Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082651

ARTICLE

DyG-Hyena: Lightweight Temporal Modeling and Efficient Information Enhancement for Continuous-Time Dynamic Graph

Suchang Yang, Hongtao Yu*, Ruiyang Huang, Huansha Wang, Ran Li and Junzheng Li

Information Engineering University, Science Avenue 62, Zhengzhou, China

*Corresponding Author: Hongtao Yu. Email: yuhongtao202508@163.com

Received: 19 March 2026; Accepted: 15 May 2026; Published: 23 July 2026

ABSTRACT: Modeling dynamic graphs in continuous time is critical for applications such as user behavior prediction and recommendation systems. These models can effectively capture fine-grained and long-term temporal dependencies. However, existing approaches often suffer from high computational costs and optimization difficulties, especially when handling time-sorted neighborhood sequences over long horizons. In this work, we propose DyG-Hyena, a novel continuous-time dynamic graph learning framework that combines conditional variational autoencoder (CVAE)-assisted temporal modeling with efficient feature fusion. Our approach has two main innovations: (i) Efficient temporal fusion—we replace the Transformer with an improved, lightweight Hyena module to model and fuse time-sorted neighborhood feature sequences, reducing computation of this process while maintaining accuracy. A CVAE layer is added before Hyena to capture relative time constraints, enhancing generalization for link prediction. (ii) Task-specific multi-dimensional information enhancement—for link prediction, we incorporate cross-order neighborhood intersection encoding; for node classification, we introduce statistical encoding of node features. Extensive experiments on benchmark dynamic graph datasets demonstrate that DyG-Hyena achieves excellent performance while substantially reducing temporal modeling complexity. Our code is available at https://github.com/yangchang666/DyG-Hyena.

KEYWORDS: Continuous-time dynamic graph; dynamic link prediction; dynamic node classification; Hyena

1 Introduction

Dynamic graph learning adds a time dimension to graph learning in order to capture historical evolution patterns and temporal order [1,2]. It plays a pivotal role in tasks such as social network analysis [3–5], user historical behavior analysis [4], and traffic prediction. To advance this field, various frameworks and unified datasets have been developed for dynamic graph learning [6–9]. These frameworks, combined with unified datasets, have achieved promising performance in dynamic link prediction [6,10–14], dynamic node classification [6,10,15,16], dynamic edge classification [11], and other related tasks.

To achieve fine-grained temporal state perception, researchers have developed continuous-time dynamic graph frameworks (CTDGs) [6,17–21], which are distinct from discrete-time dynamic graph frameworks (DTDGs) that only perform fixed snapshot-based learning. However, these CTDG frameworks face challenges regarding computational efficiency, generalizability, and information utilization. Specifically, models based on RNNs [4,20,22] are prone to gradient explosion when handling long sequences, which leads to suboptimal performance and compromised generalization. For sampling-based aggregation models [12,17,19,23,24] like GraphMixer [19], RepeatMixer [24] that rely on short-term dependencies, their neighbor sampling strategies lack generality, resulting in information loss and reduced performance. Notably,

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the recent FreeDyG [12], which incorporates enhanced Node Interaction Frequency (NIF) encoding, still adopts a short-term sampling-aggregation sequence learning paradigm, resulting in limited performance on datasets with long temporal dependencies. The improved models based on temporal random walk and structural counting, such as NAT [25], PINT [22], and TPNet [26], have achieved promising results. However, they exhibit limitations including random neighbor sampling and high uncertainty. Models based on Transformers [6,27,28], such as DyGFormer [6], have achieved promising results. However, they exhibit limitations including random neighbor sampling and high uncertainty. However, since the computational cost of Transformer-based architectures scales quadratically with sequence length, this results in prohibitive overhead. ScadyG [13] introduces a paradigm for large-scale dynamic graph learning. By discretizing continuous datasets, it requires specific preprocessing steps that are implementation-heavy. DyG-Mamba [10,29] represents a state-of-the-art (SOTA) model with a balance between performance and efficiency; however, it directly adopts DyGFormer's input encoding scheme without further exploiting available information. Therefore, we posit that computational efficiency, information exploitation, and generalization ability are crucial prerequisites for continuous-time dynamic graph learning. In this work, we achieve a balance among these three aspects through two key innovations integrated within a unified framework.

We adopt an improved and simplified version of Hyena for temporal fusion and modeling of time-sorted neighborhood feature sequences. As an efficient sequence processing model, Hyena [30,31] has been widely applied in language modeling and text generation tasks. It achieves enhanced long-sequence processing capabilities with lower computational overhead through long convolutions and data-controlled gating mechanisms. In this work, we leverage Hyena to model temporal features and nonlinear transformations of dynamic graph node neighborhood sequences. This results in a significant reduction in computational overhead for the sequence modeling layer compared to Transformer-based baselines. For link prediction, we further incorporate a CVAE layer [32] before Hyena, which takes temporal encoding as conditional input to model relative temporal constraints. This design facilitates better utilization of temporal information and improves the model's generalization ability.

We design task-specific information enhancement schemes. To fully exploit the information inherent in the data, we customize enhancement strategies for different tasks. For link prediction, we enhance graph structural information by introducing cross-order neighborhood intersection encoding—extending the information aggregation of neighbor node interactions from one-hop to two-hop neighborhoods, which effectively enhances the model's utilization of structural information. For node classification, we supplement statistical encoding of node attributes to enrich feature representation.

We preserve the one-hop full-neighbor truncation and aggregation framework. Our investigation reveals that retaining as much one-hop neighbor information as possible enables the model to maintain strong performance, regardless of dataset granularity or the presence of unseen nodes. Therefore, we adopt DyGFormer's one-hop full-neighbor aggregation framework to enhance the model's generalization ability. To summarize our contributions:

- This work introduce an improved and simplified Hyena module for temporal modeling and capturing dependencies of node neighbor sequence features, which achieves a substantial reduction in computational overhead compared to the Transformer and is well-suited for sequence modeling tasks in continuous-time dynamic graphs.
- Our task-specific information enhancement schemes expand the pool of available information for modeling dynamic graph evolution patterns, thereby improving downstream task performance.
- Extensive experiments demonstrate that DyG-Hyena achieves an effective balance between downstream task performance, computational efficiency, and generalization ability.

2 Related Work

2.1 Continuous Time Dynamic Graph Learning (CTDG)

Dynamic graph learning is typically categorized into discrete-time dynamic graph learning (DTDG) and continuous-time dynamic graph learning (CTDG). DTDG frameworks partition dynamic graph datasets into multiple snapshots at fixed intervals and apply static graph learning methods to each snapshot. This approach struggles to effectively capture inter-snapshot relationships and lacks fine-grained temporal awareness [14,33–37]. In contrast, CTDG frameworks [6,17,38–40], by iteratively modeling dynamic graph evolution, can effectively learn fine-grained patterns and long-term dependencies from dynamic graph datasets. The main differences among existing CTDG models lie in: (i) approaches to acquiring neighborhood information. In CTDG models, the first step is to acquire neighborhood information for each node during each interaction, including one-hop neighbor sampling [17,19,23], multi-hop neighbor sampling [21], and adaptive truncation of all one-hop neighbors tailored to the dataset [6]. (ii) strategies for aggregating neighborhood information. Beyond widely used node and edge features in graph learning, existing baselines also incorporate Neighbor Co-occurrence Encoding (NCE) features [6] and Node Interaction Frequency (NIF) features [12]. (iii) architectures for temporal context learning. Depending on how aggregated neighborhood information is processed, models are based on MLPs [19,24,26], random walk and its improvements [22,25,26], memory networks [4,20,41], Transformers [6,12], Mamba [10,29], etc.

2.2 Hyena for Temporal Fusion and Modeling of Neighborhood Feature Sequences

Hyena [30,31] is an $O(Lk)$ sub-quadratic sequence model (where L denotes the sequence length and k denotes the kernel size) built upon H3 [42], replacing Transformer-based attention mechanisms [43] with implicit long convolutions and data-controlled gating for efficient long-sequence modeling. Geometric Hyena [44] further extends Hyena to geometric graph tasks with SE(3) equivariance, using vector-based geometric long convolutions to capture global geometric context, thereby demonstrating Hyena's versatility across structured data domains.

