denotes the Gaussian Error Linear Unit activation (Hendrycks, 2016) and $W_1^{(\ell)}, W_2^{(\ell)}$ are learnable parameters. The alternating application of multi-head retentive propagation and feed-forward refinement enables each layer to encode both fine-grained temporal signals and higher-order structural context.

Normalization. To stabilize the numerical flow, we apply several normalization techniques that preserve scale-invariant properties: i) The readout $O^{j,(\ell)}_t$ for each head h is scaled as $Q^{j,(\ell)}_t S^{j,(\ell)}_t / \sqrt{d}$; ii) The structural propagation weights $\psi^{(\ell)}_{ij}(t)$ and temporal attention weights ω_{ij} are each normalized over the corresponding neighborhood as $\tilde{\psi}^{(\ell)}_{ij}(t) = \psi^{(\ell)}_{ij}(t) / \max(\sum_{i'} \psi^{(\ell)}_{i'j}(t), 1)$ and $\tilde{\omega}_{ij} = \omega_{ij} / \max(\sum_{i'} |\omega_{i'j}|, 1)$; iii) All time intervals Δt involved in decay functions are transformed using $\log(1 + \Delta t)$ prior to exponentiation, improving numerical stability across various time scales (from days to years).

After stacking K DSRD layers, the final node embedding is obtained by mean pooling over the H heads, $h^u_t = \mathrm{MEAN}^H_{h=1}(O^{u,h,(K)}_t)$. For link prediction, the likelihood of a future interaction (u, v, t^+) is computed as $p^{u,v}_t = \mathrm{MLP}([h^u_t, h^v_t])$. For node classification, the label probability is given by $\hat{y}^u_t = \sigma(h^u_t W_{\mathrm{cls}})$, where W_{cls} is a learnable weight matrix.

4 Experiments

In this section, we conduct extensive experiments on real-world datasets with diverse downstream tasks and settings to evaluate DSRD on effectiveness and scalability.

4.1 Setup

Datasets and Baselines. We conduct experiments on fourteen benchmark datasets sourced from real-world dynamic systems collected by Poursafaei et al. (2022), including Wikipedia, Reddit, MOOC, Myket, LastFM, Enron, Social Evo., UCI, Flights, Can. Parl., USLegis., UN Trade, UN Vote, and Contact. Since the original node classification datasets are extremely imbalanced, we further include

PubMed as a less severely imbalanced biomedical dataset. Specifically, the positive-class ratios are only 0.14% on Wikipedia, 0.05% on Reddit, and 0.99% on MOOC, while PubMed (ARA Passos et al., 2024) contains 19,732 (22.26%) and 68,916 (77.74%) instances in the two classes. Twelve strong dynamic graph learning methods are included as baselines, i.e., JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020), TGN (Rossi et al., 2020), CAWN (Wang et al., 2021b), EdgeBank (Poursafaei et al., 2022), TCL (Wang et al., 2021a), GraphMixer (Cong et al., 2023), NAT (Luo & Li, 2022), PINT (Souza et al., 2022), DyGFormer (Yu et al., 2023), and TPNet (Lu et al., 2024). We further compare against four recent methods, i.e., CTWalks (Yu & Liu, 2025), TAWRMAC (Farokhi et al., 2026), GraphHeuristic (Cornell et al., 2025), and GRN (Chang et al., 2026), with results reported in Appendix C.4 and C.5. Additional details about datasets and baselines are provided in the Appendix C.1 and C.2.

Evaluation Settings. As stated in Section 2.1, we evaluate all baselines and our method on dynamic link prediction and dynamic node classification tasks, and the link prediction task is further assessed under both transductive and inductive settings in order to capture performance across different generalization regimes. Three negative sampling strategies (i.e., random, historical and inductive negative sampling strategies) are used for evaluation, where we employ the random negative sampling strategy for model training. For evaluation metrics, Average Precision (AP) is used to measure predictive precision, while the Area Under the Receiver Operating Characteristic Curve (ROC-AUC) provides an estimate of sensitivity and robustness for both link prediction and node classification.

Implementation and Configurations. We follow the standard training pipeline and evaluation protocol proposed by the Dynamic Graph Library (DyGLib) (Yu et al., 2023) for the training and evaluation of all models. To ensure fair comparison, all methods are trained and evaluated under identical pipelines with chronological data splits and the same negative sampling protocols. More details about the implementation and configurations are provided in Appendix C.3

4.2 Main Experimental Results

Across all benchmarks and evaluation protocols, the proposed method achieves the strongest overall performance (see statistical tests in Appendix C.6), attaining outstanding average ranks on dynamic link prediction (Tables 1, 4, 5, 6, 7, and 8) and dynamic node classification tasks (Table 2), which establishes its overall effectiveness and scalability. We further observe the following:

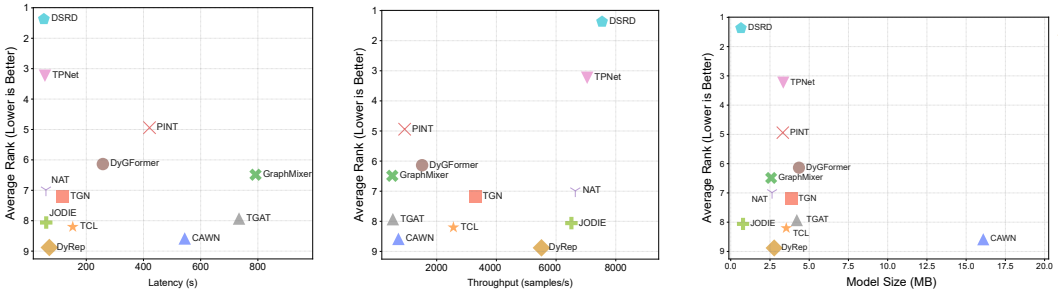
(i) Generalization. Across various evaluation settings, including multiple negative sampling strategies as well as transductive and inductive splits, performance remains strong. In particular, generalization to unseen nodes under inductive evaluation is stable, whereas several baseline methods exhibit noticeable degradation under the same conditions (see Tables 1, 7, and 8). **(ii) Downstream tasks.** Across downstream tasks, competitive results are observed for both link prediction and node classification. Link prediction performance remains top-ranked across datasets and settings, while node classification results achieve the best average rank, with the highest ROC-AUC attained on MOOC and competitive performance maintained under highly imbalanced label distributions (see Appendix C.7). **(iii) Dataset regimes.** Across datasets with diverse temporal and structural characteristics (see Table 3), consistent performance trends emerge. Clear advantages are observed on discrete-time dynamic graphs with coarse temporal granularity, competitive results are maintained on continuous-time graphs with fine-grained timestamps, and strong performance is achieved on high-density graphs with rich structural connectivity (see Sections 4.3 and 4.4), while the high scores reached by several methods under certain protocols reflect a regime-dependent effect analyzed in Appendix C.8. **(iv) Scalability.** In terms of both efficiency and scalability, favorable characteristics are observed, as shown in Figure 2 and Table 10. DSRD attains lower running latency (i.e., running time) and higher throughput than existing methods while using fewer parameters, and its memory grows only mildly as the event stream scales to millions of edges. Detailed time complexity and memory usage measurements are provided in Appendix C.10 and C.11.

4.3 How do dual-scale adaptive decays reveal the temporal and structural effects?

To examine whether the proposed joint decay mechanism can adaptively align with the diverse temporal and structural dynamics, we analyze the learned decay behaviors across all datasets. Specifically, we extract three types of decay patterns from the trained models, including long-term dependency

Table 1: AP for inductive dynamic link prediction with random, historical, inductive negative sampling strategies (NSS). Avg. Rank is computed across datasets for each NSS. The best model is shown in **bold** and the second best model is underlined. Shaded cells indicate relative performance across top 50% models on a given dataset.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	PINT	NAT	DyGFormer	TPNet	DSRD
rnd	Wikipedia	89.46±0.25	87.45±0.24	90.31±0.30	93.90±0.08	97.70±0.01	91.16±0.32	95.10±0.05	97.77±0.09	91.97±0.11	97.89±0.06	98.72±0.00	**99.55±0.10**
	Reddit	93.21±0.23	91.76±0.25	92.66±0.50	93.48±0.27	98.03±0.04	85.47±0.30	94.55±0.03	98.36±0.05	94.29±0.09	98.30±0.01	98.33±0.07	**99.94±0.02**
	MOOC	76.27±3.23	71.37±1.27	78.42±0.50	85.20±0.76	73.46±0.43	77.08±0.13	79.41±0.19	86.88±0.59	80.38±0.47	75.34±1.69	88.05±0.69	**95.75±0.66**
	LastFM	82.91±0.14	79.93±3.58	75.19±0.23	77.90±1.50	88.16±0.23	79.53±2.04	82.13±0.25	89.54±0.88	89.18±0.74	93.01±0.33	94.13±0.20	**99.93±0.04**
	Myket	82.44±0.27	77.34±0.71	56.33±1.25	70.55±0.37	76.99±0.72	56.70±1.10	80.90±0.02	78.63±0.17	64.09±2.58	72.02±0.17	84.24±0.25	**99.92±0.02**
	Enron	75.97±2.39	66.57±3.13	63.21±1.65	68.44±3.23	85.47±0.20	65.31±0.93	75.25±0.46	87.00±0.10	84.13±1.02	82.39±0.50	88.17±0.08	**93.78±1.10**
	Social Evo.	89.32±0.56	81.80±0.78	90.63±0.22	85.03±1.18	79.59±0.06	92.06±0.11	91.35±0.04	91.62±0.10	82.81±7.87	92.95±0.06	92.72±0.14	**92.97±2.00**
	UCI	72.18±1.78	48.44±1.09	78.63±1.02	79.20±0.40	90.86±0.37	80.97±0.16	90.74±0.41	94.60±0.12	81.95±0.40	93.61±0.14	**95.12±0.11**	92.59±0.05
	Flights	90.22±1.18	87.85±0.68	87.70±0.24	91.90±0.04	97.04±0.04	82.68±0.07	81.96±0.02	96.58±0.23	92.96±0.32	**97.77±0.01**	96.40±0.26	96.29±0.17
	Can. Parl.	50.98±2.45	52.66±1.02	54.52±0.84	47.37±1.83	55.10±0.98	54.22±1.07	55.56±0.55	53.48±0.48	55.48±1.31	66.01±0.34	58.08±1.71	**99.93±0.02**
	US Legis.	53.05±0.37	55.04±0.67	50.79±1.20	54.87±1.08	53.51±0.53	51.33±1.20	51.25±0.54	52.06±1.35	56.68±2.72	54.07±0.25	58.69±0.95	**84.42±1.03**
	UN Trade	57.36±0.47	54.82±0.58	60.06±0.15	52.77±1.66	62.94±0.46	61.31±0.16	59.11±0.43	59.81±2.00	67.39±4.51	59.05±0.79	79.56±3.86	**99.23±0.07**
	UN Vote	53.78±0.92	48.99±0.89	52.27±0.51	47.94±3.55	47.71±0.38	50.44±0.97	50.86±0.32	56.21±1.05	63.43±2.38	51.01±0.28	55.48±1.08	**96.32±0.66**
	Contact	92.90±0.71	88.17±0.26	95.01±0.11	88.83±2.33	88.55±1.14	90.22±0.37	90.17±0.03	95.86±0.57	93.91±0.25	**97.92±0.02**	97.89±0.08	96.35±0.42
	Avg. Rank	8.14	9.93	9.00	8.79	6.93	8.86	7.29	4.50	5.86	4.64	2.43	**1.64**
hist	Wikipedia	67.52±0.27	62.74±1.00	84.17±0.48	82.50±0.53	66.42±1.46	85.00±0.31	86.89±0.08	80.01±0.53	44.80±2.89	58.33±3.66	74.03±1.53	**87.68±0.58**
	Reddit	60.67±0.87	59.22±1.07	61.11±0.22	62.19±0.33	63.93±0.94	59.56±0.18	64.65±0.25	67.44±0.24	55.98±1.85	64.34±1.53	62.20±0.40	**91.69±2.18**
	MOOC	60.99±2.67	63.18±1.44	74.69±0.78	71.49±1.60	67.03±2.13	73.50±0.80	73.57±0.24	67.00±3.40	68.31±0.76	69.27±2.76	77.06±1.44	**88.19±1.94**
	LastFM	70.54±0.86	69.23±3.93	77.86±0.37	74.58±2.13	69.14±1.10	70.00±0.71	76.38±0.22	79.54±0.20	68.79±0.16	71.81±2.28	79.87±1.41	**98.93±0.03**
	Myket	47.99±0.04	48.83±0.87	50.72±0.38	50.43±0.34	44.41±3.53	50.73±0.60	52.18±0.09	49.71±0.00	38.52±0.97	33.96±0.11	38.68±0.43	**89.84±2.34**
	Enron	64.77±4.54	58.89±1.03	58.99±1.23	59.50±1.83	62.26±0.12	60.67±0.25	74.47±0.70	80.77±0.38	64.90±1.66	66.21±0.42	73.83±1.35	**92.72±0.47**
	Social Evo.	83.40±3.84	80.70±1.94	93.11±0.43	83.70±2.25	80.36±0.02	95.95±0.26	94.43±0.07	96.15±0.13	82.23±6.81	**96.95±0.08**	96.24±0.20	96.20±2.03
	UCI	65.89±2.70	54.66±0.08	68.36±1.13	68.83±0.51	54.99±1.69	68.73±0.69	80.00±0.24	85.77±0.01	52.65±1.04	58.44±0.63	77.60±0.71	**94.04±0.57**
	Flights	59.42±0.77	59.53±0.69	65.18±0.88	57.30±0.75	57.02±0.55	64.17±0.31	64.67±0.09	52.67±0.45	50.27±0.91	57.38±0.28	53.64±0.49	**75.18±1.16**
	Can. Parl.	49.19±1.84	50.66±0.63	57.06±0.21	48.27±0.74	56.18±1.02	54.43±1.33	55.48±0.63	51.65±0.48	56.15±0.65	67.54±0.31	60.15±2.04	**99.86±0.00**
	US Legis.	53.70±1.14	57.65±2.22	51.04±2.20	55.50±1.41	54.11±1.45	51.18±1.06	54.02±0.44	52.24±0.80	59.08±1.60	52.70±1.21	65.18±2.38	**66.21±1.33**
	UN Trade	55.37±0.73	56.36±1.62	52.04±1.12	54.36±1.09	55.29±2.56	55.18±1.56	53.14±1.76	56.02±1.59	68.21±3.53	61.16±2.59	68.32±1.53	**96.00±0.51**
	UN Vote	58.62±0.72	53.53±0.74	50.87±0.79	51.67±1.40	48.13±0.78	51.54±0.73	47.80±0.37	58.91±1.99	59.74±0.57	52.18±1.14	64.72±0.71	**89.96±2.68**
	Contact	88.08±0.66	83.14±0.68	94.24±0.16	83.07±6.09	72.15±0.78	90.87±0.29	89.84±0.40	89.10±0.78	82.82±0.96	92.80±0.34	94.38±0.95	**96.95±0.39**
	Avg. Rank	8.14	8.71	6.50	7.50	8.86	6.86	5.29	5.86	8.64	6.36	4.14	**1.14**
ind	Wikipedia	67.51±0.27	62.74±1.00	84.15±0.48	82.51±0.54	66.36±1.49	84.99±0.31	86.89±0.08	80.02±0.52	44.80±2.89	58.33±3.65	74.02±1.52	**87.77±0.59**
	Reddit	60.67±0.87	59.22±1.07	61.14±0.20	62.20±0.32	63.93±0.94	59.55±0.19	64.66±0.25	67.44±0.24	55.98±1.85	64.34±1.53	62.20±0.40	**91.70±2.25**
	MOOC	60.99±2.67	63.18±1.44	74.69±0.79	71.49±1.61	67.03±2.13	73.49±0.80	73.57±0.25	67.00±3.40	68.32±0.76	69.27±2.76	77.06±1.44	**88.59±1.69**
	LastFM	70.54±0.86	69.23±3.93	77.87±0.37	74.58±2.13	69.14±1.10	70.00±0.71	76.38±0.22	79.55±0.19	68.79±0.16	71.81±2.28	79.88±1.41	**98.92±0.04**
	Myket	48.00±0.04	48.83±0.87	50.72±0.38	50.44±0.34	44.41±3.53	50.73±0.60	52.18±0.09	49.72±0.00	38.52±0.97	33.96±0.11	38.68±0.43	**89.84±2.33**
	Enron	64.77±4.54	58.89±1.03	58.98±1.23	59.49±1.83	62.24±0.12	60.67±0.26	74.46±0.70	80.77±0.38	64.90±1.66	66.21±0.42	73.83±1.35	**92.50±0.43**
	Social Evo.	83.40±3.84	80.70±1.94	93.11±0.44	83.70±2.25	80.37±0.02	95.95±0.26	94.42±0.07	96.15±0.13	82.23±6.81	**96.95±0.08**	96.24±0.20	96.18±2.02
	UCI	65.88±2.72	54.68±0.08	68.40±1.15	68.86±0.52	55.02±1.72	68.81±0.71	80.03±0.24	85.78±0.01	52.66±1.04	58.51±0.65	77.62±0.71	**93.97±0.59**
	Flights	59.44±0.76	59.54±0.67	65.18±0.89	57.28±0.74	56.95±0.55	64.16±0.31	64.66±0.09	52.64±0.45	50.24±0.91	57.34±0.28	53.60±0.49	**75.20±1.15**
	Can. Parl.	49.22±1.85	51.00±0.63	57.13±0.22	48.16±0.66	56.38±1.27	54.47±1.53	55.72±0.71	51.60±0.50	56.11±0.59	67.56±0.31	60.09±2.29	**99.88±0.01**
	US Legis.	53.70±1.14	57.65±2.22	51.04±2.20	55.50±1.41	54.11±1.45	51.18±1.06	54.02±0.44	52.24±0.80	59.08±1.60	52.70±1.21	65.18±2.38	**65.80±0.86**
	UN Trade	55.28±0.73	56.33±1.65	51.86±1.07	54.34±1.07	55.29±2.57	55.49±1.58	53.34±1.80	56.02±1.59	68.16±3.53	61.24±2.63	68.33±1.55	**96.00±0.49**
	UN Vote	58.59±0.69	53.56±0.75	50.96±0.81	51.57±1.54	48.23±0.81	51.36±0.56	47.84±0.32	58.89±1.99	59.82±0.51	52.22±1.10	64.72±0.75	**90.20±2.74**
	Contact	88.08±0.66	83.14±0.67	93.87±0.62	83.07±6.09	72.16±0.78	90.86±0.29	89.84±0.40	89.11±0.78	82.82±0.96	92.80±0.34	94.38±0.96	**96.95±0.38**
	Avg. Rank	8.29	8.71	6.50	7.50	8.86	6.71	5.29	5.86	8.64	6.36	4.14	**1.14**
Overall Avg. Rank		8.19	9.12	7.33	7.93	8.21	7.48	5.95	5.40	7.71	5.79	3.57	**1.31**



(a) Latency per epoch (s) (b) Throughput (samples/s) (c) Model Size (MB)

Figure 2: Scalability comparison of dynamic graph methods. (a) Latency, (b) Throughput, and (c) Model size, each plotted against average rank (from Figure 7) across all datasets. All methods are run on the same hardware under an identical training pipeline and sampling budget (Appendix C.3).

decay γ governed by state update steps m (see Figure 3), short-term temporal decay ω shaped by time interval sensitivity (see Figure 9), and topological decay ψ reflecting hop-level topological attenuation (see Figure 10). Relative to the graph properties in Table 3, discrete-time graphs with high interaction scale (e.g., UN Trade, UN Vote, Can. Parl.) learn strong long-term retention, preserving information across sparse temporal snapshots. Dense proximity graphs (Contact, Social Evo.) show rapid state

Table 2: ROC-AUC for node classification. Avg. Rank is computed across datasets. Shaded cells indicate relative performance across top 50% models on a given dataset.