Dynamic graphs feature time-sorted neighborhood sequences with local temporal relevance, long-sequence interactions, and irregular inter-event intervals. Existing methods [6,10,29] often use block-based truncation or shallow aggregation, losing temporal dependencies or sacrificing efficiency. We tailor Hyena as the temporal fusion and modeling layer for time-sorted neighborhood feature sequences in continuous-time dynamic graphs: replacing its implicit convolution with Conv1d to align with the time axis, leveraging gating to filter heterogeneous temporal features, and stacking two layers (consistent with the baseline) for cumulative long-range temporal modeling.

This customization enables efficient aggregation of time-aware neighborhood information, generating embeddings that integrate structural and temporal dependencies to support time-sensitive link prediction. Unlike general applications of Hyena, we focus on its adaptation to dynamic graph temporal characteristics, balancing long-sequence efficiency and temporal dependency capture which is a direction that is rarely explored in prior work.

Compared to lightweight sequence models for dynamic graphs such as Mamba [29,45] and RWKV [46], Hyena is more adaptable: Mamba's SSM-based selective scanning requires complex discretization steps and is optimized for global long sequences, which aligns poorly with the local structural sequences of time-sorted neighbors that link prediction relies on; RWKV's recurrence suffers from long-sequence gradient attenuation and weak parallelism. Hyena's convolution—gating architecture is directly tied to the time axis, is inherently suited for local structural sequence modeling, enables efficient parallel processing of variable-length neighbor sequences, and incurs lower customization overhead, balancing temporal dependency capture, efficiency, and flexibility with $O(Lk)$ complexity.

2.3 CVAE: Conditional Variational Autoencoder

Conditional Variational Autoencoder (CVAE) [32] is a variant of the variational autoencoder (VAE) that accepts both primary and conditional inputs, models their joint relationship, and generates new data that align with the conditional constraints while respecting the distribution of the primary input. Prior work has shown that CVAE performs well in tasks such as knowledge graph completion. In dynamic graph link prediction, it generates diverse potential link-state scenarios and effectively models conditional variables—thereby improving conditional modeling capacity, accelerating convergence, and enhancing generalization to unseen data.

Compared with simpler conditional mechanisms, such as temporal embeddings and diffusion-based encoders, CVAE is particularly effective for dynamic graph link prediction. Temporal embeddings inject conditional signals (e.g., time) into node features independently, without jointly modeling the conditional constraints and the data distribution. Diffusion-based encoders rely on local diffusion to capture conditional dependencies, which limits flexibility in modeling complex relationships and hinders generalization. By contrast, CVAE explicitly models conditional posterior distributions, integrating conditional signals with the data distribution more precisely to capture dynamic, context-aware link patterns.

3 Preliminary

3.1 Dynamic Graph Learning

A continuous-time dynamic graph is denoted as $G = (V, E)$, where V is the node set ($n = |V|$) and $E = \{(u_i, v_i, t_i) \mid i = 1, \ldots, m\}$ ($m = |E|$) is the edge set with timestamps. In this work, we focus on fine-grained continuous-time dynamic graphs, where timestamps satisfy a *strictly increasing order*:

$$t_i < t_{i+1}, \quad \forall i, \tag{1}$$

meaning *at most one interaction occurs at the same timestamp* for any node pair. Each node and edge is associated with initial features. CTDG learning iteratively aggregates neighborhood information along timestamps to obtain time-aware node embeddings $\mathbf{X}_u^t$ at target time t, which are used for downstream tasks. For clarity, we provide a schematic illustration of the temporal structure in Fig. 1. Key variables are summarized in Appendix A.1. We also define the key variables listed in Appendix A.1.

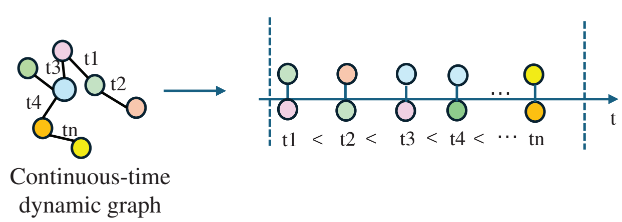


Figure 1: Schematic illustration of the temporal structure in CTDG.

3.2 Downstream Tasks

Our downstream tasks include dynamic link prediction and dynamic node classification. Dynamic link prediction aims to predict whether nodes u and v are (or will be) connected at timestamp t. Using positive–negative sampling, positive samples are observed time-stamped edges, and negative samples are formed by fixing one node (either u or v) and sampling the other from non-interacting nodes at t, we adopt three strategies (random, historical, and inductive [6]), detailed in Appendix C.3. The model distinguishes the two sample types, assigning predicted probabilities close to 1 for positive samples and close to 0 for negative samples, as illustrated in Fig. 2a. Dynamic node classification predicts node u's category at t by learning the mapping between nodes' time-aware embeddings and their category labels, as shown in Fig. 2b.

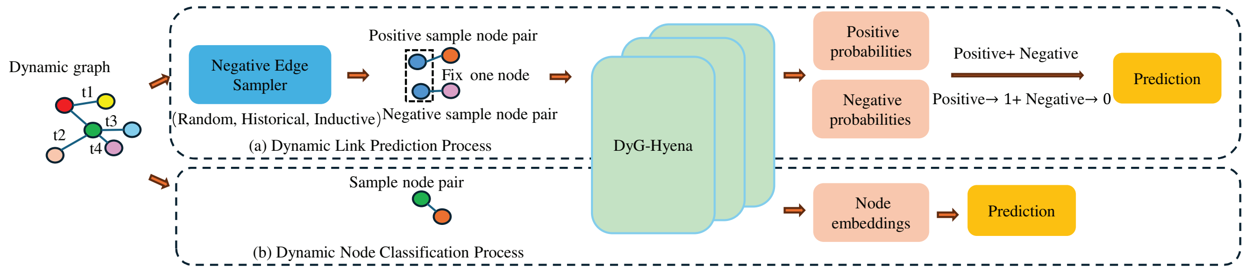


Figure 2: The downstream task framework of dynamic graph: (**a**) for dynamic link prediction and (**b**) for dynamic node classification.

4 Methodology

In this section, we present the design rationale behind the DyG-Hyena model, as illustrated in Fig. 3. For each time-stamped interaction (u, v, t) in the continuous-time dynamic graph, we encode the node features, edge features, and timestamps of all one-hop neighbors of both u and v. For the link prediction task, and in order to leverage the underlying graph structure, we incorporate both neighbor co-occurrence encodings and a cross-stage neighbor intersection encoding scheme. To strengthen temporal information modeling, we introduce a conditional modeling approach based on the CVAE. For the node classification task, we incorporate a statistical node feature encoding. All these encodings are subsequently fed into the Hyena-based temporal fusion and modeling layer, which operates on time-sorted neighborhood feature sequences to generate time-aware embeddings for each node at each timestamp, thereby supporting the downstream tasks.

4.1 Dynamic Graph Feature Encoding

Consistent with [6], we obtain node representations by aggregating one-hop neighbor interaction sequences for each node. Specifically, for interacting nodes u and v in time-stamped interaction (u, v, t), we first retrieve their historical time-stamped one-hop interactions before t. Using hyperparameters, we truncate or pad these sequences to a unified one-hop neighbor sequence length L (a hyperparameter detailed in Appendix C.3) across datasets, yielding interaction sequences $S_u^t = \{(u, a_1, t_{u1}), \ldots, (u, a_L, t_{uL})\}$ and $S_v^t = \{(v, b_1, t_{v1}), \ldots, (v, b_L, t_{vL})\}$ with $t_{u1} < t_{u2} < \ldots < t_{uL} < t$ and $t_{v1} < t_{v2} < \ldots < t_{vL} < t$. We then perform the following encodings on these sequences.

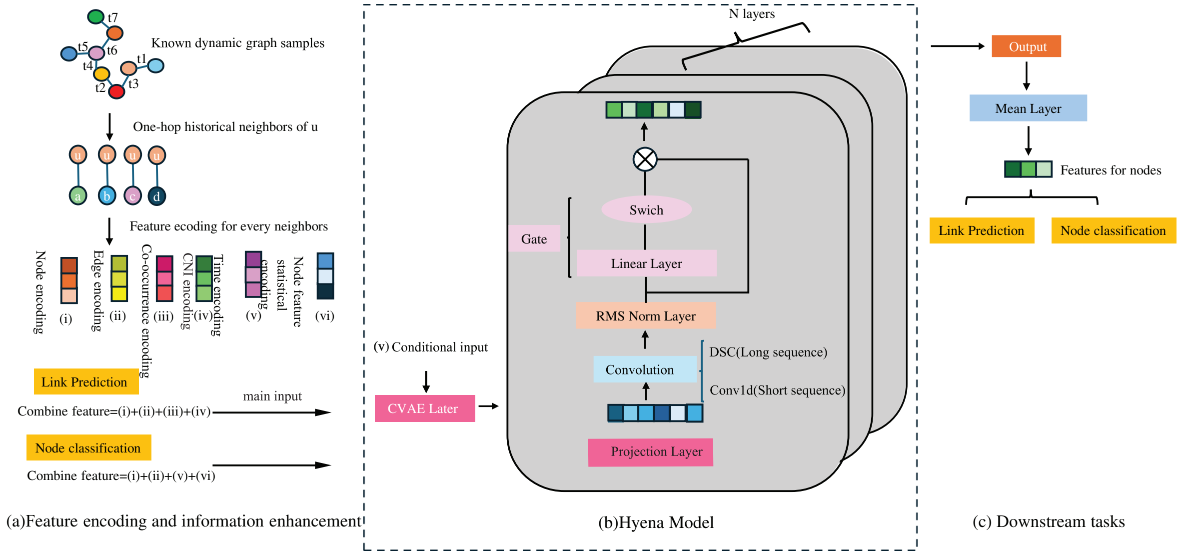


Figure 3: The overall process of DyG-Hyena: (**a**) feature encoding and information enhancement, (**b**) Hyena Model, (**c**) Downstream tasks.

4.1.1 Encodings of Baseline

Similar to the baseline [6], we construct node, edge, time, and neighbor co-occurrence encodings for one-hop neighbor sequences. For node u, its node encoding is $\mathbf{X}_{u,N}^{t} = \{n_{a_1}, \ldots, n_{a_L}\} \in \mathbb{R}^{L \times d_N}$ (d_N: node feature dimension), and edge encoding of its neighbor interactions is $\mathbf{X}_{u,E}^{t} = \{e_{a_1}, \ldots, e_{a_L}\} \in \mathbb{R}^{L \times d_E}$ (d_E: edge feature dimension). If node/edge features are unavailable in the dataset, we pad them with zero vectors uniformly. For brevity, we use node u as a unified example for all subsequent encodings, with the same logic applying to other nodes (e.g., v). For time encoding, we convert absolute timestamps to relative intervals $\Delta t_l = t - t_{ul}$ (target timestamp t, t_{ul}: timestamp of the l-th interaction in S_u^t, $l = 1, \ldots, L$), then encode Δt_l via cosine function to capture temporal periodicity: $\mathbf{X}_{u,T}^{t} = \cos(\Delta t_l \cdot \omega) \in \mathbb{R}^{L \times d_T}$ (d_T: time feature dimension). Here, $\omega = \left\{\alpha_\omega^{-(i-1)/\beta}\right\}_{i=1}^{d_T}$, with α_ω and β as hyperparameters controlling time encoding frequency range, selected to make $\cos(\Delta t_{\max} \cdot \alpha_\omega^{-(i-1)/\beta}) \approx 0$ when $i \to d_T$ ($\Delta t_{\max}$: maximum relative time interval in the dataset). ω is fixed during model training to avoid extreme values.

Consistent with the baseline [6], we also construct neighbor co-occurrence encodings for the one-hop neighbors of interacting nodes u and v. Specifically, we calculate the frequency of each neighbor in S_u^t and S_v^t (within their respective one-hop sequences) as the neighbor co-occurrence feature, denoted as $\mathbf{C}_u^t \in \mathbb{R}^{L \times 2}$ and $\mathbf{C}_v^t \in \mathbb{R}^{L \times 2}$ (with the same logic for v as u, per our unified notation). These encodings are then activated and projected to the target feature dimension via:

$$\mathbf{X}_{u,C}^{t} = f(\mathbf{C}_u^t[:,0]) + f(\mathbf{C}_u^t[:,1]) \in \mathbb{R}^{L \times d_C}, \tag{2}$$

where $f(\cdot)$ projects 1-dimensional frequency features to the d_C-dimensional space, implemented via two ReLU functions followed by a linear layer.

4.1.2 Encodings of Enhanced Information

We introduce cross-stage neighbor intersection (CNI) encoding $\mathbf{X}_{u,G}^{t}$ for link prediction and node feature statistical encoding $\mathbf{X}_{u,M}^{t}$ for node classification.

For link prediction, we argue that graph structural encodings, such as dynamic interaction information, should not be limited to one-hop interactions. Recent research on graph structures has also confirmed this conclusion. Inspired by graph ring structures, we contend that cross-stage neighbor intersections between nodes u and v can also characterize their correlation, with the correlation strength proportional to the intersection size. To enrich graph structural information, we propose CNI encoding. To reduce computational burden, we only compute intersections between two-hop and one-hop neighborhoods: specifically, the intersection of u's two-hop neighbors with v's one-hop neighbors, and vice versa. We adopt a sampling strategy to select two-hop neighbors for intersection counting. The concept of CNI encoding is illustrated in Fig. 4.

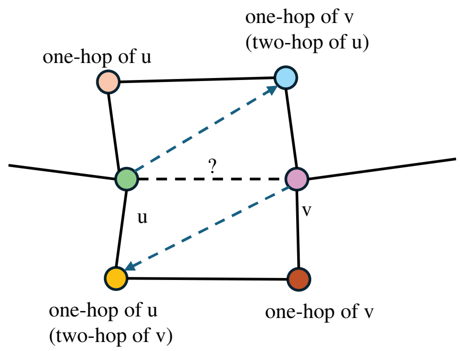


Figure 4: The concept of cross-stage neighbor intersection encoding.

Formally, we construct two-hop neighbor interaction sequences $S_u^{t(2)}$ and $S_v^{t(2)}$ (superscript (2) denotes two-hop, distinguishing from one-hop S_u^t and S_v^t). We truncate or pad these sequences to a unified two-hop neighbor sequence length L_2 (hyperparameter reported in Appendix C.3) across datasets, consistent with one-hop processing. For each neighbor in one-hop S_u^t and S_v^t, we compute its occurrence frequencies in u's two-hop $S_u^{t(2)}$ and v's two-hop $S_v^{t(2)}$ respectively, deriving a 2D vector (consistent with one-hop co-occurrence feature dimension). Stacking these vectors yields cross-stage neighbor intersection (CNI) features of u and v at target timestamp t, denoted as $\mathbf{G}_u^t \in \mathbb{R}^{L\times 2}$ and $\mathbf{G}_v^t \in \mathbb{R}^{L\times 2}$ (L: unified one-hop neighbor sequence length; same logic for v). For example, if $S_u^t = \{a, b, a, c\}$, $S_u^{t(2)} = \{b, a, c, b, b\}$, and $S_v^{t(2)} = \{c, c, b, b, a\}$, the CNI feature for u is $\mathbf{G}_u^t \in \mathbb{R}^{L\times 2} = [[1, 1, 2, 2, 1, 1, 1, 2]]^T$. These features are encoded as:

$$\mathbf{X}_{u,G}^t = f(\mathbf{G}_u^t[:, 0]) + f(\mathbf{G}_u^t[:, 1]) \in \mathbb{R}^{L\times d_G}, \tag{3}$$

where d_G is CNI encoding dimension, and $f(\cdot)$ shares the same implementation as one-hop co-occurrence encoding (two ReLUs + linear layer).