Dataset	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	PINT	NAT	DyGFormer	TPNet	DSRD
Wikipedia	86.11±1.21	**88.06±1.35**	69.78±0.73	86.63±0.38	80.03±1.74	73.59±5.49	84.70±0.16	86.27±0.26	64.98±1.39	84.35±1.27	83.06±0.69	87.72±1.14
Reddit	61.34±2.31	60.94±2.10	58.85±2.44	63.98±2.48	62.31±0.95	61.23±0.72	65.72±2.73	64.90±0.34	58.07±1.85	64.86±0.97	**65.72±1.09**	62.85±2.15
MOOC	68.80±0.77	68.74±1.76	63.89±0.61	69.24±0.55	64.88±0.81	70.14±1.22	67.02±0.56	68.03±0.32	55.86±5.48	76.04±0.70	64.91±2.63	**78.68±0.53**
PubMed	54.59±0.88	54.24±0.36	52.69±1.95	57.31±0.48	54.61±1.91	58.79±0.50	55.25±0.26	56.32±0.96	71.92±3.21	60.29±0.39	54.40±0.56	**76.46±1.03**
Avg. Rank	6.75	7.00	11.25	4.25	8.50	6.50	5.62	5.00	9.50	4.00	7.12	**2.50**

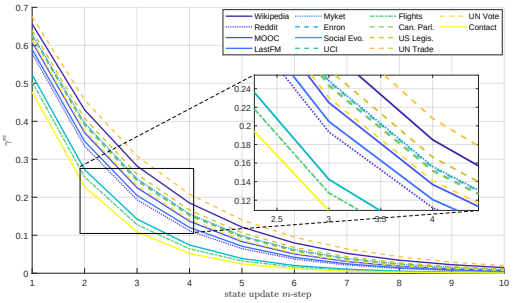


Figure 3: Cumulative long-term retention for all datasets, where $\gamma = \gamma^{(1)}$.

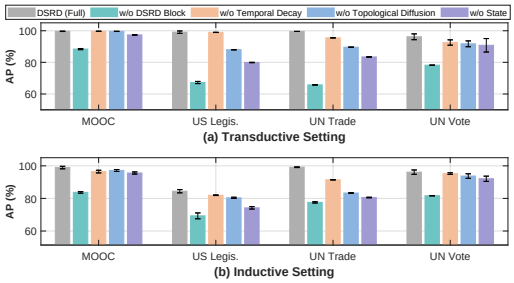


Figure 4: Partial ablation results of DSRD under transductive and inductive settings.

turnover but broad receptive fields, whereas sparse bipartite graphs (Reddit, Myket) exhibit high sensitivity to temporal recency and structural locality. Since Proposition A.4 expresses each walk contribution as a product of these decay factors, the learned parameters directly quantify the effective spatiotemporal receptive field of each dataset, complementing the aggregate benchmark results. In particular, recency-dominated graphs with heavy-tailed inter-event gaps learn a large λ and a small exponent $\alpha \to 0$ (see Appendix C.9).

4.4 Ablation Study

We conduct an ablation study to assess the contribution of individual components in DSRD. All variants are evaluated under identical data splits, training settings, and evaluation protocols as the full model. The following ablated variants are considered: *i)* w/o DSRD Block, which removes the entire DSRD module (Eqs. 2, 10, and 9); *ii)* w/o Temporal Decay, which disables the adaptive temporal decay in short-term injection (Eq. 6); *iii)* w/o Topological Diffusion, which restricts propagation to single-hop neighborhoods (Eq. 10); *iv)* w/o State, which removes the retentive state update (Eqs. 10 and 9). Figure 4 reports the results under both transductive and inductive settings. Across datasets, removing any component degrades performance relative to the full model, with more pronounced drops on datasets exhibiting denser structure or longer temporal spans. Additional ablation results and extended comparisons are provided in Appendix C.12.

5 Related Work

We review dynamic graph learning methods along three dimensions: (i) long-term dependency and memory mechanisms (Appendix D.1), (ii) structural propagation and high-order dependencies (Appendix D.2), and (iii) simplification, scalability, and generalization gaps (Appendix D.3).

6 Conclusion

In this paper, we propose a dual-scale retentive dynamics framework that couples temporal memory and structural diffusion within a unified state representation. By introducing learnable decay kernels and gated memory updates, DSRD adaptively balances short-term responsiveness and long-term retention. Experiments on 14 benchmarks demonstrate state-of-the-art performance across diverse temporal regimes and evaluation settings. We discuss limitations and future directions in Appendix E.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods, specifically for code completion, refactoring, and debugging during the development of our model and experimental pipeline, and for writing assistance, namely polishing and proofreading the manuscript for clarity and grammar. We have not used generative AI tools to develop theoretical models, to formulate mathematical claims, to assist in writing the proofs, to propose or refine hypotheses, to design our research methodology or experiments, or to interpret results. The remaining required-disclosure categories, such as generating or cleaning datasets and translation, are not applicable to this work, which relies on established public benchmarks. We have reviewed all AI-assisted work: all AI-assisted code was verified and tested for correctness by the authors, and all AI-edited text was checked and revised by the authors. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Nelson ARA Passos, Emanuele Carlini, and Salvatore Trani. Deep community detection in attributed temporal graphs: Experimental evaluation of current approaches. In *Proceedings of the 3rd GNNet Workshop on Graph Neural Networking Workshop*, pp. 1–6, 2024.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Qian Chang, Xia Li, Xiufeng Cheng, Runsong Jia, Jinqing Yang, Guoping Hu, and Ciprian Doru Giurcaneanu. Graph retention networks for dynamic graphs. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 511–522, New York, NY, USA, 2026. Association for Computing Machinery. doi: 10.1145/3774904.3792107.

Hsing-Huan Chung, Shravan Chaudhari, Xing Han, Yoav Wald, Suchi Saria, and Joydeep Ghosh. Between linear and sinusoidal: Rethinking the time encoder in dynamic graph learning. *arXiv preprint arXiv:2504.08129*, 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.

Filip Cornell, Oleg Smirnov, Gabriela Zarzar Gandler, and Lele Cao. On the power of heuristics in temporal graphs. *arXiv preprint arXiv:2502.04910*, 2025.

Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11065–11074, 2019.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Songgaojun Deng, Huzefa Rangwala, and Yue Ning. Learning dynamic context graphs for predicting social events. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1007–1016, 2019.

Soheila Farokhi, Xiaojun Qi, and Hamid Karimi. Tawrmac: A novel dynamic graph representation learning method. In *Proceedings of the ACM Web Conference 2026*, pp. 684–695, 2026.

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*, 2025.

Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 317–326, 2016.

Chang Gong, Chuzhe Zhang, Di Yao, Jingping Bi, Wenbin Li, and Yongjun Xu. Causal discovery from temporal data: An overview and new perspectives. *ACM Computing Surveys*, 57(4):1–38, 2024.

Alessio Gravina and Davide Bacciu. Deep learning for dynamic graphs: models and benchmarks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11788–11801, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.

Di Jin, Yingli Gong, Zhiqiang Wang, Zhizhi Yu, Dongxiao He, Yuxiao Huang, and Wenjun Wang. Graph neural network for higher-order dependency networks. In *Proceedings of the ACM Web Conference 2022*, pp. 1622–1630, 2022.

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.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *Advances in Neural Information Processing Systems*, 37:127030–127058, 2024.

Rui Li, Fan Zhang, Tong Li, Ning Zhang, and Tingting Zhang. Dmgan: Dynamic multi-hop graph attention network for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9088–9101, 2022.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 1449–1457, 2015.

Meng Liu, Ke Liang, Yawei Zhao, Wenxuan Tu, Sihang Zhou, Xinbiao Gan, Xinwang Liu, and Kunlun He. Self-supervised temporal graph learning with temporal and structural intensity alignment. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6355–6367, 2024.

Moshe Looks, Marcello Herreshoff, DeLesley Hutchins, and Peter Norvig. Deep learning with dynamic computation graphs. *arXiv preprint arXiv:1702.02181*, 2017.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. *Advances in Neural Information Processing Systems*, 37:141153–141182, 2024.

Yuhong Luo and Pan Li. Neighborhood-aware scalable temporal network representation learning. In *Learning on Graphs Conference*. PMLR, 2022.

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *Companion proceedings of the web conference 2018*, pp. 969–976, 2018.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Bo Peng, Ruichong Zhang, Daniel Goldstein, Eric Alcaide, Xingjian Du, Haowen Hou, Jiaju Lin, Jiaxing Liu, Janna Lu, William Merrill, et al. Rwkv-7" goose" with expressive dynamic state evolution. *arXiv preprint arXiv:2503.14456*, 2025.

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.

Amauri Souza, Diego Mesquita, Samuel Kaski, and Vikas Garg. Provably expressive temporal graph networks. *Advances in neural information processing systems*, 35:32257–32269, 2022.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

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

Aniq Ur Rahman, Ahmed A Elhag, and Justin P Coon. A primer on temporal graph learning. *ACM Computing Surveys*, 58(5):1–28, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Przemysław Andrzej Wałega and Michael Rawson. Expressive power of temporal message passing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21000–21008, 2025.

Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan W Liew, Shirui Pan, and Baocai Yin. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Advances in Neural Information Processing Systems*, 37:8384–8410, 2024.

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.

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.

Zehong Wang, Qi Li, and Donghua Yu. Tpgnn: Learning high-order information in dynamic graphs via temporal propagation. *arXiv preprint arXiv:2210.01171*, 2022.

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

He Yu and Jing Liu. Community-aware temporal walks: Parameter-free representation learning on continuous-time dynamic graphs. *arXiv preprint arXiv:2501.11880*, 2025.

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.

Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.

Appendix

Table of Contents

A Proofs **14**

A.1 Proof of Proposition 2.1 14

A.2 Proof of Theorem 3.1 16

A.3 Walk Expansion of the Operational Update 18

B Additional Descriptions of Methodology **19**

B.1 Dual-Scale Retentive Diffusion Algorithm 19

B.2 Topological Diffusion via Temporal Walks 20

C Additional Experimental Descriptions and Results **21**

C.1 Datasets 21

C.2 Baselines 22

C.3 Detailed Configurations and Hyperparameters 23

C.4 Additional Results of Transductive Dynamic Link Prediction 24

C.5 Experimental Results of Inductive Dynamic Link Prediction 25

C.6 Statistical Significance of Average Ranking Comparisons 26

C.7 Additional Description of Dynamic Node Classification 27

C.8 A Closer Look at Evaluation Regimes in Temporal Link Prediction 28

C.9 Detailed Analysis of Adaptive Decay Behaviors 29

C.10 Complexity Analysis 30

C.11 Scalability 30

C.12 Additional Results of Ablation Study 31

D Additional Related Work **31**

D.1 Long-Term Dependency and Memory Mechanisms 31

D.2 Structural Propagation and High-Order Dependencies 32

D.3 Simplification, Scalability, and Generalization Gaps 33

E Limitations and Future Prospects **33**

A PROOFS

A.1 PROOF OF PROPOSITION 2.1

We first establish a key lemma characterizing the walk diffusion kernel.

Lemma A.1. *Let $\mathcal{T}_t$ denote the ordered set of event timestamps up to t. We adopt the convention that the entries of the set $\mathcal{T}_t$ are $\tau_1 < \tau_2 < \cdots < \tau_n$. For $\ell, t > 0$, we have:*

$$A_t^{(\ell)} = \begin{cases} \displaystyle\sum_{\substack{\tau_1,\ldots,\tau_\ell \in \mathcal{T}_t \\ \tau_1 < \tau_2 < \cdots < \tau_\ell}} T_{\tau_1} T_{\tau_2} \cdots T_{\tau_\ell}, & \text{if } |\mathcal{T}_t| \geq \ell, \\ 0, & \text{otherwise}. \end{cases} \tag{15}$$

Proof of Lemma A.1. For ease of calculation, we first show that the following identity holds for all $\ell, t > 0$:

$$A_t^{(\ell)} = \sum_{\tau \in \mathcal{T}_t} A_{\tau^-}^{(\ell-1)} T_\tau. \tag{16}$$

Base case: $t = \tau_1$. For $\ell = 1$, from the recursion in Equation (3), we have

$$A_{\tau_1}^{(1)} = A_0^{(1)} + A_0^{(0)} T_{\tau_1} = 0 + I \cdot T_{\tau_1} = T_{\tau_1}. \tag{17}$$

Similarly, the right-hand side of Equation (16) gives:

$$\sum_{\tau \in \mathcal{T}_{\tau_1}} A_{\tau^-}^{(0)} T_\tau = A_0^{(0)} T_{\tau_1} = T_{\tau_1}, \tag{18}$$

which confirms that Equation (16) holds for $\ell = 1$ at time τ_1.

For any $\ell \geq 2$, we have

$$A_{\tau_1}^{(\ell)} = A_0^{(\ell)} + A_0^{(\ell-1)} T_{\tau_1} = 0 + 0 \cdot T_{\tau_1} = 0, \tag{19}$$

and

$$\sum_{\tau \in \mathcal{T}_{\tau_1}} A_{\tau^-}^{(\ell-1)} T_\tau = A_0^{(\ell-1)} T_{\tau_1} = 0, \tag{20}$$

confirming Equation (16) for all $\ell \geq 1$ at time τ_1.

Assume that Equation (16) holds for a fixed ℓ at time τ_{q-1} where $q \in \{2, \ldots, n\}$. We demonstrate that it also holds at time τ_q.

Using the recursion in Equation (3), we have

$$\begin{aligned} A_{\tau_q}^{(\ell)} &= A_{\tau_{q-1}}^{(\ell)} + A_{\tau_{q-1}}^{(\ell-1)} T_{\tau_q} \\ &= \sum_{\tau \in \mathcal{T}_{\tau_{q-1}}} A_{\tau^-}^{(\ell-1)} T_\tau + A_{\tau_{q-1}}^{(\ell-1)} T_{\tau_q} \\ &= \sum_{\tau \in \mathcal{T}_{\tau_q}} A_{\tau^-}^{(\ell-1)} T_\tau, \end{aligned}$$

which concludes the proof of Equation (16).

Now we prove the walk expansion (15) using Equation (16).

Base case: $\ell = 1$. From Equation (16), we have

$$A_t^{(1)} = \sum_{\tau \in \mathcal{T}_t} A_{\tau^-}^{(0)} T_\tau = \sum_{\tau \in \mathcal{T}_t} T_\tau, \quad \text{for } t > 0, \tag{21}$$

which confirms Equation (15) for $\ell = 1$.

Assume that Equation (15) holds for $\ell - 1$ at all time moments $\tau_1, \ldots, \tau_n$. We show it also holds for ℓ.

For the case $1 \le q \le \ell - 1$ (i.e., $|\mathcal{T}_{\tau_q}| < \ell$), from Equation (16) we get

$$A_{\tau_q}^{(\ell)} = \sum_{\tau \in \mathcal{T}_{\tau_q}} A_{\tau^-}^{(\ell-1)} T_\tau = 0, \tag{22}$$

since $A_{\tau^-}^{(\ell-1)} = 0$ whenever $|\mathcal{T}_{\tau^-}| < \ell - 1$.

For $\ell \le q \le n$, applying Equation (16) and the induction hypothesis yields

$$\begin{aligned} A_{\tau_q}^{(\ell)} &= \sum_{\tau \in \mathcal{T}_{\tau_q}} A_{\tau^-}^{(\ell-1)} T_\tau \\ &= A_{\tau_{q-\ell+1}^-}^{(\ell-1)} T_{\tau_{q-\ell+1}} + \cdots + A_{\tau_{q-1}}^{(\ell-1)} T_{\tau_q} \\ &= \sum_{\substack{r_1,\ldots,r_\ell \in \{\tau_1,\ldots,\tau_q\} \\ r_1 < r_2 < \cdots < r_\ell}} T_{r_1} T_{r_2} \cdots T_{r_\ell}. \end{aligned} \tag{23}$$

This concludes the proof of the lemma. □

Proof. We first note that $A_t^{(\ell)}(i,j)$ denotes a scalar diffusion score, while Δ_t^i is a matrix-valued increment, hence $\sum_{i \in \mathcal{V}} A_t^{(\ell)}(i,j)\, \Delta_t^i \in \mathbb{R}^{d \times d}$ in Equation (4) is well-defined as a weighted sum of matrices. We prove the identity in Equation (5) by mathematical induction.