Unlike the high-order neighbor node sampling based on sliding windows in Reaptmixer [24], we obtain the full set of second-order neighbors and adaptively select the first L2 second-order repeated neighbors before the dataset is truncated. This approach, based on empirical adaptation to the dataset, does not cause a significant increase in computational load, while ensuring that no high-order neighbors are missed and avoiding data loss.

For node classification, we argue that node intrinsic properties significantly impact classification performance. However, prior methods only aggregate node, edge, and time encodings, which is insufficient. We therefore compute statistical features of node encodings, specifically the number of non-zero elements, and encode them via the same method as cross-stage neighbor intersection (CNI) encoding, yielding the node feature statistical encoding $\mathbf{X}_{u,M}^t \in \mathbb{R}^{L\times d_M}$.

4.2 CVAE and Graph Structure Enhance Hyena for Link Prediction

For link prediction, we concatenate and project the node, edge, neighbor co-occurrence, and cross-stage neighbor intersection (CNI) encodings of time-sorted one-hop neighbors, ordered by interaction time from earliest to most recent, to match the node feature dimension, yielding the fused encoding $\mathbf{X}_u^t$ that integrates structural information with preliminary temporal order cues:

$$\mathbf{X}_u^t = \begin{bmatrix} \mathbf{X}_{u,N}^t \parallel \mathbf{X}_{u,E}^t \\ \mathbf{X}_{u,C}^t \parallel \mathbf{X}_{u,G}^t \end{bmatrix} \in \mathbb{R}^{L\times(d_N+d_E+d_C+d_G)}, \tag{4}$$

$$\mathbf{X}_u^t = \mathbf{X}_u^t W_{\text{fuse}} + b_{\text{fuse}} \in \mathbb{R}^{L\times d_N} \tag{5}$$

Here, W_{fuse} and b_{fuse} denote the weight and bias of the fusion linear projection layer; L is the unified one-hop neighbor sequence length defined in Section 4.1 (controlling the maximum temporal dependency range); and d_N, d_E, d_C, d_G are the dimensions of the node, edge, co-occurrence, and CNI encodings, respectively. The parameter L adapts to long-sequence efficiency by retaining the most recent L interactions and unifying lengths via padding, enabling efficient modeling of long-term dependencies without the quadratic complexity overhead. This fused encoding retains baseline-effective encodings [6] while incorporating graph-structure-aware CNI encoding.

Unlike the baseline's direct fusion of time features which neglects temporal-structural conditional correlations, we employ CVAE-based conditional variational inference to model relative time constraints. The relative time encoding $\mathbf{X}_{u,T}^t$ (i.e., the time difference between each neighbor interaction and the current time t) is treated as the conditional input, and $\mathbf{X}_u^t$ serves as the main input containing both structural and temporal information. CVAE models the conditional distribution $p(\mathbf{X}_u^t \mid \mathbf{X}_{u,T}^t)$ via a latent variable $Z \in \mathbb{R}^{\text{latent\_dim}}$, where latent_dim (CVAE latent dimension; values listed in Appendix C.3) controls the richness of the temporal–structural joint representation:

$$Z = \mu(\mathbf{X}_{u,T}^t, \mathbf{X}_u^t) + \sigma(\mathbf{X}_{u,T}^t, \mathbf{X}_u^t) \odot \epsilon, \tag{6}$$

$$\hat{\mathbf{X}}_u^t = \text{CVAEDecoder}(Z, \mathbf{X}_{u,T}^t) \tag{7}$$

Here, $\mu(\cdot)$ and $\sigma(\cdot)$ share the same input $\text{Lin}(\text{Cat}(\mathbf{X}_{u,T}^t, \mathbf{X}_u^t))$, where Lin denotes a linear transformation and Cat is concatenation. The function $\sigma(\cdot)$ applies an additional `Softplus` activation to ensure a non-negative standard deviation. The latent dimension latent_dim balances expressive power and generalization (recommended range: latent_dim $\in [d_N/2, 2d_N]$; excessively small values may cause underfitting, whereas overly large values increase overfitting risks). The noise term $\epsilon \sim \mathcal{N}(0, I)$ enhances robustness to irregular time intervals, and the CVAEDecoder, implemented as a two-layer feed-forward network (FFN), reconstructs $\hat{\mathbf{X}}_u^t$ by binding latent representations with $\mathbf{X}_{u,T}^t$. Both the CVAE encoder and decoder are implemented as FFN blocks.

We then employ the Hyena layer as the temporal fusion and modeling module to capture fine-grained neighborhood temporal dependencies, targeting both local temporal relevance and long-sequence efficiency. For fair comparison with the baseline, we stack two Hyena layers, feeding separate inputs for u and v to preserve their individual temporal trajectories which is a critical factor for time-sensitive link prediction. The Hyena layer integrates Conv1d, adaptive gating, and nonlinear transformations to model cumulative temporal dependencies with linear complexity. Details of its implementation are provided in Section 4.4. The final time-aware embedding of node u is given by:

$$\mathbf{Y}_u^t = \text{Hyena}(\hat{\mathbf{X}}_u^t) \tag{8}$$

Here, $\mathbf{Y}_u^t$ and $\mathbf{Y}_v^t$ (for node v) integrate structural information, CVAE-driven conditional temporal constraints, and Hyena-modeled cumulative temporal dependencies, thereby enhancing link prediction by capturing time-sensitive link formation patterns.

4.3 Statistical Encoding Enhance Hyena for Dynamic Node Classification

For dynamic node classification, we omit the CVAE module, as reparameterization-induced "new node generation" is unnecessary for this task and may even adversely affect model performance. Instead, we incorporate relative time encoding directly into the fusion features (without misusing it as a label) and exclude both neighbor co-occurrence encoding and cross-stage neighbor intersection (CNI) encoding unlike link prediction, because these two encodings are likely to introduce noise into the fused representation of a single node. As shown in Appendix D.3, they yield no performance gain for node classification.

We further introduce a node feature statistical encoding, denoted $\mathbf{X}_{u,M}^t$. Specifically, the fused encoding of node u is defined as:

$$\mathbf{X}_u^t = \begin{bmatrix} \mathbf{X}_{u,N}^t \parallel \mathbf{X}_{u,E}^t \\ \mathbf{X}_{u,T}^t \parallel \mathbf{X}_{u,M}^t \end{bmatrix} \in \mathbb{R}^{L\times(d_N+d_E+d_T+d_M)}, \tag{9}$$

$$\mathbf{X}_u^t = \mathbf{X}_u^t W_{\text{fuse-nodecls}} + b_{\text{fuse-nodecls}} \in \mathbb{R}^{L\times d_N} \tag{10}$$

Here, $W_{\text{fuse-nodecls}}$ and $b_{\text{fuse-nodecls}}$ denote the weight and bias of the fusion projection layer used for node classification, with the subscript "fuse-nodecls" distinguishing them from the corresponding parameters in link prediction (W_{fuse} and b_{fuse}).

The fused encoding is fed directly into stacked Hyena layers, identical in design to those used in link prediction, as detailed in Section 4.4, to derive time-aware node embeddings for binary classification. The final embedding of node u is given by:

$$\mathbf{Y}_u^t = \text{Hyena}(\mathbf{X}_u^t) \tag{11}$$

Finally, each node's time-aware embedding $\mathbf{Y}_u^t$ is passed to the downstream classification module to produce the final binary classification output (the same process applies to all other nodes).

4.4 Detailed Implementation of Hyena Temporal Fusion and Modeling Layer

As described in Sections 4.2 and 4.3, the neighbor feature sequences are input to the Hyena layer for feature fusion and temporal modeling. The specific architecture and discussion are provided below.