Base case. Let $t = \tau_1$, where $\tau_1 > 0$ is the first event timestamp. It follows that $\mathcal{T}_t = \{\tau_1\}$ and $\mathcal{T}_{(\tau_1,t]} = \emptyset$.

Using the recursion in Equation (4), we have for all j and ℓ that

$$S_t^{j,(\ell)} = a_t \cdot 0 + b_t \bar{\Delta}_t^{j,(\ell)} = \sum_{\tau \in \mathcal{T}_t} \left(\prod_{r \in \mathcal{T}_{(\tau,t]}} a_r \right) b_\tau \bar{\Delta}_\tau^{j,(\ell)}, \tag{24}$$

where the summation contains a single term corresponding to $\tau = \tau_1$, and we adopt the convention that $\prod_{r \in \mathcal{T}_{(\tau_1,t]}} a_r = 1$ whenever $\mathcal{T}_{(\tau_1,t]} = \emptyset$.

Assume that the identity in Equation (5) holds for t^- and demonstrate that it also holds for t:

$$\begin{aligned} S_t^{j,(\ell)} &= a_t \left[\sum_{\tau \in \mathcal{T}_{t^-}} \left(\prod_{r \in \mathcal{T}_{(\tau,t^-]}} a_r \right) b_\tau \bar{\Delta}_\tau^{j,(\ell)} \right] + b_t \cdot \bar{\Delta}_t^{j,(\ell)} \\ &= \sum_{\tau \in \mathcal{T}_{t^-}} \left(\prod_{r \in \mathcal{T}_{(\tau,t]}} a_r \right) b_\tau \cdot \bar{\Delta}_\tau^{j,(\ell)} + \sum_{\tau = t} \underbrace{\left(\prod_{r \in \mathcal{T}_{(t,t]}} a_r \right)}_{=1} b_\tau \cdot \bar{\Delta}_\tau^{j,(\ell)} \\ &= \sum_{\tau \in \mathcal{T}_t} \left(\prod_{r \in \mathcal{T}_{(\tau,t]}} a_r \right) b_\tau \cdot \bar{\Delta}_\tau^{j,(\ell)}, \end{aligned} \tag{25}$$

which concludes the proof of Equation (5). □

Remark A.2. It remains to justify that $A_t^{(\ell)}$ in Equation (3) indeed represents time-respecting diffusion scores of increasing walk length. By Lemma A.1, for $\ell, t > 0$ with $|\mathcal{T}_t| \ge \ell$:

$$A_t^{(\ell)} = \sum_{\substack{\tau_1,\ldots,\tau_\ell \in \mathcal{T}_t \\ \tau_1 < \tau_2 < \cdots < \tau_\ell}} T_{\tau_1} T_{\tau_2} \cdots T_{\tau_\ell}, \tag{26}$$

where the constraint $\tau_1 < \tau_2 < \cdots < \tau_\ell$ arises because each recursive step uses $A_{\tau^-}^{(\ell-1)}$, ensuring that the $(\ell-1)$-hop prefix is computed strictly before time τ_ℓ.

Since each T_τ is an instantaneous transition matrix induced by events at time τ, the matrix product $T_{\tau_1} \cdots T_{\tau_\ell}$ sums over all length-ℓ paths whose step times satisfy $\tau_1 < \cdots < \tau_\ell$. Therefore, each entry $A_t^{(\ell)}(i,j)$ equals the total count (or weighted score, if T_τ entries are weighted) of all time-respecting walks from i to j of length ℓ up to time t. Combining Equation (24) with Equation (26), $S_t^{j,(\ell)}$ becomes a temporally retentive accumulation of matrix-valued increments Δ_τ^i whose influence is propagated to j through length-ℓ time-respecting walks, and the readout O_t^j performs the local contraction with Q_t^j.

The increment Δ_t^i aggregates rank-one outer products $(K_{t_e}^i)^\top V_{t_e}^i$, a standard second-order feature construction used to capture multiplicative channel interactions (Lin et al., 2015; Gao et al., 2016; Dai et al., 2019; Peng et al., 2023; 2025; Sun et al., 2023; Gu & Dao, 2024). Proposition 2.1 shows that once these second-order increments are injected into the graph and propagated via time-respecting walk scores, the resulting state simultaneously admits a closed-form historical aggregation over time and an explicit high-order structural diffusion over increasing walk length. Notably, the recursion Equation (3) with unit coefficients ($a_t = b_t = 1$) recovers exact temporal walk counts, while the state dynamics in (4) with learnable (a_t, b_t) introduce adaptive temporal weighting that balances recent versus historical information.

A.2 Proof of Theorem 3.1

We recall that the operational short-term injection $\tilde{\Delta}_t^{j,(\ell)}$ (Equation (9)) is uniformly bounded under standard architectural constraints:

- **Bounded input features:** Node features are typically normalized (e.g., layer normalization or batch normalization), ensuring $\|x_v\| \le B_{\text{node}}$, and the augmented temporal-edge features $\tilde{\phi}_e$ in Equation (12) are bounded by B_{edge} since they are produced by bounded time encodings and edge attributes. We write $B_x = B_{\text{node}} + B_{\text{edge}}$ for the resulting bound on the augmented projection inputs $x_v + \tilde{\phi}_e$.
- **Bounded projection matrices:** We assume the projection matrices have bounded norm, $\|W_K\|, \|W_V\| \le L$, which is encouraged in practice by weight decay regularization during training.
- **Normalized attention weights:** The temporal attention weights $\tilde{\omega}_{ij}(t_e)$ are normalized over each target neighborhood as described in Section 3.2, ensuring $\sum_{(i,j,t_e)\in\mathcal{E}_t(j)} \tilde{\omega}_{ij}(t_e) \le 1$.

Lemma A.3. *Assume the augmented projection inputs are bounded ($\|x_v + \tilde{\phi}_e\| \le B_x$) and projection matrices are bounded ($\|W_K\|, \|W_V\| \le L$). With the normalized temporal attention weights $\tilde{\omega}_{ij}(t_e)$ defined in Section 3.2, the operational short-term injection satisfies*

$$\|\tilde{\Delta}_t^{j,(\ell)}\| \le L^2 B_x^2 \tag{27}$$

for all t, j, and ℓ.

Proof. By definition, $\tilde{\Delta}_t^{j,(\ell)} = \sum_{(i,j,t_e)\in\mathcal{E}_t} \tilde{\omega}_{ij}(t_e)\,(\tilde{K}_{t_e}^i)^\top \tilde{V}_{t_e}^i$, where $\tilde{K}_{t_e}^i = (x_i + \tilde{\phi}_e)W_K$ and $\tilde{V}_{t_e}^i = (x_i + \tilde{\phi}_e)W_V$ are the augmented projections of Equation (12). Hence $\|\tilde{K}_{t_e}^i\|, \|\tilde{V}_{t_e}^i\| \le B_x L$, so $\|(\tilde{K}_{t_e}^i)^\top \tilde{V}_{t_e}^i\| \le L^2 B_x^2$. Since $\tilde{\omega}_{ij}(t_e) \ge 0$ and $\sum_{(i,j,t_e)\in\mathcal{E}_t(j)} \tilde{\omega}_{ij}(t_e) \le 1$ by normalization, the triangle inequality gives:

$$\|\tilde{\Delta}_t^{j,(\ell)}\| \le \sum_{(i,j,t_e)\in\mathcal{E}_t(j)} \tilde{\omega}_{ij}(t_e)\,\|(\tilde{K}_{t_e}^i)^\top \tilde{V}_{t_e}^i\| \le L^2 B_x^2. \tag{28}$$

□

The operational gated update, Equation (9), has the form $S_t^{j,(\ell)} = \gamma^{(\ell)} S_{t^-}^{j,(\ell)} + (1-\gamma^{(\ell)})\tilde{\Delta}_t^{j,(\ell)}$, where the short-term injection satisfies $\|\tilde{\Delta}_t^{j,(\ell)}\| \le M$ by Lemma A.3, with $M = L^2 B_x^2$. The normalized structural weights satisfy $\tilde{\psi}_{ij}^{(\ell)}(t_e) \ge 0$ and $\sum_i \tilde{\psi}_{ij}^{(\ell)}(t_e) \le 1$, so the interlayer aggregation is non-expansive. Whenever $\|S_t^{i,(\ell)}\| \le M_\ell$ for all source nodes i, the triangle inequality gives $\big\|\sum_{(i,j,t_e)\in\mathcal{E}_t(j)} \tilde{\psi}_{ij}^{(\ell)}(t_e)\, S_t^{i,(\ell)}\big\| \le M_\ell$. We now prove Theorem 3.1.

Proof. **Part (i): state bound.** Consider first a layer driven by the gated recurrence alone, as for layer K, which is never a propagation target. Writing $s_k = \|S^{j,(\ell)}_{\tau_k}\|$, the recurrence and the bounded injection give $s_k \le \gamma^{(\ell)} s_{k-1} + (1-\gamma^{(\ell)})M$, and induction on the event index from $s_0 = 0$ yields $s_n \le \big(1-(\gamma^{(\ell)})^n\big)M \le M$. In particular $\|S^{j,(K)}_t\| \le M$.

For the full update we track the layer-wise bounds over events. The two per-event operations act on different axes. The gated update treats each layer separately in time and does not couple layers, whereas the interlayer propagation is the only step that couples them, carrying each layer into the one directly below it in decreasing order $\ell = K, \dots, 2$. We index the events in increasing time order $\tau_1 < \tau_2 < \cdots$, so event t occurs at time τ_t and event $t-1$ immediately precedes it. Let $F_{t,\ell}$ be a uniform bound over all nodes j on $\|S^{j,(\ell)}\|$ right after the gated update of event t (before its propagation), and $C_{t,\ell}$ the bound right after the full update of event t. Because the state is unchanged between events, the gated update of event t acts on $S^{j,(\ell)}_{(\tau_t)^-} = S^{j,(\ell)}_{\tau_{t-1}}$, the state left by event $t-1$, so its bound is $C_{t-1,\ell}$, giving the per-layer temporal bound

$$F_{t,\ell} = \gamma^{(\ell)} C_{t-1,\ell} + (1-\gamma^{(\ell)})M. \tag{29}$$

At the first event the empty initial state $S_0 = 0$ gives $\|S^{j,(\ell)}\| \le M$, so $F_{1,\ell} = M$. The propagation of event t sends each layer into the one directly below. Since it is processed in decreasing order $\ell = K, \dots, 2$, when layer $\ell - 1$ receives from layer ℓ the latter has already been updated, so by the non-expansiveness above the after-update bound of layer $\ell - 1$ is its post-gated value $F_{t,\ell-1}$ plus the after-update bound $C_{t,\ell}$ of the layer above,

$$C_{t,\ell-1} = F_{t,\ell-1} + C_{t,\ell}. \tag{30}$$

Unrolling this relation from the top, where $C_{t,K} = F_{t,K} = M$ because layer K is never a propagation target, gives the suffix sum

$$C_{t,\ell} = \sum_{q=\ell}^{K} F_{t,q}. \tag{31}$$

With the nested sums $\mathscr{S}_0(\ell) = K - \ell$ and $\mathscr{S}_m(\ell) = \sum_{q=\ell}^{K} \gamma^{(q)} \mathscr{S}_{m-1}(q)$ for $m \ge 1$, a direct induction on t solves both recurrences. At $t = 1$, $F_{1,\ell} = M$ and $C_{1,\ell} = (K-\ell+1)M = M[1 + \mathscr{S}_0(\ell)]$. For the inductive step (induction on n), assume the after-update form holds at $n-1$ for *every* layer, $C_{n-1,q} = M[1 + \sum_{m=0}^{n-2} \mathscr{S}_m(q)]$. The gated update then gives, for each $q \ge \ell$,

$$F_{n,q} = \gamma^{(q)} C_{n-1,q} + (1-\gamma^{(q)})M = M\Big[1 + \gamma^{(q)} \sum_{m=0}^{n-2} \mathscr{S}_m(q)\Big], \tag{32}$$

where the $\gamma^{(q)}$ and $(1-\gamma^{(q)})$ terms recombine to leave the leading 1. Summing over $q = \ell, \dots, K$, exchanging the order of the two sums, and applying the defining recursion $\sum_{q=\ell}^{K} \gamma^{(q)} \mathscr{S}_m(q) = \mathscr{S}_{m+1}(\ell)$,

$$\begin{aligned} C_{n,\ell} = \sum_{q=\ell}^{K} F_{n,q} &= M\Big[\sum_{q=\ell}^{K} 1 + \sum_{m=0}^{n-2} \sum_{q=\ell}^{K} \gamma^{(q)} \mathscr{S}_m(q)\Big] = M\Big[(K-\ell+1) + \sum_{m=0}^{n-2} \mathscr{S}_{m+1}(\ell)\Big] \\ &= M\Big[(K-\ell+1) + \sum_{m=1}^{n-1} \mathscr{S}_m(\ell)\Big] = M\Big[1 + \sum_{m=0}^{n-1} \mathscr{S}_m(\ell)\Big], \end{aligned} \tag{33}$$

using $K - \ell + 1 = 1 + \mathscr{S}_0(\ell)$ in the last step. This is the claimed after-update form at n, and the before-update form follows the same way. In summary,

$$F_{n,\ell} = M\Big[1 + \gamma^{(\ell)} \sum_{m=0}^{n-2} \mathscr{S}_m(\ell)\Big], \qquad C_{n,\ell} = M\Big[1 + \sum_{m=0}^{n-1} \mathscr{S}_m(\ell)\Big]. \tag{34}$$

The series $\sum_{m \ge 0} \mathscr{S}_m(\ell)$ converges by scalar comparison. Unrolling the recursion m times writes $\mathscr{S}_m(\ell)$ as a nonnegative sum over weakly increasing length-m layer-index sequences starting at ℓ, each contributing a product of m retention factors times a weight at most $K - \ell$. Sequences

that reach layer K contribute zero, so the indices stay within the $K-\ell$ values $\ell, \dots, K-1$. With $\bar{\gamma} = \max_{\ell \le q \le K-1} \gamma^{(q)} < 1$ each such product is at most $\bar{\gamma}^{\,m}$, and there are $\binom{m+K-\ell-1}{K-\ell-1}$ such sequences, so each term obeys $\mathscr{S}_m(\ell) \le (K-\ell)\binom{m+K-\ell-1}{K-\ell-1}\bar{\gamma}^{\,m}$. The majorant converges by the ratio test, since its successive ratio $\bar{\gamma}\,(m+K-\ell)/(m+1) \to \bar{\gamma} < 1$. Hence $C_{t,\ell}$ is nondecreasing and bounded in t, so $\|S_t^{j,(\ell)}\| \le C_{t,\ell} \le B_\ell = \lim_{t\to\infty} C_{t,\ell}$ for every event t. Letting $t \to \infty$ in the two updates gives $B_{\ell-1} = F_{\ell-1} + B_\ell$ with $F_{\ell-1} = \gamma^{(\ell-1)} B_{\ell-1} + (1-\gamma^{(\ell-1)})M$. Solving for $B_{\ell-1}$,

$$B_{\ell-1} = M + \frac{B_\ell}{1-\gamma^{(\ell-1)}}, \qquad B_K = M, \tag{35}$$

which are exactly the constants in the statement. This proves part (i).

Part (ii): readout bound. By part (i), $\|S_t^{j,(\ell)}\| \le B_\ell$. Combined with $\|Q_t^{j,(\ell)}\| \le LB_x$,

$$\|O_t^{j,(\ell)}\| = \|Q_t^{j,(\ell)} S_t^{j,(\ell)}\| \le \|Q_t^{j,(\ell)}\|\,\|S_t^{j,(\ell)}\| \le LB_x\,B_\ell, \tag{36}$$

which completes the proof. □

A.3 Walk Expansion of the Operational Update

We now characterize exactly what the operational updates in Equations (9) and (10) compute, thereby making precise the sense in which they implicitly encode temporal walk aggregation. We assume distinct event timestamps (events sharing a timestamp are discussed in Remark A.5). Following Algorithm 1, each arriving event (u, v, τ) is processed in two steps: the gated update (9) is applied at all layers, where the injection $\tilde{\Delta}_\tau^{j,(\ell)}$ vanishes for every node without an arriving edge, and the interlayer propagation (10) is then applied for the arriving edge at layers $\ell = K, \dots, 2$ in descending order.

Proposition A.4. *Assume distinct event timestamps. For every node j, layer $\ell \in \{1, \dots, K\}$, and time $t \ge 0$, the state produced by Equations* (9) *and* (10) *admits the closed-form expansion*

$$\begin{aligned} S_t^{j,(\ell)} = \sum_{w=1}^{K-\ell+1} \sum_{\substack{\tau_1 < \cdots < \tau_w \le t \\ v_w = j}} \big(1-\gamma^{(\ell+w-1)}\big) \prod_{r \in \mathcal{T}_{(\tau_w, t]}} \gamma^{(\ell)} \\ \times \prod_{k=2}^{w} \left[\psi_{v_{k-1} v_k}^{(\ell+w-k+1)}(\tau_k) \prod_{r \in \mathcal{T}_{(\tau_{k-1}, \tau_k]}} \gamma^{(\ell+w-k+1)} \right] \tilde{\Delta}_{\tau_1}^{v_1,(\ell+w-1)}, \end{aligned} \tag{37}$$

where w denotes the walk length and the inner sum ranges over all injection times $\tau_1 \in \mathcal{T}_t$, nodes $v_1 \in \mathcal{V}$, and time-respecting edge sequences $(v_{k-1}, v_k, \tau_k) \in \mathcal{E}_t$ for $k = 2, \dots, w$. The same expansion holds verbatim with the normalized weights $\tilde{\psi}$ of Section 3.2 in place of ψ.

Proof. We proceed by induction on the events, ordered by timestamp. Before the first event, $S_0^{j,(\ell)} = 0$ for all j and ℓ, and the right-hand side of Equation (37) is an empty sum, so the claim holds. Suppose the expansion holds for all nodes and layers immediately before an event (u, v, τ). We track how each processing step transforms the expansion.

(1) *Gated update.* At every layer, Equation (9) gives $S_\tau^{j,(\ell)} = \gamma^{(\ell)} S_{\tau^-}^{j,(\ell)} + (1-\gamma^{(\ell)})\,\tilde{\Delta}_\tau^{j,(\ell)}$. The factor $\gamma^{(\ell)}$ appends the new event time τ to the trailing retention product $\prod_{r \in \mathcal{T}_{(\tau_w, t]}} \gamma^{(\ell)}$ of every existing term, which matches the expansion evaluated at $t = \tau$. The injection contributes the terms with $w = 1$ and $\tau_1 = \tau$, whose weight $(1-\gamma^{(\ell)})$ and empty retention products agree with Equation (37).

(2) *Interlayer propagation.* For $\ell = K, \dots, 2$ in descending order, Equation (10) adds $\psi_{uv}^{(\ell)}(\tau)\, S_\tau^{u,(\ell)}$ to $S_\tau^{v,(\ell-1)}$. At this point, $S_\tau^{u,(\ell)}$ has received its decay at τ but no propagation, since the arriving edge targets v, so node u is only read, never written, while processing this event. By the induction hypothesis, its expansion consists of terms of length $w - 1 \le K - \ell + 1$ ending at $v_{w-1} = u$ with $\tau_{w-1} < \tau$ and trailing retention product over $\mathcal{T}_{(\tau_{w-1}, \tau]}$. Multiplying such a term by $\psi_{uv}^{(\ell)}(\tau)$ and appending the hop to v at time $\tau_w = \tau$ yields precisely the length-w term ending at $j = v$ in the

expansion at layer $\ell-1$: the injection layer satisfies $\ell+(w-1)-1=(\ell-1)+w-1=\ell+w-2$, the trailing product over $\mathcal{T}_{(\tau_{w-1},\tau]}$ becomes the interior retention factor of the new final hop, and the new trailing product over $\mathcal{T}_{(\tau_w,\tau]}$ is empty, reflecting that the propagation is applied after the decay at τ. Conversely, every length-w term with $\tau_w=\tau$ and $v_w=v$ arises in this way. Because each layer receives propagated content only from the layer above, and that content was written at strictly earlier events, no term acquires more than one additional hop per event and all timestamps remain strictly increasing.

Both steps preserve the claimed expansion, which completes the induction. □

Note that the terms with $w=1$ recover exactly the closed form of Proposition 2.1 with $a_t=\gamma^{(\ell)}$ and $b_t=1-\gamma^{(\ell)}$, while the terms with $w\geq 2$ are the contributions added by topological diffusion.

Remark A.5. Proposition A.4 establishes the precise relationship between the operational update and the idealized walk aggregation of Equations (3) and (4). Each injection $\tilde{\Delta}_{\tau_1}^{v_1,(\ell+w-1)}$ is routed to node j along every sequence of edges with strictly increasing timestamps in $(\tau_1,t]$, which is precisely the family of temporal walks aggregated by $A_t^{(K-\ell+1)}$ in Lemma A.1. The operational update therefore performs the routing of the walk matrix without materializing $A_t^{(K-\ell+1)}$. The two formulations differ only in the weight assigned to each walk: the idealized model accumulates uniform walk weights, whereas the operational model applies the learned weight in Equation (37), composed of the structural attenuation ψ at each hop, the retention factor γ at each layer, and a single injection gate. Setting $\psi\equiv 1$ and removing the gating (taking $a_t=b_t=1$ in Equation (4)) reduces every weight to one and recovers the unweighted walk counts of Lemma A.1. The operational update additionally enforces a stronger form of causality at the level of individual edges: each injection travels only along edges occurring strictly after its injection time, whereas the factored form $\sum_i A_t^{(K-\ell+1)}(i,j)\,\Delta_t^i$ decouples routing times from emission times.

For events sharing a timestamp, which are processed as a simultaneous batch (see Appendix C.3), the expansion holds with the first inequality relaxed to $\tau_1\leq\tau_2$: an injection may traverse at most one hop within its own batch, and all subsequent hops occur at strictly later timestamps, since the descending layer order of Equation (10) advances each state by at most one layer per propagation pass.

B Additional Descriptions of Methodology

B.1 Dual-Scale Retentive Diffusion Algorithm

Algorithm 1 summarizes the per-event processing of DSRD, and Figure 5 provides the corresponding overview of DSRD architecture.

Algorithm 1 Dual-Scale Retentive Diffusion for DSRD

Input: Dynamic subgraph $\mathcal{G}_t=(\mathcal{V},\mathcal{E}_t)$, walk depth K.
Initialize retentive states $S_0^{j,(\ell)}$ for all $j\in\mathcal{V}$, $\ell\in\{1,\ldots,K\}$.
for each event $(i,j,t_e)\in\mathcal{E}_t$ **do**
  for $\ell=1$ **to** K **do**
    Compute projections via Eq. (1).
    Compute short-term injection $\tilde{\Delta}_t^{j,(\ell)}$ via Eq. (8).
    Fuse short-term and long-term signals via Eq. (9).
  end for
  Obtain output $O_t^{j,(K)}=Q_t^j S_t^{j,(K)}$.
  for $\ell=K$ **down to** 2 **do**
    Update topological diffusion via Eq. (10).
  end for
end for
Output: Final node representations $\{O_t^{j,(K)}\}_{j\in\mathcal{V}}$

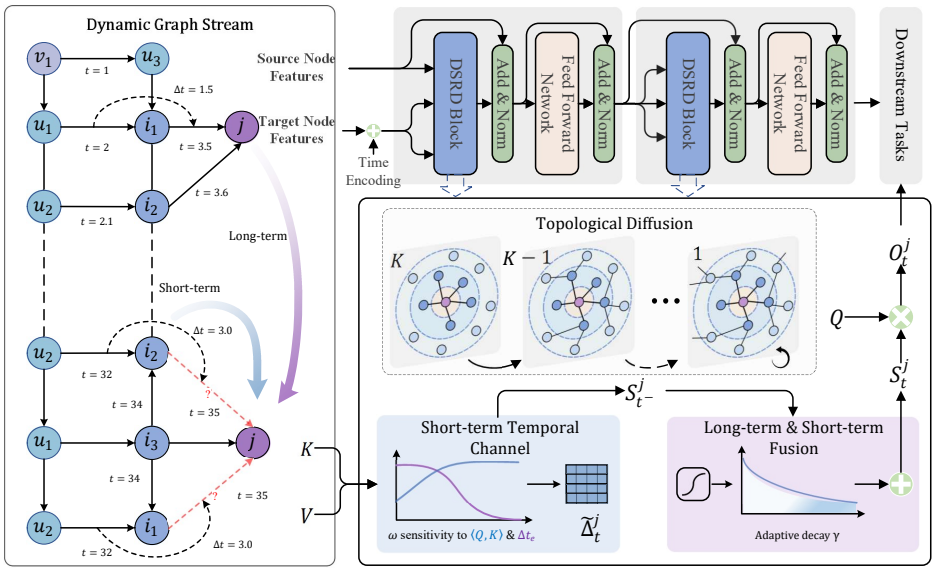
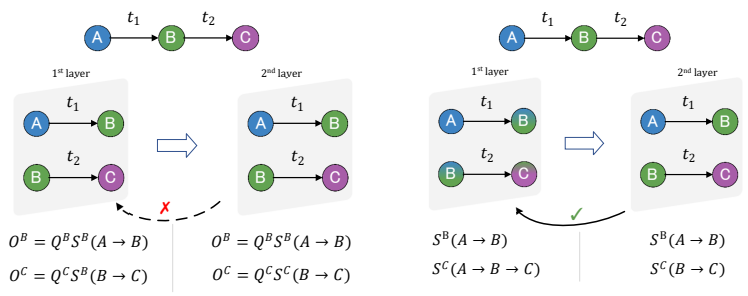


Figure 5: Overview of DSRD. Left: i, u, v are 1-hop, 2-hop, and 3-hop temporal neighbors. Lower-right: the three core operations (short-term injection, topological diffusion, and gated long/short-term fusion). Upper-right: stacked DSRD layers with feed-forward sublayers.

B.2 Topological Diffusion via Temporal Walks



(a) w/o layer-wise propagation (b) w/ layer-wise propagation

Figure 6: An illustration of topological diffusion via temporal walks.

Additional Descriptions for Topological Diffusion via Temporal Walks. In Figure 6, we provide an illustration of a minimal temporal chain $A \xrightarrow{t_1} B \xrightarrow{t_2} C$ to explain why topological diffusion in dynamic graphs should be realized through temporally valid walks rather than relying on layer-local retention. We contrast two designs that differ solely in whether the retentive state S is propagated across layers.

Case (a). In Figure 6a, each layer maintains its own retentive state without cross-layer transmission. At the first layer, node B produces its output as $O^B = Q^B S^B(A \to B)$, while node C produces $O^C = Q^C S^B(B \to C)$. Here, the superscript of S indicates that both outputs are computed from layer-local states that only encode one-hop interactions at the corresponding time steps. Although the outputs O^B and O^C are forwarded to the next layer, this transmission occurs exclusively through the QS interaction, which acts as a compressed readout. Consequently, at the second layer, node C can only form $O^C = Q^C S^C(B \to C)$, where the higher-order dependency $A \to B \to C$ has already been discarded. The information originating from $A \to B$ is never preserved in a persistent state accessible to node C, but is instead repeatedly filtered through successive QS projections, leading to severe structural information loss.

Case (b). In Figure 6b, the retentive state S is explicitly propagated across layers. Node B first constructs the state $S^B(A \to B)$, which is then carried forward and extended at the next layer when processing the interaction $B \to C$. As a result, node C maintains $S^C(A \to B \to C)$, which represents a temporally valid walk rather than an isolated one-hop edge. The output is subsequently obtained as $O^C = Q^C S^C(A \to B \to C)$, allowing higher-order temporal dependencies to be preserved without repeated compression.

This comparison shows that, in the absence of layer-wise state propagation, multi-layer retentive architectures collapse into stacks of one-hop retention modules whose effective receptive field remains local. Topological diffusion via temporal walks is therefore essential for maintaining long-range structural information in dynamic graphs, which directly motivates the design adopted in this work.

C Additional Experimental Descriptions and Results

C.1 Datasets

Table 3: Statistics of datasets. Type C denotes continuous-time dynamic graphs in which each interaction is associated with a precise timestamp, and Type D denotes discrete-time dynamic graphs whose interactions are aggregated into coarser temporal windows such as days or years. N.&F. Feat. represents node and link feature dimensions. Graph density is approximated as $|\mathcal{E}|/|\mathcal{V}|^2$ and the scale metric is defined as average links per unique timestep, reflecting the temporal intensity of interactions.

Datasets	Type	Domains	Bipartite	#Nodes	#Links	#N.&L. Feat.	#Unique Steps	Duration	Time Granularity	Density	Scale
Wikipedia	C	Social	True	9227	157474	- & 172	152757	1 month	Unix time	1.85e-03	1.03
Reddit	C	Social	True	10984	672447	- & 172	669065	1 month	Unix time	5.57e-03	1.00
MOOC	C	Interaction	True	7144	411749	- & 4	345600	17 months	Unix time	8.07e-03	1.19
LastFM	C	Interaction	True	1980	1293103	- & -	1283614	1 month	Unix time	3.30e-01	1.01
Myket	C	Interaction	True	17988	694121	- & -	693774	197 days	Unix time	2.15e-03	1.00
Enron	C	Social	False	184	125235	- & -	22632	3 years	Unix time	3.70e+00	5.53
Social Evo.	C	Proximity	False	74	2099519	- & 2	565932	8 months	Unix time	3.83e+02	3.71
UCI	C	Social	False	1899	59835	- & -	58911	196 days	Unix time	1.66e-02	1.01
Flights	D	Transport	False	13169	1927145	- & 1	122	4 months	days	1.11e-02	15796.27
Can. Parl.	D	Politics	False	734	74478	- & 1	14	14 years	years	1.38e-01	5320.00
US Legis.	D	Politics	False	225	60396	- & 1	12	12 congresses	congresses	1.19e+00	5033.00
UN Trade	D	Economics	False	255	507497	- & 1	32	32 years	years	7.80e+00	15859.28
UN Vote	D	Politics	False	201	1035742	- & 1	72	72 years	years	2.56e+01	14385.31
Contact	C	Proximity	False	692	2426279	- & 1	8064	1 month	5 minutes	5.06e+00	300.94

We provide below the sources and details of the datasets[1] used in this work:

- **Wikipedia**: This bipartite CTDG captures a month-long snapshot of editorial activity on Wikipedia, where editors interact with pages through timestamped editing events. Each interaction carries a 172-dimensional LIWC feature vector derived from the edit text, and dynamic labels reflect temporary editing bans imposed on users.
- **Reddit**: This dataset chronicles user participation across Reddit communities during a single month. Users and posts form a bipartite CTDG, and each posting event is time-stamped and accompanied by a 172-dimensional LIWC vector. Posting bans appear as dynamic labels on edges.
- **MOOC**: Student engagement in an online course ecosystem is represented over 17 months, with interactions linking students to videos, quizzes, or problem sets. Each timestamped access event forms an edge enriched with four interaction features, producing a dense bipartite CTDG.
- **LastFM**: User listening behavior over one month is encoded as a CTDG connecting users to songs. Each listening action is treated as a temporal edge. No node or edge features are included, making temporal patterns the primary signal.
- **Myket**: This large-scale bipartite CTDG follows app-download activity in the Myket marketplace over 197 days. Users download apps through timestamped interactions, forming edges that carry no additional features beyond timing and identity.
- **Enron**: Internal email exchanges within Enron Energy over a three-year period are modeled as a CTDG. Employees serve as nodes, and each email forms a timestamped edge; although featureless, the dataset exhibits rich temporal bursts and communication cycles.

[1] https://zenodo.org/records/7213796#.Y1c06y8r30o

- **Social Evo.**: This proximity-based CTDG tracks face-to-face encounters among students in an undergraduate dormitory over eight months. Each interaction is time-stamped and includes two proximity-related features that quantify encounter strength or duration, resulting in a densely connected human-centric temporal network.
- **UCI**: This CTDG captures online communication among nearly two thousand UC Irvine students over 196 days. Edges represent time-stamped message exchanges without additional attributes, highlighting temporal messaging dynamics.
- **Flights**: Air-traffic patterns during the COVID-19 pandemic are represented as a directed DTDG, with airports as nodes and daily flight volumes forming weighted edges. The network spans four months and exhibits strong temporal variation in global mobility.
- **Can. Parl.**: Parliamentary co-voting among Canadian Members of Parliament from 2006 to 2019 produces a political DTDG. Nodes correspond to MPs, and weighted edges track how frequently two MPs agree by voting "yes" on the same bill within a year.
- **US Legis.**: Legislative collaboration in the US Senate is encoded through co-sponsorship behavior. Senators form the nodes of this DTDG, and edge weights record the number of jointly sponsored bills during each congressional session.
- **UN Trade**: This economic DTDG follows agricultural trade flows among 181 countries across 32 years. Directed edges quantify normalized trade values, producing a temporally evolving network dominated by weighted interactions.
- **UN Vote**: Voting alignment in the United Nations General Assembly is represented over more than seven decades. Countries are connected whenever they vote "yes" on the same proposal, with edge weights accumulating agreement counts across time.
- **Contact**: This proximity-based CTDG monitors physical interactions among roughly 700 university students over one month. Timestamped proximity events form edges whose weights capture the duration or intensity of close-range encounters.

C.2 BASELINES

The detailed descriptions of the baselines are provided as follows:

- **JODIE** (Kumar et al., 2019) learns coupled recurrent updates for users and items and predicts future embeddings through a projection module that models continuous temporal trajectories.
- **DyRep** (Trivedi et al., 2019) captures graph evolution by modeling communication and association events with temporal point processes, allowing node embeddings to evolve through event-driven message passing.
- **TGAT** (Xu et al., 2020) applies temporal graph attention with a functional time encoding derived from Bochner's theorem to aggregate temporal–topological information in a continuous-time setting.
- **TGN** (Rossi et al., 2020) maintains a memory vector for each node and updates it through temporal message passing, generating time-conditioned embeddings that reflect long-term and short-term dynamics.
- **CAWN** (Wang et al., 2021b) represents interactions through causal anonymous walks and encodes temporal walk patterns with recurrent architectures, extracting fine-grained causal and temporal dependencies.
- **EdgeBank** (Poursafaei et al., 2022) stores previously observed edges in memory and predicts future interactions by directly querying the edge history, serving as a simple yet competitive memory-based baseline.
- **TCL** (Wang et al., 2021a) introduces a two-stream Transformer that processes the temporal neighborhoods of interacting nodes and applies cross-attention to learn robust temporal representations via contrastive learning.
- **GraphMixer** (Cong et al., 2023) adopts an MLP-Mixer architecture with fixed time encodings and lightweight mixing operations, enabling efficient and stable temporal representation learning without attention or recurrence.

- **NAT** (Luo & Li, 2022) incorporates neighborhood-aware temporal aggregation that combines time features with attention-based structural context, producing expressive time-sensitive embeddings.
- **PINT** (Souza et al., 2022) enhances the expressive power of temporal message passing by enforcing injective aggregation functions and encoding relative temporal positions to distinguish complex interaction patterns.
- **DyGFormer** (Yu et al., 2023) models dynamic graphs using a Transformer architecture with aligned temporal encodings and co-occurrence modeling, capturing long-range temporal and structural dependencies.
- **TPNet** (Lu et al., 2024) introduces a time-decay temporal walk matrix and projects it into node representations through random feature propagation, preserving temporal walk statistics efficiently for link prediction.
- **CTWalks** (Yu & Liu, 2025) performs parameter-free representation learning on continuous-time dynamic graphs through community-aware temporal walks that bias sampling toward locally cohesive neighborhoods.
- **TAWRMAC** (Farokhi et al., 2026) is a recent dynamic graph representation learning method that combines temporal-aware random walks with memory-augmented context for link prediction.
- **GraphHeuristic** (Cornell et al., 2025) is a strong heuristic baseline that predicts interactions from simple temporal and structural statistics, highlighting the competitiveness of non-parametric rules on temporal graphs.
- **GRN** (Chang et al., 2026) is a graph retention network that adapts the retention/state-space paradigm to dynamic graphs with multi-scale temporal decays and a unified memory-aggregation update, serving as a strong recent state-space baseline.