4.4.1 Concise Computation Process

We propose an improved Hyena layer with adaptive convolution selection (based on L) to balance performance and efficiency, retaining two stacked layers, consistent with the baseline, for fair comparison. Given input encodings ($\hat{\mathbf{X}}_u^t$ for link prediction and $\mathbf{X}_u^t$ for node classification), we perform time-aligned convolution with adaptive operator selection:

$$\hat{\mathbf{Y}}_u^t = \begin{cases} \text{Conv1d}(\hat{\mathbf{X}}_u^t) & L \leq 256, \\ \text{DSC}(\hat{\mathbf{X}}_u^t) & L > 256, \end{cases} \tag{12}$$

Here, k denotes the convolution kernel size (candidate values listed in Appendix C.3), and DSC refers to depthwise separable Conv1d, comprising depthwise convolution for temporal dependency modeling and pointwise convolution for channel fusion. The subsequent operations are:

$$\hat{\mathbf{Y}}_u^t = \mathrm{RMSNorm}(\hat{\mathbf{Y}}_u^t), \tag{13}$$

$$\mathrm{Gate} = \mathrm{Swish}(W_1\hat{\mathbf{Y}}_u^t + b_1), \tag{14}$$

$$\hat{\mathbf{Y}}_u^t = \hat{\mathbf{Y}}_u^t \odot \mathrm{Gate}, \tag{15}$$

$$\mathbf{Y}_u^t = \mathrm{ReLU}(\hat{\mathbf{Y}}_u^t W_2 + b_2) W_3 + b_3 \tag{16}$$

Here, $W_1, b_1, W_2, b_2, W_3, b_3$ are trainable parameters, and d_{hidden} (the FFN hidden dimension) is specified in Appendix C.3. Stacking two layers extends the effective temporal dependency range to $2k$ (linear complexity), while separate inputs for u and v preserve their unique temporal trajectories, which is critical for time-sensitive dynamic graph tasks.

4.4.2 Adaptive Convolution Selection for Datasets with Varying L

To accommodate datasets with differing time-sorted neighborhood sequence lengths L, we explicitly provide two targeted convolution selection schemes in the improved Hyena layer. For short sequences ($L \leq 256$), standard Conv1d is adopted, as it is suitable for scenarios where local temporal dependencies are concentrated; under a fixed d_N, its computational overhead is manageable, enabling integration of temporal correlations and channel interactions without significant efficiency loss. For long sequences ($L > 256$), depthwise separable Conv1d (DSC) is selected, designed to control computational scale for long sequences (especially ultra-long sequences such as $L = 2048$). Its split design isolates temporal dependency propagation from channel fusion, mitigating long-sequence feature drift while retaining modeling accuracy.

4.4.3 Computational Complexity Analysis

Both variants of the improved Hyena layer maintain linear complexity $O(L)$ (scaling with sequence length L), in contrast to the baseline Transformer's quadratic $O(L^2)$. For $L \leq 256$, the Conv1d-based Hyena outperforms the Transformer by avoiding redundant self-attention computations. For $L > 256$, the DSC-based Hyena further improves efficiency by eliminating channel–temporal coupling redundancy, thereby avoiding the computational explosion seen in Transformers for ultra-long sequences. These results confirm that the adaptive scheme provides superior efficiency compared to the baseline across varying sequence lengths, while preserving temporal modeling accuracy.

5 Experiments

In this section, we perform experiments on DyG-Hyena to demonstrate its superior performance and analyze the role of each component in the Hyena layer.

5.1 Experimental Settings

5.1.1 Datasets and Baselines

We adopt 12 mainstream datasets for continuous-time dynamic graph (CTDG) learning, covering social and communication network scenarios. Seven datasets are reported in the main text, while the other five are placed only in the appendix. These excluded datasets contain massive simultaneous timestamps and coarse-grained temporal features, which are incompatible with the fine-grained modeling paradigm of general CTDG frameworks. All datasets are split into training, validation and test sets at a ratio of 70%/15%/15%, with detailed statistics illustrated in Appendix C.1. For fair comparison, we combine classic and recent

SOTA baselines. Classical methods cover RNN-based models (JODIE, DyRep, TGN, CAWN), memory-based EdgeBank, sampling aggregation-based methods (TGAT, TCL, GraphMixer), and Transformer-based DyGFormer. Recent SOTA competitors include FreeDyG, DyG-Mamba, TPNet, RepeatMixer and ScaDyG. We conduct comparisons even if partial SOTA methods lack results on individual datasets, and the detailed baseline introductions are presented in Appendix C.2.

5.1.2 Evaluation Configurations and Metrics

Consistent with most prior continuous-time dynamic graph learning works, we adopt the unified library and experimental setup from [6] for downstream task evaluation and baseline reproduction. For link prediction, we use two evaluation settings: transductive and inductive. The transductive setting focuses on future link prediction for nodes observed during training, while the inductive setting targets future link prediction for nodes unseen during training. We use Average Precision (AP) and ROC-Area Under the Curve (ROC-AUC) as evaluation metrics. Following [6,7], we employ three negative sampling strategies: random (rnd), historical (hist), and inductive (ind), where each strategy generates distinct types of negative samples to comprehensively evaluate model performance. For node classification, we use ROC-AUC to assess classification performance.

5.1.3 Experimental Details

We perform experiments using the unified library Dyclib [6], ensuring fair comparison against baselines under a consistent evaluation framework. Consistent with prior works, DyG-Hyena is trained with the Adam optimizer, using a learning rate of 10^{-4}, a maximum of 100 epochs, and a batch size of 200. We run five independent training trials with random seeds ranging from 0 to 4 to mitigate potential bias. An early stopping mechanism is employed for each trial with a patience of 20. Detailed implementation and evaluation configurations are provided in Appendix C.3.

5.2 Main Performance Comparison

5.2.1 Performance of Downstream Tasks

We report the Average Precision (AP) scores for transductive dynamic link prediction under the random negative sampling strategy in Table 1, and the AP scores for inductive dynamic link prediction under the same sampling strategy in Table 2, all based on 7 commonly used datasets for CTDG baselines. For baseline results, we present the outcomes of our reproduced FreeDyG [6], DyGFormer [6], and TPNet [26], as well as other previously reported baseline results; our reproduced results exhibit negligible differences from those reported in the original papers. It is worth noting that we cannot reproduce the results of ScaDyG [13], DyG-Mamba [10,29], and RepeatMixer [24], as their code is either unpublished or incomplete. Thus, we directly adopt the results from their original papers, which are obtained under the same experimental benchmarks as ours. We note that the DyG-Mamba results cited herein are from [29], given the consistency between the results reported in the two referenced works. As documented in their studies, the dataset splitting and negative sampling strategy settings are consistent with ours.

Table 1: AP for transductive dynamic link prediction with random negative sampling strategies. Bold red values denote the best performance, blue underlined values represent the second-best performance, and N/A means unavailable data or unused method.

Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
Wikipedia	96.50 ± 0.14	94.86 ± 0.06	96.94 ± 0.06	98.45 ± 0.06	98.76 ± 0.03	90.37 ± 0.00	96.47 ± 0.16	97.25 ± 0.03	99.03 ± 0.02	99.15 ± 0.02	<u>99.32 ± 0.03</u>	99.16 ± 0.02	99.26 ± 0.01	N/A	**99.38 ± 0.01** (+0.06%)
Reddit	98.31 ± 0.14	98.22 ± 0.04	98.52 ± 0.02	98.63 ± 0.06	99.11 ± 0.01	94.86 ± 0.00	97.53 ± 0.02	97.31 ± 0.01	99.22 ± 0.01	99.32 ± 0.01	99.27 ± 0.00	99.22 ± 0.01	**99.48 ± 0.01**	N/A	<u>99.33 ± 0.01 (−0.15%)</u>
MOOC	80.23 ± 2.44	81.97 ± 0.49	85.84 ± 0.15	89.15 ± 1.60	80.15 ± 0.25	57.97 ± 0.00	82.38 ± 0.24	82.78 ± 0.15	87.52 ± 0.49	89.21 ± 0.08	<u>96.39 ± 0.09</u>	92.76 ± 0.10	89.61 ± 0.19	<u>90.10 ± 1.20</u>	**96.92 ± 0.33** (+0.55%)
LastFM	70.85 ± 2.13	71.92 ± 2.21	73.42 ± 0.21	77.07 ± 3.97	86.99 ± 0.06	79.29 ± 0.00	67.27 ± 2.16	75.61 ± 0.24	93.00 ± 0.12	93.35 ± 0.20	<u>94.50 ± 0.08</u>	94.14 ± 0.06	92.15 ± 0.16	N/A	**95.58 ± 0.01** (+1.14%)
Enron	84.77 ± 0.30	82.38 ± 3.36	71.12 ± 0.97	86.53 ± 1.11	89.56 ± 0.09	83.53 ± 0.00	79.70 ± 0.71	82.25 ± 0.16	92.47 ± 0.12	92.65 ± 0.12	**92.90 ± 0.17**	92.66 ± 0.07	92.51 ± 0.05	N/A	<u>92.72 ± 0.11 (−0.19%)</u>
Social Evo.	89.89 ± 0.55	88.87 ± 0.30	93.16 ± 0.17	93.57 ± 0.17	84.96 ± 0.09	74.95 ± 0.00	93.13 ± 0.16	93.37 ± 0.07	94.73 ± 0.01	94.77 ± 0.01	94.73 ± 0.02	N/A	<u>94.91 ± 0.01</u>	N/A	**94.97 ± 0.01** (+0.06%)
UCI	89.43 ± 1.09	65.14 ± 2.30	79.63 ± 0.70	92.34 ± 1.04	95.18 ± 0.06	76.20 ± 0.00	89.57 ± 1.63	93.25 ± 0.57	95.79 ± 0.17	95.91 ± 0.15	<u>97.35 ± 0.04</u>	96.74 ± 0.08	96.28 ± 0.11	95.60 ± 0.20	**97.44 ± 0.03** (+0.09%)
Avg. Rank	11.14	12.14	10.57	8.00	8.86	12.57	11.71	10.00	5.71	4.29	<u>2.43</u>	N/A	3.71	N/A	**1.29**