C.3 Detailed Configurations and Hyperparameters

All experiments are conducted on a computing platform running Ubuntu 24.04.3 LTS, equipped with an Intel(R) Xeon(R) Gold 6548Y+ CPU @ 2.50 GHz and two NVIDIA L40S 48Q GPUs, while the software environment consists of Python 3.12, CUDA 13.0, and PyTorch 2.6.0. All models are trained and evaluated under the unified pipeline of the Dynamic Graph Library (DyGLib).[2] Our datasets and evaluation protocols follow the standard settings established by DyGLib and the Temporal Graph Benchmark (TGB),[3] which define the widely adopted datasets and negative sampling strategies for dynamic graph learning. As baselines, we select both classic and current state-of-the-art dynamic graph models spanning memory-based, attention-based, temporal-walk, heuristic, and state-space approaches. Each dataset is split chronologically into 70%/15%/15% for training, validation, and test, and all reported results are averaged over five runs with different random seeds.

Event Processing and Training. We process the event stream in chronological mini-batches, gathering the retentive states of the involved nodes from a per-node cache, updating them with the DSRD block, and writing them back. Gradients are truncated at the batch boundary by detaching the cached state, which amounts to truncated backpropagation through time with truncation length equal to the batch span. In CTDGs, events carry distinct timestamps and are processed in strict chronological order, so the temporal walk semantics in Lemma A.1 hold exactly. In DTDGs, edges sharing a timestamp form one snapshot and are treated as a simultaneous batch: their contributions are aggregated into $\tilde{\Delta}_t^{j,(\ell)}$ via Equation (8) before the state update, so no causal order is imposed within a snapshot, and the interlayer propagation in Equation (10) is applied once per snapshot, keeping the snapshot boundary as the causal boundary. All decay factors are evaluated at the update time t over the sampled temporal neighborhood: both ω and ψ are computed from the elapsed time $\Delta t_e = t - t_e$ of each sampled historical edge, so the newest edge enters with weight close to one while older edges are attenuated, and the structural attenuation ψ is therefore nontrivial whenever historical edges are propagated.

[2] https://github.com/yule-BUAA/DyGLib

[3] https://tgb.complexdatalab.com/

For the proposed DSRD, we adopt a unified set of hyperparameters across all experiments, with minor adjustments depending on the dataset scale. DSRD is configured with two model layers and a channel embedding dimension of 64. A dropout rate of 0.1 is used throughout. The number of short-term temporal neighbors is 20. All other hyperparameters follow the global training configuration, including a learning rate of 10^{-4}, batch size of 200, early stopping with a patience of 10, and the Adam optimizer. For baseline models, we use the configurations and hyperparameters reported by Yu et al. (2023) and Lu et al. (2024), which we verified through independent validation.

C.4 Additional Results of Transductive Dynamic Link Prediction

To complement the main findings presented in Section 4.2, we report the detailed transductive dynamic link prediction results here. The transductive AP across all datasets under random, historical, and inductive negative sampling is reported in Table 4, and the corresponding ROC-AUC scores are reported in Table 5. We further compare DSRD against four recent methods, i.e., CTWalks (Yu & Liu, 2025), TAWRMAC (Farokhi et al., 2026), GraphHeuristic (Cornell et al., 2025), and the state-space model GRN (Chang et al., 2026). As several of these recent baselines incur substantial computational cost on larger graphs, evaluating them across the full benchmark suite is time-consuming. Following their reported protocols, we therefore restrict this comparison to the eight commonly shared, representative datasets under random negative sampling, with transductive AP reported in Table 6. DSRD attains the best average rank, with pronounced margins on discrete-time graphs such as Can. Parl., US Legis., and UN Trade, where competing methods struggle to capture long-range structural dependencies, while CTWalks remains competitive on dense proximity graphs such as UCI.

Table 4: AP for transductive dynamic link prediction with random (rnd), historical (hist), and inductive (ind) negative sampling strategies (NSS). Avg. Rank is computed across datasets for each NSS. The best model is shown in **bold** and the second best model is underlined. Shaded cells indicate relative performance across top 50% models on a given dataset.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	PINT	NAT	DyGFormer	TPNet	DSRD
rnd	Wikipedia	94.12±0.68	93.20±0.41	91.39±0.11	96.08±0.03	98.21±0.01	91.10±0.00	91.98±0.46	96.15±0.07	98.22±0.04	97.89±0.08	98.57±0.03	99.14±0.00	**99.76±0.07**
	Reddit	98.23±0.08	98.16±0.05	96.02±0.29	98.63±0.02	98.70±0.03	95.49±0.00	90.71±0.27	96.81±0.01	99.17±0.02	98.95±0.11	98.79±0.00	99.35±0.01	**99.98±0.00**
	MOOC	82.36±0.60	79.97±1.69	80.23±0.61	88.62±0.67	76.10±0.10	59.35±0.00	79.09±0.01	81.35±0.07	88.40±1.57	86.04±0.73	79.66±0.97	92.66±0.64	**99.25±0.18**
	LastFM	69.51±0.84	68.27±0.97	64.08±0.28	64.24±0.38	83.41±0.27	77.95±0.00	70.46±2.28	71.39±0.04	85.39±0.55	84.74±2.20	89.96±0.32	92.44±0.30	**99.91±0.06**
	Myket	89.11±0.13	88.06±0.48	60.03±1.72	87.46±0.39	84.71±0.55	58.15±0.00	62.36±1.04	86.72±0.01	90.06±0.04	86.21±0.53	79.26±0.14	91.88±0.14	**99.98±0.00**
	Enron	82.55±2.38	74.36±1.17	65.84±0.37	80.99±4.24	88.70±0.19	84.83±0.00	69.17±0.27	81.45±0.08	92.40±0.06	90.94±0.15	86.00±0.37	93.18±0.11	**94.66±0.88**
	Social Evo.	88.10±0.59	85.57±0.41	92.67±0.16	90.93±1.15	85.01±0.12	76.70±0.00	93.31±0.06	92.85±0.04	94.40±0.05	87.29±5.33	94.38±0.04	94.68±0.05	**99.62±0.01**
	UCI	71.72±3.05	47.30±1.09	79.08±0.50	86.92±1.00	93.61±0.43	75.72±0.00	84.51±0.17	92.87±0.71	95.84±0.05	89.59±0.72	95.32±0.01	**96.82±0.14**	95.29±0.20
	Flights	94.44±1.55	94.65±0.57	93.65±0.06	98.19±0.20	98.50±0.00	90.39±0.00	91.19±0.02	90.84±0.02	98.80±0.04	98.53±0.10	98.92±0.00	98.92±0.02	**99.84±0.00**
	Can. Parl.	72.72±1.50	56.27±2.97	67.51±2.81	63.46±1.02	69.44±3.23	66.42±0.00	62.74±6.31	77.83±0.27	70.02±0.18	75.41±2.41	81.58±0.67	85.89±1.66	**99.89±0.04**
	US Legis.	78.82±0.12	65.71±5.24	69.57±1.54	75.89±0.45	70.96±1.85	61.80±0.00	69.70±0.98	71.55±0.94	74.90±0.38	81.03±0.27	71.92±0.58	81.59±0.56	**99.97±0.00**
	UN Trade	64.37±0.47	59.53±0.84	60.75±0.18	64.72±0.55	61.07±0.61	63.00±0.00	59.77±0.17	60.15±0.16	64.40±1.15	69.30±2.34	59.41±0.32	75.94±2.26	**99.29±0.08**
	UN Vote	63.58±0.20	62.06±0.10	51.89±0.63	55.84±0.80	52.05±0.43	60.84±0.00	52.25±0.18	52.31±0.03	68.13±1.16	64.15±1.86	51.85±0.01	74.48±0.14	**96.40±0.98**
	Contact	92.76±0.06	94.37±0.36	95.40±0.09	96.23±0.65	89.39±1.06	92.75±0.00	91.70±0.31	91.64±0.06	97.58±0.19	97.38±0.34	98.13±0.02	98.47±0.02	**99.09±0.01**
	Avg. Rank	7.50	9.86	10.29	7.36	8.21	10.64	10.43	8.21	3.93	5.29	6.11	1.96	**1.21**
hist	Wikipedia	85.72±0.24	83.77±0.42	86.03±0.37	88.65±0.18	74.47±2.21	76.07±0.00	87.10±0.74	91.28±0.21	92.74±0.14	73.75±4.63	75.02±3.44	86.23±1.39	**94.53±0.17**
	Reddit	81.48±0.61	81.34±0.47	75.17±0.24	81.51±0.57	80.17±0.72	76.27±0.00	71.16±0.16	77.76±0.33	85.25±0.78	82.45±0.81	77.50±3.43	80.56±0.53	**97.93±0.25**
	MOOC	76.09±2.62	77.96±3.89	78.98±0.33	83.60±0.48	66.81±2.54	61.17±0.00	75.75±0.63	76.88±0.51	82.22±3.51	84.44±0.19	76.77±2.01	85.81±1.49	**97.65±0.23**
	LastFM	73.73±1.35	73.65±1.59	69.17±0.59	74.71±3.30	64.68±0.26	73.42±0.00	66.09±1.14	67.26±0.40	80.70±1.34	77.56±0.69	76.22±0.63	83.86±0.41	**99.04±0.16**
	Myket	62.97±0.65	63.02±0.29	51.86±0.79	60.93±1.63	47.72±4.50	52.00±0.00	52.01±0.69	58.50±0.43	63.55±0.05	57.80±1.19	42.82±0.29	59.07±0.39	**97.76±0.43**
	Enron	74.46±4.08	69.14±4.19	61.17±0.66	72.31±1.22	66.58±0.29	78.76±0.00	63.08±0.59	79.85±1.56	85.71±0.24	76.25±2.22	75.03±0.57	83.74±0.93	**90.64±0.18**
	Social Evo.	82.66±12.78	91.04±0.67	94.58±0.24	88.82±2.28	86.87±0.14	82.58±0.00	96.15±0.08	95.05±0.12	96.97±0.11	90.78±5.68	97.24±0.12	96.46±0.30	**99.45±0.26**
	UCI	70.04±5.07	41.79±0.38	68.35±0.94	73.53±2.77	58.45±1.85	64.66±0.00	69.43±0.83	85.92±0.84	93.95±0.12	72.39±0.34	74.43±0.47	86.65±0.20	**97.75±0.48**
	Flights	70.86±1.32	70.89±1.04	73.25±0.21	73.29±1.32	65.57±0.70	72.63±0.00	71.41±0.10	72.14±0.24	70.01±1.25	69.58±2.64	67.71±0.51	72.72±0.46	**86.85±1.12**
	Can. Parl.	53.85±0.81	57.28±2.93	67.47±2.74	59.82±4.67	63.47±2.10	65.77±0.00	63.72±12.50	73.08±1.12	71.19±0.77	68.72±8.02	77.44±0.53	80.67±4.28	**99.57±0.51**
	US Legis.	47.26±0.66	61.67±3.16	63.56±10.17	85.26±0.50	74.59±9.37	68.10±0.00	74.53±6.76	70.29±1.32	52.63±3.09	92.56±1.09	79.78±2.87	94.97±1.24	**99.60±0.22**
	UN Trade	56.85±1.02	61.62±1.09	51.74±1.13	61.99±1.19	55.33±2.22	83.26±0.00	54.53±1.63	54.64±1.82	60.68±0.53	64.02±3.57	61.53±3.79	67.93±1.10	**99.83±0.02**
	UN Vote	69.68±0.62	64.82±0.31	51.30±0.65	53.09±4.34	49.05±1.19	87.01±0.00	51.30±0.93	51.81±2.15	74.99±1.06	64.29±7.15	52.76±0.06	75.82±1.45	**97.60±0.57**
	Contact	94.38±1.05	91.96±2.42	96.01±0.15	93.51±2.48	83.16±1.12	89.33±0.00	94.19±0.34	93.50±0.40	96.47±0.28	96.45±0.73	**97.45±0.23**	97.40±0.32	95.88±0.48
	Avg. Rank	8.43	8.43	8.93	6.21	11.14	8.57	9.36	7.21	4.50	6.29	7.07	3.50	**1.36**
ind	Wikipedia	81.11±0.06	78.42±0.69	84.35±0.51	**88.32±0.26**	81.24±2.09	83.36±0.00	85.72±0.37	87.99±0.21	86.99±0.31	61.47±4.39	71.94±7.38	85.76±2.03	87.07±0.68
	Reddit	88.09±0.21	87.59±0.32	83.66±0.60	87.94±0.62	90.89±0.62	87.91±0.00	78.84±0.40	84.25±0.15	88.59±0.20	74.90±1.89	83.44±1.72	88.98±0.21	**98.42±0.34**
	MOOC	69.42±1.75	66.45±2.23	75.96±0.81	75.65±1.08	67.89±2.42	51.70±0.00	74.21±0.80	74.27±0.07	72.02±4.63	78.47±0.40	73.07±2.37	81.11±0.43	**97.33±0.49**
	LastFM	62.88±0.61	63.61±0.93	68.51±0.62	66.67±4.35	64.27±1.57	73.93±0.00	61.05±0.48	62.33±0.35	70.65±0.61	62.90±1.04	67.74±2.81	71.67±0.69	**98.80±0.07**
	Myket	62.05±0.14	62.11±0.12	52.36±0.45	61.96±0.65	50.02±5.01	55.78±0.00	52.78±0.84	59.84±0.07	64.03±0.05	44.44±0.66	35.52±0.22	45.66±0.32	**97.75±0.49**
	Enron	71.88±3.27	66.55±2.56	61.78±0.39	71.07±0.71	76.74±0.16	75.71±0.00	64.02±0.37	76.77±0.88	81.98±0.15	75.23±1.08	77.67±0.49	82.28±0.93	**90.86±1.03**
	Social Evo.	82.78±12.78	90.43±0.62	94.41±0.25	88.73±2.52	89.02±0.15	85.34±0.00	96.18±0.10	94.92±0.12	97.13±0.11	91.63±5.39	97.44±0.10	96.78±0.25	**99.45±0.25**
	UCI	60.71±0.89	50.92±0.31	67.59±0.96	71.44±1.82	59.76±3.00	58.01±0.00	67.87±0.47	79.04±0.35	84.53±0.16	61.33±0.29	59.39±0.49	79.80±0.28	**94.24±0.60**
	Flights	73.01±0.83	72.59±1.22	76.47±0.22	77.42±2.01	69.95±0.64	82.91±0.00	75.22±0.18	75.61±0.29	64.58±0.16	64.86±3.12	71.74±0.59	73.57±1.03	**88.30±0.99**
	Can. Parl.	49.28±0.78	55.78±3.45	66.06±1.38	54.79±3.66	63.41±2.22	63.40±0.00	61.03±7.32	67.11±0.75	61.01±0.53	68.01±8.98	82.15±0.50	78.29±6.36	**99.99±0.00**
	US Legis.	46.63±0.39	61.99±4.97	62.69±11.24	83.66±0.96	67.89±11.51	70.05±0.00	68.37±5.73	69.58±1.46	50.03±2.07	92.18±1.39	79.20±3.50	92.78±1.52	**98.42±1.67**
	UN Trade	62.13±0.77	64.77±1.43	54.62±1.00	63.99±1.91	60.46±2.74	79.86±0.00	57.44±2.57	57.95±2.30	64.19±0.29	64.27±3.19	66.84±4.56	71.13±1.71	**99.18±0.09**
	UN Vote	69.52±0.93	65.59±0.54	53.83±0.85	51.54±3.83	53.40±2.02	73.53±0.00	49.22±1.24	51.62±0.11	69.91±1.09	61.91±9.35	50.77±0.72	77.39±0.69	**94.42±0.28**
	Contact	91.07±1.81	89.51±2.55	93.99±0.74	91.69±2.65	89.46±1.19	87.01±0.00	91.55±0.30	90.79±0.35	93.38±0.68	91.88±0.95	94.83±0.34	**95.88±0.26**	94.55±0.45
	Avg. Rank	9.14	9.21	7.64	6.79	8.86	7.29	8.93	7.00	5.86	8.07	7.29	3.64	**1.29**
Overall Avg. Rank		8.36	9.17	8.95	6.79	9.40	8.83	9.57	7.48	4.76	6.55	6.82	3.04	**1.29**

Table 5: ROC-AUC for transductive dynamic link prediction with random, historical, and inductive negative sampling strategies (NSS). Avg. Rank is computed across datasets for each NSS. The best model is shown in **bold** and the second best model is underlined. Shaded cells indicate relative performance across top 50% models on a given dataset.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	PINT	NAT	DyGFormer	TPNet	DSRD
rnd	Wikipedia	93.80±0.53	92.79±0.42	90.60±0.12	95.84±0.07	97.74±0.01	91.50±0.00	91.14±0.52	95.85±0.08	97.88±0.06	97.61±0.10	98.36±0.01	99.10±0.01	**99.74±0.07**
	Reddit	98.18±0.07	98.15±0.04	96.04±0.29	98.64±0.02	98.48±0.04	95.99±0.00	90.89±0.25	96.70±0.01	99.12±0.03	98.94±0.12	98.67±0.03	99.31±0.00	**99.98±0.00**
	MOOC	86.41±0.58	84.66±1.28	81.89±0.58	90.94±0.59	76.25±0.11	62.22±0.00	80.70±0.14	82.80±0.12	90.32±1.56	87.86±0.67	78.68±1.31	93.51±0.79	**99.21±0.20**
	LastFM	70.32±0.45	68.99±0.91	63.04±0.17	65.58±0.73	81.81±0.38	82.80±0.00	64.21±2.09	69.11±0.06	84.45±0.47	83.19±3.09	89.85±0.43	92.00±0.34	**99.89±0.09**
	Myket	88.40±0.10	87.11±0.62	60.60±1.48	86.84±0.52	82.99±0.51	58.19±0.00	65.52±0.49	85.24±0.02	89.76±0.02	85.24±0.18	78.46±0.10	90.90±0.12	**99.98±0.00**
	Enron	86.15±1.58	77.48±1.09	64.15±0.41	83.56±3.63	89.09±0.34	88.04±0.00	66.80±0.51	84.28±0.12	92.75±0.10	91.93±0.06	85.69±0.47	**94.02±0.20**	93.00±1.82
	Social Evo.	90.63±0.48	88.80±0.24	94.24±0.15	92.99±1.15	87.08±0.15	83.21±0.00	95.05±0.10	94.89±0.02	96.09±0.02	89.04±5.60	96.01±0.03	96.39±0.05	**99.60±0.04**
	UCI	74.53±2.30	50.39±2.02	77.43±0.92	86.66±1.15	92.25±0.46	76.67±0.00	84.15±0.18	90.91±0.89	94.90±0.06	87.56±0.92	94.10±0.01	**96.06±0.16**	94.29±0.33
	Flights	95.13±1.29	95.34±0.41	93.82±0.03	98.43±0.20	98.46±0.01	91.27±0.00	91.24±0.02	91.13±0.01	98.86±0.06	98.61±0.06	98.95±0.00	98.98±0.02	**99.81±0.00**
	Can. Parl.	82.30±0.93	61.65±3.33	72.39±4.08	65.10±1.25	74.48±4.08	66.29±0.00	65.65±6.50	82.66±0.55	81.68±0.26	79.59±4.07	83.78±0.66	89.80±0.99	**99.90±0.04**
	US Legis.	85.70±0.02	70.49±3.99	76.47±0.94	82.38±0.18	76.86±1.83	66.72±0.00	75.12±1.28	77.75±0.51	82.43±0.47	87.30±0.06	78.82±0.46	87.81±0.19	**99.98±0.00**
	UN Trade	70.16±0.24	64.89±1.01	63.31±0.12	69.05±0.37	64.11±0.24	69.96±0.00	62.83±0.17	61.75±0.53	69.13±0.67	74.12±2.35	64.35±0.41	80.62±0.87	**99.19±0.08**
	UN Vote	68.86±0.23	67.40±0.17	52.55±0.55	59.23±0.88	52.51±0.45	66.36±0.00	52.11±0.30	52.63±0.07	73.05±1.13	69.14±1.86	52.78±0.06	79.07±0.16	**96.09±1.07**
	Contact	95.05±0.18	95.27±0.26	96.28±0.02	97.25±0.41	88.64±1.65	94.47±0.00	93.55±0.23	93.79±0.04	97.87±0.15	97.76±0.32	98.36±0.02	98.71±0.03	**99.14±0.01**
	Avg. Rank	7.07	9.36	10.43	7.57	8.57	10.07	11.07	8.71	3.86	5.36	5.86	1.86	**1.21**
hist	Wikipedia	84.05±0.13	81.60±0.35	80.10±0.37	84.94±0.17	73.16±1.05	79.71±0.00	81.25±1.41	87.89±0.27	90.19±0.22	74.82±2.53	77.14±2.37	83.52±1.04	**92.40±0.24**
	Reddit	82.78±0.58	82.66±0.61	75.25±0.43	82.34±0.53	79.67±0.38	81.20±0.00	70.99±0.19	77.42±0.29	84.06±0.28	81.97±0.47	75.97±2.33	82.72±0.14	**97.68±0.31**
	MOOC	81.43±0.83	83.96±2.09	75.46±0.48	87.07±0.33	59.11±5.04	62.06±0.00	69.01±0.69	73.52±1.41	85.47±2.27	84.88±0.39	70.19±2.47	86.86±1.95	**97.50±0.27**
	LastFM	75.74±0.76	75.73±1.62	62.00±0.42	75.24±2.64	63.15±0.37	77.87±0.00	55.11±0.55	58.10±0.28	76.44±2.18	74.40±0.74	74.40±0.52	80.41±0.71	**98.81±0.24**
	Myket	64.17±0.29	62.61±0.42	50.93±0.80	62.26±0.99	42.15±6.20	53.07±0.00	50.75±0.42	59.19±0.19	63.86±0.10	57.25±1.01	31.95±0.24	53.64±0.50	**97.07±0.39**
	Enron	78.93±3.04	71.45±5.43	57.86±1.19	75.92±1.44	67.12±0.09	81.39±0.00	58.18±1.27	76.97±2.66	82.97±0.36	77.06±0.93	73.25±0.77	83.27±0.71	**85.90±0.27**
	Social Evo.	88.12±7.59	91.83±0.53	92.20±0.28	88.15±1.99	88.03±0.11	87.45±0.00	95.30±0.25	93.63±0.17	96.90±0.10	91.26±5.36	97.31±0.11	96.93±0.19	**99.32±0.33**
	UCI	74.50±2.13	38.58±0.66	58.25±0.53	67.80±4.19	53.31±0.46	68.53±0.00	60.47±1.15	80.48±1.39	92.10±0.18	67.10±0.61	74.25±0.19	80.75±0.20	**96.86±0.75**
	Flights	74.17±1.32	74.15±0.54	73.04±0.13	74.82±1.06	66.45±0.50	76.75±0.00	71.25±0.08	71.15±0.28	72.59±1.21	72.61±1.86	68.53±0.54	74.66±0.80	**84.39±1.04**
	Can. Parl.	66.00±1.30	62.64±2.11	72.35±4.09	63.06±4.01	67.28±3.14	65.78±0.00	67.05±14.88	76.83±0.75	84.44±0.93	73.84±8.09	81.82±0.54	82.30±4.24	**99.03±0.72**
	US Legis.	54.46±1.38	69.47±1.87	75.03±8.41	88.76±0.45	81.62±5.82	72.36±0.00	77.31±6.66	81.17±0.84	63.22±4.87	95.30±0.83	87.24±2.04	96.23±0.96	**99.84±0.05**
	UN Trade	65.77±1.27	68.22±1.06	55.65±1.32	66.00±1.28	59.55±1.33	87.78±0.00	58.39±1.20	53.69±2.42	66.80±0.34	71.57±2.88	64.86±4.01	73.09±2.12	**99.82±0.02**
	UN Vote	76.23±0.27	71.92±0.59	52.09±1.39	57.16±3.94	48.80±0.87	91.00±0.00	52.02±1.09	53.62±2.55	80.87±0.91	71.45±5.14	54.42±0.13	79.27±1.45	**97.34±0.91**
	Contact	96.10±0.54	93.05±1.43	95.18±0.15	94.49±1.40	82.14±1.56	92.52±0.00	93.61±0.49	93.40±0.32	96.37±0.02	96.15±0.80	97.04±0.24	**97.07±0.22**	95.93±0.52
	Avg. Rank	6.36	7.93	9.57	6.57	11.07	7.50	10.14	8.07	4.21	6.82	7.96	3.43	**1.36**
ind	Wikipedia	77.31±0.10	75.64±0.70	77.98±0.48	**84.50±0.15**	78.17±0.89	83.99±0.00	79.44±1.04	82.57±0.13	81.41±0.17	64.94±2.89	71.53±6.23	80.89±1.91	84.14±0.51
	Reddit	85.85±0.22	85.52±0.29	81.53±0.72	85.69±0.43	87.49±0.54	88.20±0.00	76.58±0.38	81.51±0.14	82.46±0.11	74.90±1.52	79.01±3.12	83.67±0.20	**98.17±0.40**
	MOOC	73.95±1.12	71.67±2.06	71.90±0.44	77.46±1.75	60.98±4.66	52.53±0.00	68.00±1.03	70.80±0.49	73.89±4.26	79.76±0.70	66.20±2.86	83.01±0.90	**96.92±0.60**
	LastFM	64.55±0.25	64.78±0.94	61.72±0.44	66.38±3.37	65.42±1.32	75.31±0.00	51.59±1.48	53.80±0.08	66.63±1.26	60.29±1.05	61.88±1.70	66.33±1.18	**98.68±0.13**
	Myket	60.26±0.28	60.91±0.16	51.24±0.42	61.62±0.42	45.17±6.70	56.73±0.00	51.95±0.59	57.97±0.09	62.99±0.00	43.19±1.10	10.06±0.23	30.18±0.47	**96.94±0.52**
	Enron	74.39±2.82	67.00±3.52	58.57±0.91	73.36±1.01	76.31±0.08	76.59±0.00	59.22±1.03	73.42±1.67	79.89±0.32	74.88±0.95	73.49±0.76	80.77±0.67	**86.99±2.58**
	Social Evo.	88.10±7.56	91.63±0.52	92.01±0.28	88.09±2.33	89.99±0.12	89.22±0.00	95.27±0.25	93.45±0.17	97.04±0.09	92.15±4.94	97.47±0.10	97.19±0.14	**99.31±0.32**
	UCI	60.73±0.69	51.19±0.59	58.46±0.52	66.49±2.21	56.10±0.83	58.70±0.00	61.23±0.93	74.67±0.47	80.07±0.12	58.54±0.21	59.99±0.22	73.04±0.32	**93.24±0.75**
	Flights	74.87±1.14	75.00±0.58	74.44±0.19	77.92±1.55	70.17±0.69	82.93±0.00	73.15±0.13	73.25±0.32	62.54±1.43	68.08±1.92	69.76±0.72	73.16±0.88	**85.73±0.94**
	Can. Parl.	56.62±1.72	61.20±2.41	69.53±2.42	59.91±2.68	66.86±2.84	63.18±0.00	61.07±9.06	67.17±0.36	67.89±0.35	73.70±8.35	83.21±0.19	80.44±4.09	**99.98±0.01**
	US Legis.	53.07±0.85	69.66±3.61	73.47±9.57	87.82±0.52	76.97±8.36	73.79±0.00	73.91±5.79	80.19±1.22	59.23±3.64	95.08±0.92	86.71±2.09	94.83±1.21	**98.31±2.20**
	UN Trade	68.85±0.93	70.37±1.79	59.76±1.41	68.36±1.48	65.72±1.18	81.56±0.00	62.47±1.80	56.68±2.84	70.07±0.19	72.25±2.63	70.28±4.30	75.10±2.79	**99.08±0.12**
	UN Vote	75.62±1.46	72.95±0.85	53.75±1.13	56.40±3.98	54.94±2.30	80.30±0.00	51.17±1.91	51.43±0.43	76.56±0.80	69.56±7.40	50.79±1.01	81.31±1.06	**94.65±0.67**
	Contact	93.42±1.09	91.04±1.54	93.81±0.32	93.06±1.48	86.95±1.85	87.50±0.00	91.54±0.47	91.07±0.29	93.70±0.19	92.87±0.95	94.91±0.35	**95.77±0.23**	94.69±0.53
	Avg. Rank	7.71	8.29	8.79	6.29	8.79	6.29	9.50	8.07	5.71	7.93	8.07	4.36	**1.21**
Overall Avg. Rank		7.05	8.52	9.60	6.81	9.48	7.95	10.24	8.29	4.60	6.70	7.30	3.21	**1.26**

Table 6: AP for transductive dynamic link prediction against recent temporal-walk, heuristic, and state-space baselines under random negative sampling. Avg. Rank is computed across datasets. The best model is shown in **bold** and the second best model is underlined. Shaded cells indicate relative performance across top 50% models on a given dataset.

Dataset	CTWalks	TAWRMAC	GraphHeuristic	GRN	DSRD
Wikipedia	97.88±0.23	99.29±0.02	94.47±0.00	97.49±0.35	**99.76±0.07**
Reddit	96.46±0.02	99.30±0.01	98.32±0.00	99.50±0.03	**99.98±0.00**
MOOC	95.88±0.07	91.46±0.66	67.65±0.00	91.63±1.18	**99.25±0.18**
Enron	93.35±0.54	92.74±0.21	90.20±0.00	83.15±2.49	**94.66±0.88**
UCI	**98.65±0.08**	96.40±0.15	89.72±0.00	80.55±1.38	95.29±0.20
Can. Parl.	69.22±0.25	76.42±1.47	70.60±0.00	91.91±0.23	**99.89±0.04**
US Legis.	57.75±0.89	73.89±0.52	60.83±0.00	79.21±3.82	**99.97±0.00**
UN Trade	61.28±0.18	68.11±0.90	55.74±0.00	71.54±0.68	**99.29±0.08**
Avg. Rank	3.38	2.88	4.38	3.13	**1.25**

C.5 Additional Results of Inductive Dynamic Link Prediction

The main inductive AP results are presented in Table 1 of Section 4.2, and the corresponding ROC-AUC scores across all datasets are reported in Table 8. The inductive AP against the four recent baselines CTWalks (Yu & Liu, 2025), TAWRMAC (Farokhi et al., 2026), GraphHeuristic (Cornell et al., 2025), and GRN (Chang et al., 2026) is reported in Table 7, where DSRD again achieves the best average rank and generalizes stably to unseen nodes.

Table 7: AP for inductive dynamic link prediction against recent temporal-walk, heuristic, and state-space baselines under random negative sampling. Avg. Rank is computed across datasets. The best model is shown in **bold** and the second best model is underlined. Shaded cells indicate relative performance across top 50% models on a given dataset.

Dataset	CTWalks	TAWRMAC	GraphHeuristic	GRN	DSRD
Wikipedia	96.01±0.22	98.93±0.03	91.48±0.00	95.69±0.57	**99.55±0.10**
Reddit	93.62±0.01	98.99±0.05	97.63±0.00	98.57±0.16	**99.94±0.02**
MOOC	93.81±0.18	91.14±0.82	61.60±0.00	90.59±0.77	**95.75±0.66**
Enron	93.17±0.19	89.45±0.12	84.62±0.00	81.39±2.94	**93.78±1.10**
UCI	**96.35±0.16**	95.08±0.33	83.73±0.00	75.27±1.35	92.59±0.05
Can. Parl.	54.98±0.31	55.90±0.83	56.86±0.00	77.58±1.90	**99.93±0.02**
US Legis.	47.52±0.18	59.28±1.12	47.03±0.00	77.52±4.37	**84.42±1.03**
UN Trade	61.74±0.44	63.86±1.83	52.88±0.00	64.26±2.15	**99.23±0.07**
Avg. Rank	3.25	2.75	4.38	3.38	**1.25**