Table 2: AP for inductive dynamic link prediction with random negative sampling strategies. Bold red values denote the best performance, blue underlined values represent the second-best performance, and N/A means unavailable data or unused method.

Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
Wikipedia	94.82 ± 0.20	92.43 ± 0.37	96.22 ± 0.07	97.83 ± 0.04	98.24 ± 0.03	N/A	96.22 ± 0.17	96.65 ± 0.02	98.59 ± 0.03	98.77 ± 0.03	98.91 ± 0.01	98.70 ± 0.05	<u>98.97 ± 0.01</u>	N/A	**99.05 ± 0.01** (+0.08%)
Reddit	96.50 ± 0.13	96.09 ± 0.11	97.09 ± 0.04	97.50 ± 0.07	98.62 ± 0.01	N/A	94.09 ± 0.07	95.26 ± 0.02	98.84 ± 0.02	<u>98.97 ± 0.01</u>	98.86 ± 0.01	98.85 ± 0.01	98.91 ± 0.01	N/A	**99.01 ± 0.01** (+0.04%)
MOOC	79.63 ± 1.92	81.07 ± 0.44	85.50 ± 0.19	89.04 ± 1.17	81.42 ± 0.24	N/A	80.60 ± 0.22	81.41 ± 0.21	86.96 ± 0.43	88.64 ± 0.08	<u>95.07 ± 0.26</u>	93.05 ± 0.12	87.75 ± 0.62	N/A	**95.96 ± 0.46** (+0.94%)
LastFM	81.61 ± 3.82	83.02 ± 1.48	78.63 ± 0.31	81.45 ± 4.29	89.42 ± 0.07	N/A	73.53 ± 1.66	82.11 ± 0.42	94.23 ± 0.09	94.42 ± 0.21	<u>95.36 ± 0.11</u>	94.98 ± 0.12	94.89 ± 0.01	N/A	**96.41 ± 0.01** (+1.10%)
Enron	80.72 ± 1.39	74.55 ± 3.95	67.05 ± 1.51	77.94 ± 1.02	86.35 ± 0.51	N/A	76.14 ± 0.79	75.88 ± 0.48	89.76 ± 0.34	89.67 ± 0.27	<u>90.34 ± 0.28</u>	87.97 ± 0.29	89.69 ± 0.17	N/A	**90.56 ± 0.27** (+0.24%)
Social Evo.	91.96 ± 0.48	90.04 ± 0.47	91.41 ± 0.16	90.77 ± 0.86	79.94 ± 0.18	N/A	91.55 ± 0.09	91.86 ± 0.06	93.14 ± 0.04	93.13 ± 0.05	93.24 ± 0.07	N/A	**94.76 ± 0.05**	N/A	<u>93.44 ± 0.01 (−1.39%)</u>
UCI	79.86 ± 1.48	57.48 ± 1.87	79.54 ± 0.48	88.12 ± 2.05	92.73 ± 0.06	N/A	87.36 ± 2.03	91.19 ± 0.42	94.54 ± 0.12	94.76 ± 0.19	<u>95.74 ± 0.05</u>	95.04 ± 0.12	94.85 ± 0.10	N/A	**95.85 ± 0.23** (+0.11%)
Avg. Rank	10.29	11.71	11.00	8.43	7.57	N/A	10.86	9.14	4.71	4.43	<u>2.29</u>	N/A	3.14	N/A	**1.14**

Additional results including those for historical and inductive negative sampling (both transductive and inductive settings), ROC-AUC scores for transductive/inductive dynamic link prediction, and results on an additional 5 commonly used datasets for CTDG baselines which are provided in Appendices D.1 and D.2. These 5 datasets are not reported in the main text as they lack continuous partitioning, contain numerous simultaneous interactions, and are thus unsuitable for CTDG evaluation; we include their results to align with baselines. Since EdgeBank is only applicable to the transductive setting, and DyG-Mamba, FreeDyG, and ScaDyG are only implemented on partial datasets, their full performance is not reported herein, and "N/A" is marked where results are unavailable.

We also present DyG-Hyena's performance on dynamic node classification in Table 3, where it achieves the second-best performance on the Wikipedia and Reddit datasets.

Table 3: ROC-AUC for dynamic node classification. Best in **red**, second in **blue**.

Methods	Wikipedia	Reddit	Avg. Rank
JODIE	**88.99 ± 1.05**	60.37 ± 2.58	5.5
DyRep	86.39 ± 0.98	63.72 ± 1.32	7
TGAT	84.09 ± 1.27	70.04 ± 1.09	6
TGN	86.38 ± 2.34	63.27 ± 0.90	8
CAWN	84.88 ± 1.33	66.34 ± 1.78	8
EdgeBank	N/A	N/A	N/A
TCL	77.83 ± 2.13	68.87 ± 2.15	7.5
GraphMixer	86.80 ± 0.79	64.22 ± 3.32	5.5
DyGFormer	87.44 ± 1.08	68.00 ± 1.74	4
DyG-Mamba	88.58 ± 0.92	**70.79 ± 1.97**	**2.00**
DyG-Hyena	**88.67 ± 0.52 (−0.32%)**	**71.12 ± 2.15 (+0.33%)**	**1.50**

5.2.2 Performance Analysis

Based on the above results, DyG-Hyena delivers overall superior performance on CTDG-adapted fine-grained datasets.

First, for dynamic link prediction, DyG-Hyena ranks first on five and six datasets respectively in transductive and inductive settings, and ranks second best on the remaining datasets. Meanwhile, it maintains strong performance under other negative sampling strategies and the remaining five datasets presented in the appendix, with an overall average ranking that achieves either the best or the second best. For dynamic node classification, it ranks first or second on two labeled datasets and achieves the first average ranking. This superiority stems from three key factors: (i) Adopting the Hyena-based temporal fusion and modeling layer for time-sorted neighborhood feature sequences inherently ensures strong temporal modeling capability and thus superior performance; (ii) The CVAE module effectively models the dependencies between conditional inputs (temporal encodings) and primary inputs (fused encodings), enhancing the utilization of temporal information; (iii) The proposed cross-stage neighbor intersection (CNI) encoding enriches the utilization of graph structural information.

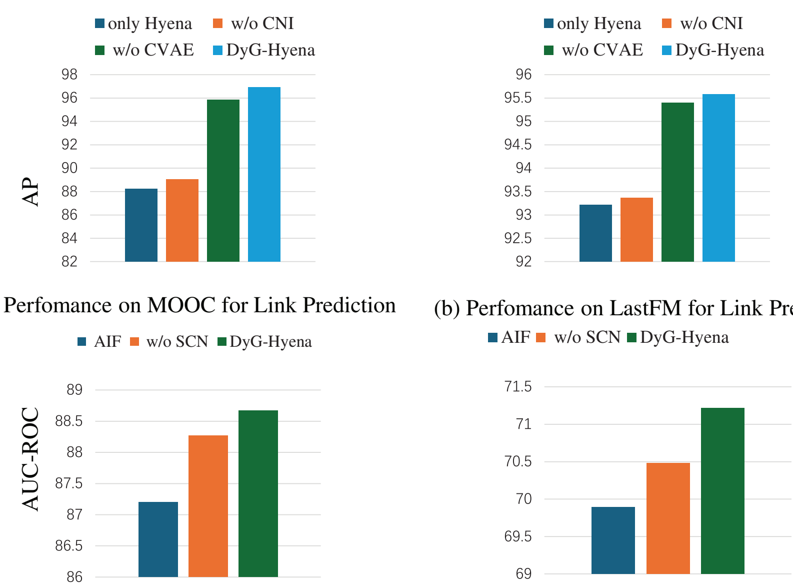
Second, DyG-Hyena possesses robust generalization ability. In link prediction, it achieves the same performance ranking in the inductive setting (for unseen nodes during training) as in the transductive setting (for seen nodes), and exhibits stable performance across datasets with varying temporal granularities. This strong generalization can be attributed to: (i) The CVAE module generates diverse potential link states/relationships via the reparameterization trick, enabling the model to generalize better to unseen data; (ii) The one-hop full-neighbor truncation-padded aggregation framework facilitates the capture of richer neighborhood information. We quantitatively analyze DyG-Hyena's generalization ability in Appendix D.4.

Notably, DyG-Hyena fails to achieve top-two performance on some of the five additional datasets in the appendix (e.g., USLegis, UNTrade), as detailed in Appendices D.1 and D.2. This stems from incompatibility with the proposed continuous-time dynamic graph (CTDG) framework: these datasets contain numerous simultaneous interactions (i.e., same timestamp), while the CTDG framework processes them sequentially. We include these datasets for baseline consistency and provide a detailed analysis of this limitation in Appendix A.2.

Subsequently, we elaborate on the advantages of each key module in DyG-Hyena.

5.3 Ablation Study

We verify the effectiveness of three key components in DyG-Hyena through ablation experiments: the Hyena layer, CVAE-based conditional modeling of the temporal dimension, and the proposed enhanced information encodings. Specifically, we perform ablation experiments for link prediction on the MOOC and LastFM datasets, and for dynamic node classification on the Wikipedia and Reddit datasets; the corresponding results are visualized as bar charts in Fig. 5. Additional ablation experiments with more datasets and comprehensive analyses are provided in Appendix D.3. The ablation results demonstrate that removing any component of DyG-Hyena leads to noticeable performance degradation, verifying that the Hyena layer inherently delivers strong performance and all proposed enhanced encodings contribute effectively to the model's overall capability.



(a) Perfomance on MOOC for Link Prediction (b) Perfomance on LastFM for Link Prediction

(c) Perfomance on Wikipedia for Node Classification (d) Perfomance on Reddit for Node Classification

Figure 5: Illustration of the ablation experiment.

5.4 Advantages of Using Hyena

We replaced the Transformer with the Hyena layer as the temporal fusion and modeling layer for time-sorted neighborhood feature sequences, reducing the computational load of temporal modeling and feature extraction while preserving more sequence information. To fairly validate the advantages of DyG-Hyena, we conducted two key comparisons:

First, we compared DyG-Hyena (and its variants) with DyGFormer under identical settings (e.g., no concatenation technique for DyGFormer). Evaluations across multiple datasets and varying numbers of neighboring nodes (64–2048, using the Can.Parl dataset to avoid GPU memory constraints) show that DyG-Hyena achieves significantly lower GPU memory usage and shorter training time than DyGFormer, even with our enhancement modules, the resulting increase in training time is acceptable (see Appendices D.3 and D.5).

Second, we compared different candidate layers for the temporal fusion and modeling module within the DyG-Hyena framework (retaining all our information enhancement and CVAE components, distinct from the technical routes of DyG-Mamba, DyGFormer, etc.). The compared layers include Mamba [45], Transformer, and RWKV [46] (baselines introduced in Chapter 2), with results shown in Table 4. Specifically: RWKV achieves the highest computational efficiency at the cost of obvious accuracy loss. Transformer provides stable performance but brings exponentially rising computational overhead with longer sequences. Both Mamba and Hyena are lightweight and perform well in general. Hyena maintains competitive overall results and exhibits unique strengths on short-sequence datasets. Benefiting from the convolution-gating design, Hyena mitigates gradient attenuation and discretization errors, making it well-suited for dynamic link prediction with local neighbor dependence and short-range sequence modeling.

Table 4: Performance and efficiency comparison of different temporal fusion and modeling layers for neighborhood feature sequences (transductive setting, random negative sampling, 100 epochs) †: Best, ‡: Second-Best.

Dataset	Model (Temporal Modeling Layer)	AP for Transductive	GPU Memory (MB)	Time per Epoch
Wikipedia	Transformer	99.23‡	5860	1 min 32 s
	Mamba	99.29	2856	1 min 33 s
	RWKV	98.91	2172†	1 min 20 s†
	Hyena	99.38†	4218‡	1 min 27 s‡
UCI	Transformer	96.78‡	2740	39 s
	Mamba	96.87	2038	40 s
	RWKV	95.11	1322†	30 s†
	Hyena	97.44†	1996‡	36 s‡
MOOC	Transformer	96.48	19,510	8 min 19 s
	Mamba	96.98†	10,232‡	7 min 49 s
	RWKV	95.27	7568†	6 min 29 s†
	Hyena	96.92‡	13,724	7 min 34 s‡
Reddit	Transformer	99.40†	4580	9 min 23 s
	Mamba	99.27	3090‡	9 min 1 s
	RWKV	99.02	2518†	7 min 58 s†
	Hyena	99.33‡	3382	8 min 49 s‡

5.5 The Role of CVAE

In this section, we explored the effects of time information enhancement on link prediction. To verify the advantages of adding a CVAE layer for time modeling (DyG-Hyena (CVAE)) over the temporal embedding concatenation mechanism (DyG-Hyena (TE)), improving convergence speed and transductive AP, we compared the two models on 4 datasets. We defined convergence as test performance reaching 99% of the optimal value. Table 5 records their convergence rounds and single-epoch time. Results show DyG-Hyena (CVAE) converges faster with acceptable single-epoch time increase, and outperforms DyG-Hyena (TE) in transductive AP, benefiting from CVAE's probabilistic framework for precise integration of temporal and structural information. Ablation experiments (Section 5.3) and generalization evaluations (Appendix D.4) further confirm the CVAE layer enhances temporal modeling, boosts transductive AP, and improves generalization on unseen data.

Table 5: Convergence metrics and transductive AP in DyG-Hyena(CVAE) and DyG-Hyena(TE).

Datasets	DyG-Hyena (CVAE)				DyG-Hyena (TE)			
	Conv. Epochs	Time per Epoch	Total Time	AP	Conv. Epochs	Time Per Epoch	Total Time	AP
Wikipedia	2	1 min 27 s	2 min 54 s	99.38	4	1 min 25 s	5 min 40 s	99.24
MOOC	40	7 min 34 s	302 min 40 s	96.92	50	7 min 23 s	369 min 10 s	95.78
UCI	10	36 s	6 min 00 s	97.44	20	35 s	11 min 40 s	96.12
Can. Parl.	30	4 min 30 s	135 min 00 s	98.99	40	4 min 03 s	162 min 00 s	96.62

6 Conclusion

This paper proposes a dynamic graph learning model based on efficient convolution, namely DyG-Hyena. Our method adopts the Hyena layer as the temporal modeling backbone for sequential features to capture temporal dependencies of node/edge interaction sequences, which significantly reduces computational complexity. We further design task-specific information enhancement strategies: for dynamic link prediction, we propose CVAE-based temporal information enhancement and graph structure-aware cross-stage neighbor intersection (CNI) encoding to enrich task-relevant information; for dynamic node classification, we introduce node feature statistical encoding to enhance the modeling of node intrinsic properties. The effectiveness of DyG-Hyena is validated through extensive experiments on multiple datasets. In future work, we plan to further enhance DyG-Hyena to better handle dynamic graph datasets with varying temporal granularities and explore additional information enhancement modules to further boost its performance.