Table 8: ROC-AUC for inductive dynamic link prediction with random, historical, inductive negative sampling strategies (NSS). Avg. Rank is computed across datasets for each NSS. The best model is shown in **bold** and the second best model is underlined. Shaded cells indicate relative performance across top 50% models on a given dataset.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	PINT	NAT	DyGFormer	TPNet	DSRD
rnd	Wikipedia	88.42±0.19	85.71±0.27	89.69±0.28	93.37±0.06	97.03±0.02	90.47±0.34	94.73±0.06	97.41±0.13	90.11±0.22	97.58±0.05	98.63±0.02	**99.52±0.10**
	Reddit	92.72±0.21	91.14±0.18	92.99±0.48	92.91±0.27	97.58±0.07	86.38±0.27	94.37±0.02	98.09±0.06	92.72±0.03	97.99±0.04	97.99±0.07	**99.94±0.02**
	MOOC	80.40±2.43	76.37±1.08	79.57±0.63	87.84±0.65	73.00±0.59	78.30±0.22	80.98±0.17	88.96±0.59	81.57±0.46	74.12±2.55	87.37±0.97	**95.28±0.74**
	LastFM	81.77±0.12	79.39±3.05	74.68±0.09	78.43±1.54	86.01±0.28	75.29±2.25	80.37±0.13	87.71±1.33	87.11±0.99	92.71±0.37	93.88±0.18	**99.93±0.04**
	Myket	79.72±0.22	74.63±0.73	55.86±1.04	70.27±0.51	73.82±0.66	57.35±0.73	78.18±0.02	77.07±0.15	58.46±2.81	68.75±0.15	82.31±0.27	**99.92±0.02**
	Enron	76.93±2.18	67.76±3.08	60.86±2.06	69.31±4.30	85.19±0.44	63.06±1.23	76.52±0.66	86.36±0.19	85.20±0.39	81.27±0.47	88.54±0.11	**92.77±2.11**
	Social Evo.	91.33±0.56	83.93±0.58	92.63±0.18	87.46±1.01	84.27±0.12	94.25±0.08	93.74±0.02	94.39±0.05	82.12±10.11	95.14±0.03	**95.18±0.05**	91.64±2.57
	UCI	73.83±1.41	48.06±1.63	76.60±0.97	76.05±0.56	88.47±0.49	79.13±0.13	88.95±0.50	93.10±0.10	77.49±0.72	91.43±0.22	**93.56±0.09**	90.59±0.15
	Flights	90.82±1.20	88.39±0.47	87.67±0.25	93.32±0.06	96.83±0.05	81.83±0.07	81.65±0.02	96.71±0.21	93.06±0.49	**97.79±0.01**	96.74±0.24	95.94±0.17
	Can. Parl.	48.69±2.48	52.21±2.00	56.57±1.49	43.95±4.22	58.08±1.23	55.29±1.43	58.11±1.00	51.05±1.25	54.43±1.80	70.84±0.17	57.66±1.35	**99.92±0.01**
	US Legis.	55.95±0.21	57.60±1.35	48.04±0.49	57.13±0.57	52.23±1.52	49.82±1.42	48.22±0.60	51.85±1.83	62.07±1.97	53.02±0.78	61.06±1.86	**83.13±0.56**
	UN Trade	59.31±0.68	57.03±0.65	61.50±0.14	53.77±2.66	64.27±0.02	62.79±0.14	59.50±0.88	62.38±1.95	67.95±4.16	62.62±0.11	81.26±2.17	**99.18±0.08**
	UN Vote	53.44±1.28	45.94±2.25	51.23±0.93	44.61±5.59	46.00±0.77	49.34±1.06	50.53±0.53	56.97±1.58	66.03±2.01	49.88±0.25	52.91±3.03	**95.74±0.65**
	Contact	94.49±0.40	89.04±0.36	95.88±0.03	91.68±1.41	87.61±1.83	92.38±0.26	92.63±0.02	95.85±0.74	92.72±0.42	98.16±0.03	**98.21±0.06**	96.78±0.41
	Avg. Rank	7.79	9.71	8.71	8.64	6.86	8.86	7.00	4.50	6.50	4.79	2.57	**2.07**
hist	Wikipedia	60.40±0.08	56.21±1.06	77.70±0.88	75.26±0.58	64.22±0.92	78.37±0.70	81.09±0.04	74.26±0.65	37.71±3.16	58.68±4.49	68.53±1.44	**83.10±0.51**
	Reddit	58.39±0.43	57.29±0.88	61.64±0.33	61.95±0.52	64.71±0.44	59.85±0.23	64.64±0.23	63.49±0.09	51.41±0.89	61.26±1.12	61.86±0.22	**91.92±1.13**
	MOOC	60.98±1.73	62.47±1.13	69.94±0.23	72.31±2.49	59.21±4.10	66.76±1.09	69.98±0.51	69.07±3.05	66.00±0.73	61.83±3.50	75.42±1.36	**87.85±2.06**
	LastFM	68.43±0.62	67.21±2.52	72.41±0.33	71.53±1.21	66.16±0.13	64.23±0.97	70.01±0.09	73.82±0.39	63.37±0.58	65.87±2.57	74.87±0.94	**98.82±0.06**
	Myket	43.58±0.36	45.85±0.90	49.85±0.57	47.38±0.39	38.90±6.37	49.65±0.71	51.75±0.10	46.89±0.33	28.36±2.50	9.49±0.13	20.33±0.41	**90.65±1.60**
	Enron	63.70±3.48	57.01±1.67	53.61±1.75	58.51±0.49	63.76±0.28	55.74±0.52	69.44±1.19	77.88±0.66	62.63±0.99	64.93±0.44	71.94±1.40	**89.72±0.32**
	Social Evo.	85.47±2.69	81.56±1.61	90.50±0.34	82.26±1.76	83.41±0.01	95.64±0.26	92.85±0.11	96.39±0.16	81.07±7.85	**97.06±0.07**	96.64±0.09	95.29±2.61
	UCI	66.30±0.72	51.87±0.33	58.70±0.88	59.47±0.29	50.60±0.50	61.19±1.07	75.46±0.38	81.84±0.11	46.01±0.62	59.15±0.23	70.13±0.36	**92.38±0.88**
	Flights	58.60±0.41	59.06±0.73	63.82±0.42	56.88±0.79	56.65±0.80	63.04±0.21	63.11±0.12	48.11±0.14	46.43±1.24	56.13±0.45	52.06±0.54	**71.02±1.12**
	Can. Parl.	46.12±1.93	48.44±1.61	59.09±0.79	44.53±2.35	59.53±1.34	54.76±1.82	56.84±1.02	46.79±0.73	54.87±0.94	71.47±0.08	58.74±1.89	**99.83±0.00**
	US Legis.	58.38±1.14	62.26±2.89	48.86±1.75	61.23±0.43	52.68±1.25	50.15±1.56	51.75±0.77	53.49±1.99	63.54±3.15	51.58±0.51	**66.35±4.08**	52.60±3.47
	UN Trade	57.43±0.51	58.72±1.53	55.64±1.32	55.67±1.57	59.45±1.86	58.93±1.09	52.13±2.41	60.70±1.84	67.96±3.80	64.52±3.36	70.69±0.84	**95.42±0.57**
	UN Vote	61.78±0.78	54.92±1.46	48.76±1.86	52.29±2.37	47.63±1.53	52.27±0.88	45.18±0.55	61.35±1.86	61.29±0.45	52.17±1.10	66.26±1.01	**90.06±2.06**
	Contact	89.61±0.45	84.66±0.29	93.63±0.13	84.79±4.09	73.42±1.08	90.93±0.47	90.12±0.35	89.50±1.05	83.97±0.97	93.64±0.39	94.18±0.57	**96.52±0.39**
	Avg. Rank	7.57	8.43	6.71	7.00	8.07	7.00	5.57	5.43	9.00	7.21	4.21	**1.79**
ind	Wikipedia	60.38±0.09	56.20±1.06	77.68±0.88	75.26±0.58	64.17±0.92	78.35±0.70	81.10±0.05	74.27±0.65	37.70±3.16	58.69±4.49	68.53±1.44	**83.24±0.49**
	Reddit	58.39±0.43	57.29±0.88	61.66±0.29	61.95±0.52	64.70±0.44	59.85±0.24	64.64±0.23	63.49±0.09	51.42±0.89	61.25±1.12	61.86±0.22	**91.96±1.15**
	MOOC	60.98±1.73	62.48±1.13	69.94±0.24	72.31±2.49	59.21±4.10	66.76±1.09	69.98±0.51	69.08±3.05	66.00±0.73	61.83±3.50	75.42±1.36	**88.19±1.89**
	LastFM	68.43±0.62	67.21±2.52	72.42±0.33	71.53±1.21	66.17±0.13	64.23±0.97	70.02±0.10	73.82±0.39	63.37±0.58	65.87±2.57	74.87±0.94	**98.83±0.06**
	Myket	43.59±0.35	45.86±0.91	49.85±0.56	47.38±0.39	38.90±6.37	49.65±0.71	51.75±0.10	46.90±0.33	28.36±2.50	9.49±0.13	20.33±0.41	**90.66±1.61**
	Enron	63.70±3.48	57.01±1.67	53.59±1.76	58.51±0.49	63.77±0.28	55.74±0.48	69.44±1.19	77.88±0.66	62.63±0.99	64.93±0.45	71.94±1.40	**89.59±0.25**
	Social Evo.	85.47±2.69	81.56±1.61	90.50±0.34	82.26±1.76	83.47±0.08	95.64±0.26	92.85±0.11	96.39±0.16	81.07±7.85	**97.07±0.07**	96.64±0.08	95.27±2.61
	UCI	66.30±0.73	51.89±0.34	58.71±0.89	59.48±0.31	50.62±0.51	61.24±1.11	75.49±0.39	81.85±0.11	46.02±0.62	59.20±0.23	70.14±0.35	**92.36±0.94**
	Flights	58.61±0.38	59.08±0.72	63.82±0.42	56.87±0.78	56.61±0.80	63.04±0.21	63.10±0.12	48.10±0.15	46.40±1.24	56.10±0.45	52.03±0.54	**71.04±1.11**
	Can. Parl.	45.77±1.89	48.55±1.72	58.90±0.79	44.08±2.29	59.48±1.54	54.83±2.05	56.83±1.22	46.67±0.81	54.76±0.94	71.31±0.06	58.69±1.94	**99.86±0.01**
	US Legis.	58.38±1.14	62.26±2.89	48.86±1.75	61.23±0.43	52.68±1.25	50.15±1.56	51.75±0.77	53.49±1.99	63.54±3.15	51.58±0.51	**66.35±4.08**	51.76±2.29
	UN Trade	57.31±0.54	58.67±1.55	55.49±1.26	55.63±1.54	59.44±1.87	59.33±1.11	51.99±2.46	60.67±1.83	67.89±3.80	64.54±3.40	70.71±0.84	**95.47±0.55**
	UN Vote	61.71±0.80	54.91±1.53	48.87±1.87	52.15±2.54	47.74±1.57	52.20±0.65	45.23±0.48	61.36±1.87	61.35±0.44	52.26±1.09	66.28±1.03	**90.22±2.09**
	Contact	89.61±0.45	84.66±0.29	93.52±0.25	84.79±4.08	73.42±1.08	90.92±0.47	90.12±0.35	89.50±1.05	83.96±0.97	93.63±0.39	94.18±0.57	**96.51±0.40**
	Avg. Rank	7.57	8.43	6.71	7.14	8.07	6.93	5.57	5.43	9.07	7.07	4.21	**1.79**
Overall Avg. Rank		7.64	8.86	7.38	7.60	7.67	7.60	6.05	5.12	8.19	6.36	3.67	**1.88**

C.6 Statistical Significance of Average Ranking Comparisons

Based on the critical difference diagrams in Figures 7 and 8, several observations can be drawn from a modeling perspective. Overall, the proposed DSRD ranks first or within the top tier across both AP and ROC-AUC metrics under all six evaluation settings. The Friedman test with Holm-corrected post-hoc comparisons (Demšar, 2006) reveals clear statistically significant gaps between DSRD and the majority of baselines. As visually indicated by the horizontal bars in Figures 7 and 8,

DSRD is not connected to most competing methods, confirming that its performance advantage is statistically significant at the 95% confidence level. Notably, the separation between DSRD and lower-ranked methods is substantial across all settings, with rank differences often exceeding 4–6 positions. Even compared to recent competitive baselines such as TPNet and DyGFormer, DSRD maintains a consistent lead, particularly under challenging inductive evaluation protocols where many baselines exhibit performance degradation. These results indicate that the performance gains of DSRD are systematic rather than driven by isolated datasets or sampling strategies.

Across different negative sampling strategies and evaluation protocols, DSRD exhibits notably smaller rank variance than competing methods. In contrast, many memory-based and attention-based baselines suffer pronounced performance degradation under inductive evaluation or harder negative sampling. This behavior suggests that DSRD is less reliant on memorizing historical edges and instead learns transferable temporal–structural patterns that generalize to unseen nodes and shifted interaction distributions. Comparisons with heuristic and memorization-oriented methods further support this interpretation. While EdgeBank achieves competitive rankings in some transductive settings, its performance deteriorates substantially under inductive sampling, reflecting its dependence on repeated interactions. DSRD, by contrast, maintains strong rankings in inductive scenarios, highlighting the advantage of its retentive state formulation over pure historical lookup.

Among methods designed to capture higher-order structure, such as CAWN, NAT, PINT, DyGformer, and TPNet, competitive performance is observed in certain settings, confirming the importance of multi-hop context in dynamic graphs. However, these approaches exhibit larger fluctuations across evaluation regimes, consistent with their reliance on fixed walk lengths, propagation depths, or static decay schemes. By integrating structural diffusion into a unified retentive state with adaptive temporal and topological regulation, DSRD achieves more stable and consistent performance across heterogeneous graph dynamics.

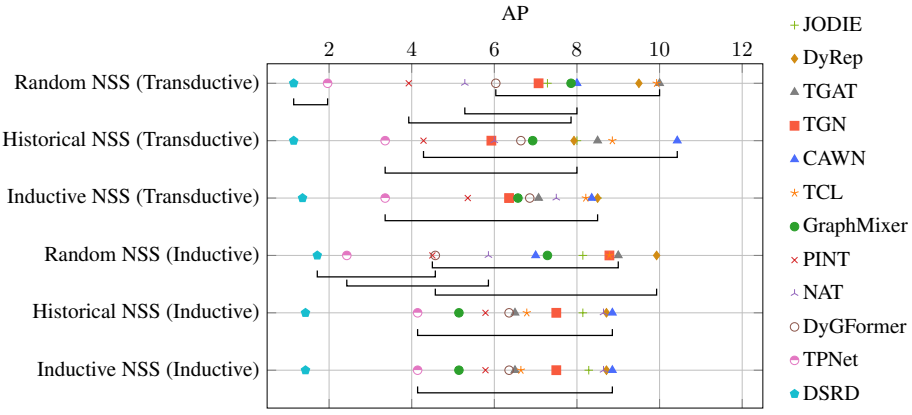


Figure 7: Average AP rank comparison of dynamic graph models across three negative sampling strategies. Statistical significance is assessed using the Friedman test with Holm-corrected post-hoc comparisons at the 95% confidence level.

C.7 Additional Description of Dynamic Node Classification

Since node classification datasets exhibit highly imbalanced label distributions, ROC-AUC alone may not fully reflect model performance on minority classes. We therefore additionally report the area under the precision–recall curve (AUPRC), which is more sensitive to class imbalance. For each dataset, we also report the relative improvement over a random classifier, where the random baseline corresponds to the positive-class ratio.

Table 9 shows that DSRD remains competitive under different degrees of class imbalance. On MOOC and PubMed, DSRD achieves the best AUPRC, reaching 4.10% on MOOC and 87.48% on PubMed, corresponding to 4.37× and 1.11× improvement over the random baselines, respectively. On Wikipedia, despite the severe imbalance, DSRD also obtains the best AUPRC with 2.47%, achieving a 13.09× improvement over random prediction. These results indicate that the proposed

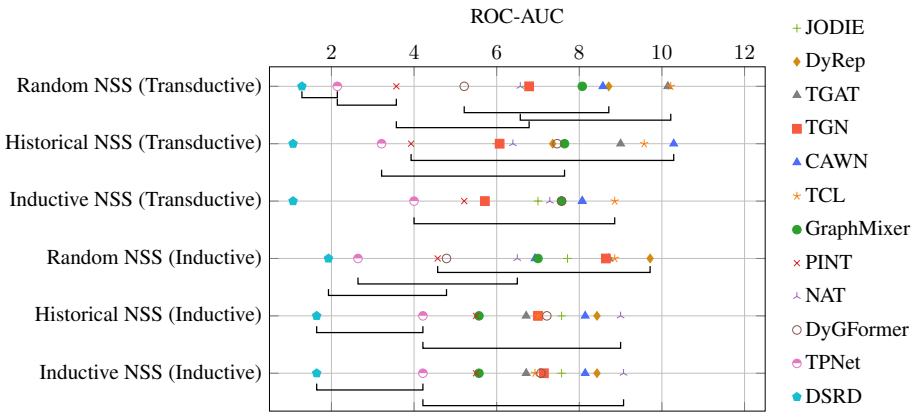


Figure 8: Average ROC-AUC rank comparison of dynamic graph models across three negative sampling strategies. All the graphical conventions are the same as in Figure 7.

Table 9: AUPRC (%) for node classification under imbalanced label distributions. The value in parentheses denotes the relative improvement over the random baseline determined by the positive-class ratio.

Dataset	Imbalance	TGN	PINT	DyGFormer	TPNet	DSRD (ours)
Wikipedia	Severe	1.13 (6.07×)	2.11 (11.33×)	1.42 (7.62×)	0.13 (0.66×)	**2.47 (13.09×)**
Reddit	Severe	**0.17 (1.79×)**	0.10 (1.11×)	0.16 (1.64×)	0.11 (1.10×)	0.13 (1.35×)
MOOC	Severe	1.75 (1.87×)	1.82 (1.94×)	3.77 (4.02×)	1.17 (1.24×)	**4.10 (4.37×)**
PubMed	Moderate	81.88 (1.04×)	82.59 (1.04×)	83.33 (1.05×)	78.11 (0.99×)	**87.48 (1.11×)**

retentive diffusion mechanism can preserve useful minority-class signals even when positive labels are sparse.

For Reddit, all methods obtain very low AUPRC values, and DSRD is slightly behind the best-performing baseline. This dataset represents an extreme imbalance regime, where the positive-class ratio is extremely small and precision–recall performance becomes highly sensitive to a small number of positive predictions. The uniformly low AUPRC values across all methods suggest that such severe imbalance remains challenging for current dynamic graph learning models. Overall, the AUPRC results complement the ROC-AUC evaluation by showing that DSRD is robust across moderately and severely imbalanced settings, while also revealing the limitations of existing methods under extreme label sparsity.

C.8 A Closer Look at Evaluation Regimes in Temporal Link Prediction

We analyze the near-saturated AP and ROC-AUC scores observed for a subset of benchmarks and evaluation settings in Tables 4 and 5. Rather than reflecting trivial memorization, these cases stem from structural properties of commonly used temporal graph datasets and evaluation protocols, and they illustrate that average scores alone can obscure meaningful differences in generalization once conditions vary across temporal, structural, and task-specific regimes.

Many widely adopted benchmarks exhibit strong temporal locality: a large fraction of test edges reconnect node pairs already seen in training, often within short intervals, so predictions are dominated by short-term cues and multiple methods approach saturation. This effect is confined to specific configurations rather than uniform across settings (see Tables 1, 2 and 8), and it is sensitive to the negative sampling strategy: when negatives are drawn structurally or temporally far from positive edges, the decision boundary coarsens and aggregate metrics lose discriminative power. Node-level tasks, which rely more on aggregated temporal and structural context, show no such saturation and retain clear gaps between methods, confirming that the effect is regime-dependent rather than intrinsic to the task.

We further rule out evaluation leakage. All splits are strictly chronological under the DyGLib protocol, so no test-period interaction informs the states used for training; the retentive state is updated causally

from only edges with timestamp $t_e \leq t$, excluding the target edge and any later interaction; and negatives are drawn at the query timestamp, so future co-occurrence cannot inform the score. Under the harder historical and inductive sampling protocols, the saturation partially recedes and clear gaps between methods re-emerge (Tables 1 and 4), as expected from genuine generalization.

C.9 Detailed Analysis of Adaptive Decay Behaviors

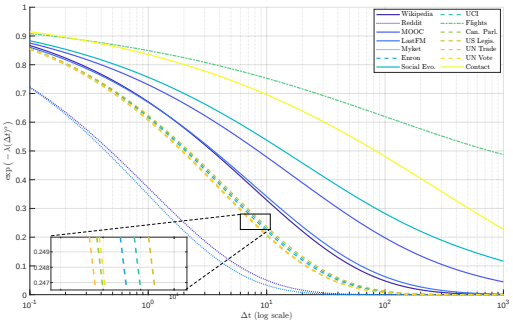


Figure 9: Visualization of learned temporal decay weights $\exp(-\lambda(\Delta t)^{\alpha})$ as a function of time interval Δt for all datasets at the first layer ($\ell = 1$).

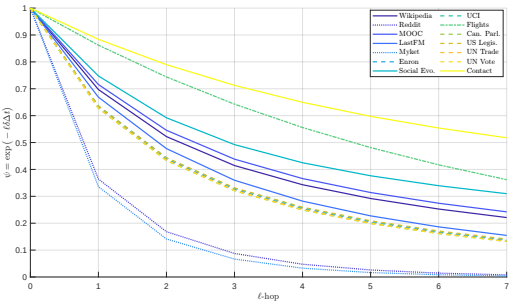


Figure 10: Visualization of learned topological decay weights $\psi = \exp(-\ell\delta^{(\ell)}\Delta t)$ across propagation depths ℓ for all datasets, evaluated at $\Delta t = 100$.

This section provides a comprehensive analysis of the learned decay parameters across all 14 datasets, complementing the summary in Section 4.3. By examining the correlation between learned parameters and dataset properties in Table 3, we reveal how DSRD automatically adapts to diverse dynamic regimes.

Long-term Retention Decay. Figure 3 shows the learned long-term retention decay $\gamma^{(\ell)}$ across state update steps. DTDG datasets with coarse temporal granularity (years) and high scale (>5000), such as UN Trade and US Legis., exhibit strong long-term retention, preserving information across sparse snapshots. Interestingly, some CTDG datasets like Wikipedia also learn strong retention despite fine-grained timestamps, likely reflecting persistent editorial dynamics in social platforms. In contrast, proximity-based graphs with high density (>3.0), such as Contact and Social Evo., learn rapid state turnover, capturing the transient nature of face-to-face interactions. Notably, Flights exhibits fast decay despite being a high-scale DTDG, suggesting that transport-domain connection patterns reset across temporal windows rather than accumulating historical dependencies.

Short-term Temporal Decay. Figure 9 shows the learned short-term decay weights $\exp(-\lambda(\Delta t)^{\alpha})$ across time intervals. High-density proximity graphs such as Contact and Social Evo., along with the coarse-grained Flights dataset, maintain broad temporal sensitivity, benefiting from smoothing over burst-like interaction patterns or extended time windows. Sparse bipartite graphs with scale ≈1.0, such as Reddit and Myket, exhibit the sharpest decay, where individual recent events carry strong predictive signals and older interactions quickly become uninformative. The remaining datasets occupy an intermediate range, reflecting balanced temporal sensitivity across diverse interaction regimes.

Distribution and Stability of λ and α. The learned short-term parameters separate into interpretable regimes that track the inter-event time statistics of each dataset. Sparse bipartite graphs with recency-dominated interactions (e.g., Reddit, Myket) learn a large effective decay rate λ together with a small sensitivity exponent α (i.e., $\alpha \to 0$), which flattens the dependence on the exact elapsed time and sharply suppresses older events, the behavior expected under heavy-tailed inter-event gaps where only the most recent interaction is informative. In contrast, proximity and coarse-grained graphs (e.g., Contact, Social Evo., Flights) learn smaller λ and larger α (closer to 1), producing smoother and more gradual forgetting over extended windows. Across layers, deeper layers tend to learn slower decay, consistent with their role in aggregating higher-order and longer-range context, while the first layer remains the most recency-sensitive.

Topological Decay. Figure 10 presents the learned structural decay weights $\psi = \exp(-\ell\delta^{(\ell)}\Delta t)$ across propagation hops. Dense non-bipartite graphs such as Contact and Social Evo. (density >3.0) maintain broad multi-hop propagation, leveraging rich higher-order connectivity. Flights, despite low overall density, achieves similar broad propagation due to its extremely high scale (15796.27) creating dense within-window structure. Conversely, sparse bipartite graphs including Reddit and Myket (density <0.01) learn sharp topological cutoffs, as higher-order paths are sparse and less informative. Political and economic DTDGs, along with most CTDG datasets, show moderate structural receptive fields aligned with their intermediate density characteristics.

Cross-dimension Correlation. Comparing the above decay dimensions reveals interpretable joint patterns. Sparse bipartite graphs like Reddit and Myket favor locality across all dimensions, while proximity graphs such as Contact and Social Evo. share broad temporal-structural receptive fields but rapid state turnover, suggesting event-driven dynamics where interactions propagate widely but history resets quickly. However, the mapping from graph properties to decay regimes is not fully deterministic: Wikipedia learns atypically strong retention for a CTDG, while Flights behaves unlike other high-scale DTDGs. These exceptions highlight that domain-specific interaction semantics shape the optimal decay regime beyond structural statistics alone.

C.10 Complexity Analysis

Let κ denote the number of sampled temporal neighbors per node, K the number of DSRD layers for multi-hop structural propagation, and $d_h = d/H$ the per-head dimension. For each event, DSRD performs neighbor sampling, second-order state injection, and layered topological diffusion. The per-event computation involves: (i) linear projections for Q and K, V with cost $\mathcal{O}(\kappa d_x d)$, (ii) second-order outer-product injection $K^\top V$ with cost $\mathcal{O}(\kappa H d_h^2)$ over sampled edges, and (iii) gated retentive state update and query contraction with cost $\mathcal{O}(H d_h^2)$ per node. Since topological diffusion is realized through K stacked DSRD layers rather than explicit K-hop neighbor expansion, the total per-event cost is $\mathcal{O}(K \cdot (\kappa d_x d + \kappa d^2/H))$. When $d_x \approx d$, this simplifies to $\mathcal{O}(K\kappa d^2)$. Over the entire event stream, the time complexity is $\mathcal{O}(|\mathcal{E}_t| \cdot K \cdot \kappa \cdot d^2)$. The per-event time complexity scales linearly with $|\mathcal{E}_t|$ and is independent of $|\mathcal{V}|$. However, maintaining the retentive state $S_t^{j,(\ell)} \in \mathbb{R}^{d\times d}$ for all active nodes and layers incurs $\mathcal{O}(|\mathcal{V}|Kd^2)$ space, which we discuss further in Appendix E.

C.11 Scalability

We evaluate the scalability of dynamic graph learning methods across three dimensions: latency per epoch in Figure 2a, throughput (samples processed per second) in Figure 2b, and model size in Figure 2c, each plotted against the average performance rank across datasets. As shown in Figure 2, DSRD achieves a favorable balance between efficiency and accuracy. It ranks among the best-performing models while maintaining moderate latency and high throughput, significantly outperforming heavy architectures such as PINT and TPNet in runtime. Moreover, the compact model size of DSRD ensures lightweight deployment without compromising predictive power. These results confirm the practicality of DSRD in large-scale streaming settings, where both computational efficiency and memory footprint are critical. We also report the scalability analysis of DSRD on synthetic graphs with varying numbers of edges (within one batch) in Table 10, detailing the computational time and memory consumption. The incremental peak GPU memory grows gracefully from 738 MB at 10^4 edges to 1,943 MB at 10^6 edges and 7,849 MB at 10^7 edges.

Isolating the cost of the retentive state. We quantify the memory attributable to storing the second-order state $S_t^{j,(\ell)} \in \mathbb{R}^{d\times d}$. With our default setting ($d = 64$, $K = 2$ layers, and H heads whose per-head states sum to d^2/H per layer), the state of a single node occupies Kd^2 floats, i.e., $2 \times 64^2 \times 4$ bytes ≈ 32 KB in fp32. Since each mini-batch (batch size 200) touches at most a few hundred distinct nodes, the state that is resident on the GPU at any step is only a few hundred $\times$ 32 KB, on the order of 10 MB, which is negligible relative to the 738–7,849 MB peak in Table 10 dominated by activations and sampled-neighbor buffers. The full persistent cache scales as $\mathcal{O}(|\mathcal{V}|Kd^2)$; even for our largest benchmark ($|\mathcal{V}| \approx 10^5$) this is about 3 GB and can reside in host memory, so the per-step GPU footprint stays independent of $|\mathcal{V}|$. This cost can be reduced further by low-rank or factorized parameterizations of $S_t^{j,(\ell)}$, which we leave to future work (Appendix E).

Table 10: Scalability analysis of DSRD on synthetic graphs. Memory denotes the incremental peak GPU memory.

Number of Edges	Forward (ms)	Backward (ms)	Total (ms)	Memory (MB)
10,000	505.62	79.02	584.64	738.04
100,000	3,269.04	1,029.95	4,298.98	1,352.16
1,000,000	16,683.26	8,209.69	24,892.95	1,942.72
10,000,000	89,044.49	70,719.09	159,763.59	7,849.26

C.12 Additional Results of Ablation Study

The additional results of ablation experiments are shown in Figures 11 and 12.

We can observe that removing the entire DSRD block leads to substantial degradation across all datasets, particularly on high-density graphs such as Enron and Social Evo., confirming that the retentive state mechanism is essential for capturing complex interaction patterns. Disabling adaptive temporal decay causes notable drops on discrete-time graphs with coarse granularity (e.g., Flights, US Legis., UN Trade, UN Vote), where high scale values (up to 15859 links per timestep) make distinguishing interaction recency crucial; continuous-time graphs with fine-grained timestamps show smaller sensitivity. Removing topological diffusion primarily affects dense graphs like Social Evo. and Enron, where multi-hop context is informative, while sparse graphs exhibit smaller degradation. Ablating the retentive state yields consistent drops across all settings, with more pronounced effects under inductive evaluation on graphs with longer temporal spans (e.g., UN Vote: 72 years), highlighting the importance of persistent memory for generalization. Overall, each component addresses distinct aspects of dynamic graph learning, and their combination enables robust performance across heterogeneous temporal and structural characteristics.

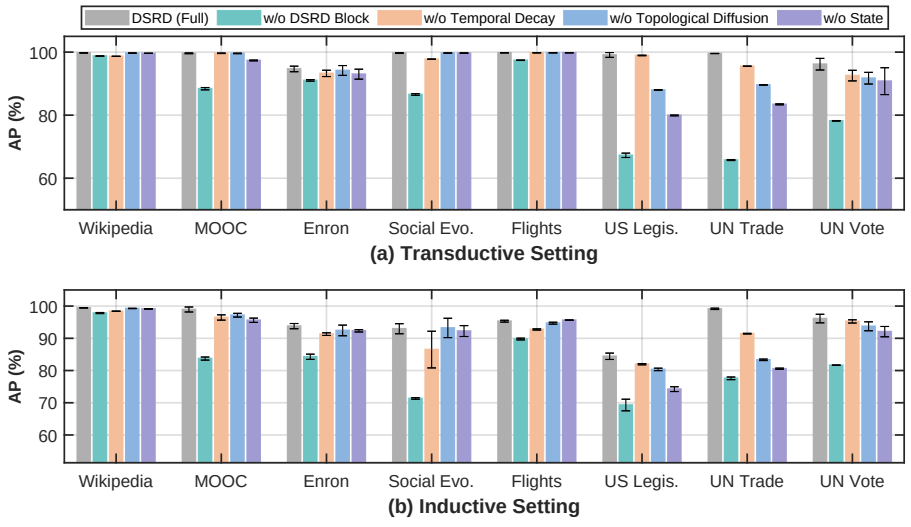


Figure 11: Ablation study on AP (%) under (a) transductive and (b) inductive settings.

D Additional Related Work

D.1 Long-Term Dependency and Memory Mechanisms

Dynamic graph models often incorporate memory modules or recurrent architectures to capture long-term dependencies (Feng et al., 2025; Gravina & Bacciu, 2024; Chang et al., 2026). Early approaches like JODIE maintain coupled recurrent states for user–item interactions, updating each entity's embedding with every event and projecting embeddings along a trajectory (Kumar et al., 2019). Similarly, DyRep models communication events via a temporal point process, using event-

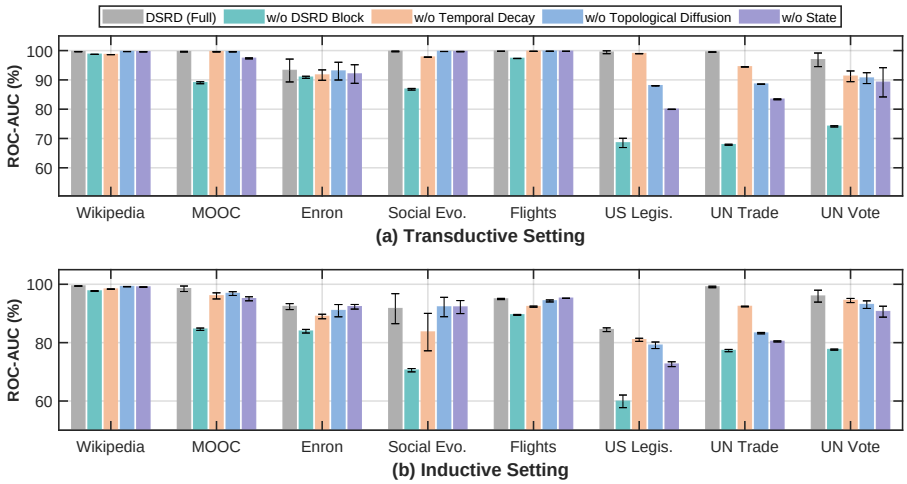


Figure 12: Ablation study on ROC-AUC (%) under (a) transductive and (b) inductive settings.

driven message passing to evolve node states over continuous time (Trivedi et al., 2019). Temporal Graph Networks (TGN) generalize this idea by assigning each node a persistent memory vector that is updated upon neighbor interactions, thereby encoding both short-term and long-term context in the node's history (Rossi et al., 2020). These memory-based frameworks enable retention of past influences beyond immediate neighbors, but they typically operate with a single fixed update scheme or decay rate. As a result, they struggle to adapt to heterogeneous dynamic regimes where different relationships exhibit vastly different temporal patterns (Cong et al., 2023). For example, interactions that are infrequent yet persistent require long-term memory, whereas bursty interactions demand rapid forgetting, a balance that fixed-decay memories cannot achieve. This limitation often forces a trade-off: a model tuned for one timescale may underperform when the temporal dynamics shift, highlighting the need for more adaptive memory mechanisms in dynamic graph learning (Lu et al., 2024). More recently, state space models have been explored as efficient backbones for dynamic graphs (Li et al., 2024), using selective recurrences to capture long-range temporal dependencies. Such architectures substantially improve efficiency on long event sequences, yet their recurrent states typically evolve along the temporal dimension alone, leaving multi-hop structural propagation to separate modules. In contrast, DSRD maintains temporal retention and topological diffusion within a single retentive state.

D.2 Structural Propagation and High-Order Dependencies

Beyond temporal aspects, dynamic graph models must capture structural dependencies that develop over multiple hops (Liu et al., 2024; Lu et al., 2024). Traditional message-passing GNNs propagate along immediate edges, but dynamic scenarios benefit from exploring high-order connections that are not apparent from one-hop interactions (Wang et al., 2022; Yu et al., 2023). Various strategies have been proposed to address this: CAWN performs guided random walks to collect temporal neighborhoods of higher-order contacts (Luo & Li, 2022); DyGFormer leverages self-attention over co-occurrence history to capture long-range dependencies without strict locality constraints (Yu et al., 2023); NAT encodes joint neighbor subgraphs to reflect motifs like triadic closure. Despite these advances, prior approaches often enforce rigid propagation depths or entangle structural and temporal signals in ways that obscure important patterns. Discrete-time methods that aggregate interactions within time windows risk conflating distinct causal orderings, while continuous-time frameworks tend to combine temporal and topological updates, limiting their ability to distinguish complex temporal substructures (Gong et al., 2024; Wang et al., 2021b; Lu et al., 2024). Recent theoretical work further shows that standard temporal GNN architectures fail to capture certain high-order properties and do not strictly improve in expressiveness with deeper stacking (Wałega & Rawson, 2025). These limitations motivate a framework that decouples temporal and structural modeling while adaptively controlling propagation depth.

D.3 Simplification, Scalability, and Generalization Gaps

As dynamic graphs grow in scale and complexity, there is a push towards simplifying models for efficiency and examining their generalization limits (Looks et al., 2017; Cong et al., 2023; Feng et al., 2025). One trend is the development of lightweight architectures that avoid heavy attention or recurrence. GraphMixer exemplifies this by using a simple MLP-Mixer style design with fixed-time positional encodings and token-mixing layers, achieving competitive temporal representations without the cost of graph attention (Veličković et al., 2017) or RNN modules (Cong et al., 2023). Such simplifications not only improve training speed and stability, but also reduce model complexity, which can help mitigate overfitting to idiosyncratic temporal patterns. For instance, scalability-focused methods like NAT introduce specialized data structures (the N-cache) and neighborhood sampling techniques to handle large dynamic graphs. By storing and updating per-node neighbor dictionaries, NAT can construct higher-order features on the fly with minimal overhead, yielding up to an order-of-magnitude speed-up over prior temporal GNNs that rely on exhaustive neighbor searches or repeated random walks (Luo & Li, 2022). Poursafaei et al. (2022) showed that a naively simple method, EdgeBank, which memorizes all observed edges in a lookup table, can outperform many sophisticated models on standard benchmarks. EdgeBank simply predicts a future link if it has been seen historically (and otherwise predicts no link), yet it achieved second-best rankings against state-of-the-art temporal GNNs in certain evaluation settings. This surprising result suggests that many dynamic link prediction benchmarks are dominated by repeated interactions, allowing pure memorization to excel (Poursafaei et al., 2022). Consequently, complex models may inadvertently be learning to memorize historical edges rather than truly extrapolate temporal-structural patterns (Martínez et al., 2016). When evaluation conditions are made more challenging, for example, predicting interactions that have not occurred before or using harder negative samples, the performance of memorization-based methods drops sharply, and models overly reliant on such memory also suffer disproportionately. These observations highlight a generalization gap: methods that perform well by exploiting easy patterns (recurrent edges) may falter on scenarios requiring inductive reasoning or adaptation to new dynamics. Therefore, an emerging theme in recent work is not only to improve scalability and efficiency, but also to design evaluation protocols and models that reward genuine temporal reasoning over simple historical lookup. Bridging this gap remains an open challenge, calling for dynamic graph learners that are both robust across diverse temporal regimes and capable of generalizing beyond the training history.

E Limitations and Future Prospects

Limitations. While DSRD demonstrates strong empirical performance across diverse benchmarks, two limitations warrant discussion. First, the current architecture applies the same learnable decay kernels across all nodes, so heterogeneous graphs with type-specific dynamics (e.g., users vs. items, or different organizational roles) may benefit from type-aware parameterizations. Second, the second-order state incurs $\mathcal{O}(d^2)$ memory per node per layer. In practice this overhead is small (Appendix C.11), and since the state is a sum of rank-one outer products, low-rank or sketched parameterizations can reduce the per-node cost to $\mathcal{O}(rd)$, which we leave to future work.

Future Prospects. Beyond its strong empirical performance, DSRD offers a broadly applicable and extensible foundation for dynamic graph learning. Its unified retentive state and adaptive dual-scale decay make it well suited to high-impact applications such as recommendation systems, fraud detection, and epidemic forecasting, where both temporal responsiveness and multi-hop structure are essential. Several promising directions further broaden its reach. First, the dual-scale framework extends naturally to heterogeneous dynamic graphs through type-specific decay mechanisms, broadening applicability across diverse entity types. Second, integrating explicit periodicity modeling or continuous-time formulations could capture richer temporal patterns. Third, endowing the retentive diffusion with input-dependent (selective) gating in the spirit of modern state-space models (Gu & Dao, 2024) would carry DSRD's efficiency from temporal memory to walk-aware structural propagation. As with other dynamic-graph methods, deployment in sensitive domains should follow appropriate privacy-preserving and ethical practices.