Acknowledgement: The authors would like to thank the Key R&D Special Project of Henan Province (Grant No. 231111211000) for funding support.

Funding Statement: This work was supported by the Key R&D Special Project of Henan Province (Grant No. 231111211000).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Suchang Yang, Hongtao Yu and Ruiyang Huang; methodology, Suchang Yang, Hongtao Yu and Ruiyang Huang; software, Suchang Yang; validation, Suchang Yang, Hongtao Yu, Ruiyang Huang, Huansha Wang and Junzheng Li; formal analysis, Suchang Yang and Junzheng Li; investigation, Suchang Yang and Junzheng Li; resources, Hongtao Yu; data curation, Suchang Yang, Ran Li and Junzheng Li; writing—original draft preparation, Suchang Yang; writing—review and editing, Suchang Yang, Hongtao Yu, Ruiyang Huang and Ran Li; visualization, Huansha Wang; supervision, Hongtao Yu and Ruiyang Huang; project administration, Hongtao Yu; funding acquisition, Hongtao Yu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository.

Ethics Approval: Not applicable

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A Additional Theoretical Details Discussion

Appendix A.1 Important Symbol Definitions

In this section, we define the key symbols used in this paper to facilitate better understanding, as summarized in Table A1.

Table A1: Important symbol definitions.

Symbol	Definition
N, E, T	Abbreviations for Node, Edge, and Time (used in encoding subscripts)
$\mathbf{X}_u^t/\mathbf{X}_v^t$	Time-aware embedding of node u/v at target timestamp t
S_u^t/S_v^t	Time-series sequence of historical one-hop neighbor interactions of u/v (truncated/padded via hyperparameters)
$\mathbf{X}_{u,N}^t$	Node encoding (feature of one-hop neighbors of node u)
$\mathbf{X}_{u,E}^t$	Edge encoding (edge feature of one-hop neighbor interactions of node u)
$\mathbf{X}_{u,T}^t$	Time encoding (cosine-encoded relative time interval for temporal periodicity)
$\mathbf{X}_{u,C}^t$	Neighbor co-occurrence encoding (co-occurrence frequency of one-hop neighbors of u and v)
$\mathbf{X}_{u,G}^t$	Cross-stage neighbor intersection (CNI) encoding (intersection frequency between two-hop and one-hop neighbors)
$\mathbf{X}_{u,M}^t$	Node feature statistical encoding (statistical feature of node intrinsic properties, e.g., non-zero element count)

Appendix A.2 Limitations

The limitations of our approach mainly lie in the following aspects: (i) We fail to achieve breakthrough performance on coarse-grained datasets, as most interactions in these datasets are concentrated at identical timestamps. This contradicts the sequential interaction learning paradigm in continuous-time dynamic graph (CTDG) learning, which inherently assumes ordered learning of interaction sequences based on temporal order, yet interactions sharing the same timestamp do not follow the model's predefined sequential order. (ii) After introducing the cross-stage neighbor intersection (CNI) encoding, the model's computational complexity increases noticeably, which conflicts with the core goal of reducing computational overhead. Although we adopted a neighbor sampling strategy to mitigate this issue, it may lead to the loss of certain useful information. (iii) We did not incorporate interaction-related features for node classification, yet this does not imply that interaction information is irrelevant to node-level tasks. We believe that with rational design, interaction information can contribute effectively to node-level modeling, and this will be a focus of our future research.

Appendix A.3 Intuitive Reasons for Advantages

The intuitive reasons behind DyG-Hyena's advantages are as follows: (i) We adopt the Hyena layer for temporal context modeling, which significantly reduces the model's computational overhead compared to Transformer-based layers (see Section 5.4). (ii) For dynamic link prediction, CVAE-based conditional modeling of relative temporal information and graph structure-aware CNI encoding enhance the input information fed into the Hyena layer from the temporal and graph structural dimensions, respectively. For dynamic node classification, while we reduce the input of inter-node interaction information, we incorporate node feature statistical encoding into the input, which strengthens the model's ability to learn node representations.

Appendix B Theoretical Derivation

In this section, we derive key formulas for the CVAE and Hyena layers, and analyze their theoretical properties.

Appendix B.1 Enhancing Effect of CVAE

Let $\mathbf{X}_u^t \in \mathbb{R}^{L\times d_{\text{in}}}$ denote the fused feature sequence of one-hop time-sorted neighbors of node u at timestamp t, $\mathbf{C}_u^t = \mathbf{X}_{u,T}^t$ denote the relative time encoding unique to CTDG, $\mathbf{y}$ denote the label of dynamic edge (u, v, t), and $\mathbf{z} \in \mathbb{R}^{d_z}$ denote the latent variable characterizing temporal-structural dependencies. The evidence lower bound (ELBO) for dynamic link prediction is formulated as:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{\mathbf{z}\sim q_\varphi(\mathbf{z}|\mathbf{X}_u^t,\mathbf{C}_u^t)}\left[\log p_\theta(\mathbf{y}|\mathbf{z},\mathbf{X}_u^t,\mathbf{C}_u^t)\right] - D_{\text{KL}}\left(q_\varphi(\mathbf{z}|\mathbf{X}_u^t,\mathbf{C}_u^t)\parallel p(\mathbf{z})\right). \tag{A1}$$

The reparameterization trick conditioned on CTDG temporal constraints is written as:

$$\mathbf{z} = \mu_\varphi(\mathbf{X}_u^t,\mathbf{C}_u^t) + \sigma_\varphi(\mathbf{X}_u^t,\mathbf{C}_u^t)\odot\varepsilon, \quad \varepsilon\sim\mathcal{N}(0,I). \tag{A2}$$

Let $\nabla_\varphi\mu_\varphi$ and $\nabla_\varphi\sigma_\varphi$ denote the gradients of encoder parameters. The gradient of the expected log-likelihood satisfies:

$$\begin{aligned}&\nabla_\varphi\mathbb{E}_\varepsilon\left[\log p_\theta\left(\mathbf{y}\mid\mu_\varphi+\sigma_\varphi\odot\varepsilon,\mathbf{X}_u^t,\mathbf{C}_u^t\right)\right]\\ &=\mathbb{E}_\varepsilon\left[\nabla_\mathbf{z}\log p_\theta(\mathbf{y}|\mathbf{z},\mathbf{X}_u^t,\mathbf{C}_u^t)\cdot\left(\nabla_\varphi\mu_\varphi+\varepsilon\odot\nabla_\varphi\sigma_\varphi\right)\right].\end{aligned} \tag{A3}$$

The expectation of $\left(\nabla_\varphi\mu_\varphi+\varepsilon\odot\nabla_\varphi\sigma_\varphi\right)$ is uniformly bounded, yielding a Monte Carlo gradient variance of $\mathcal{O}(1/K)$, which reduces the iteration complexity to $\mathcal{O}(\log(1/\varepsilon))$ and accelerates convergence on irregular CTDG interaction sequences.

For generalization, let m be the number of synthetic latent samples:

$$\mathbf{z}^{(j)} = \mu + \alpha\sigma\odot\varepsilon^{(j)}, \quad \varepsilon^{(j)}\sim\mathcal{N}(0,I),\ j=1,\ldots,m. \tag{A4}$$

Decoding $\mathbf{z}^{(j)}$ generates m temporally consistent synthetic edges that expand the support of the CTDG training distribution. The generalization gap satisfies:

$$\text{GenGap} \propto \sqrt{\frac{\text{model complexity}}{n(1+m)}}, \tag{A5}$$

where n denotes the number of original observed edges. Increasing m tightens the gap and improves generalization to unseen nodes and timestamps.

Appendix B.2 Implementation and Formulation of Hyena

We detail Hyena's implementation for dynamic graph learning and its differences from the standard version. Building on the CVAE layer (modeling fused features with enhanced temporal information), the Hyena layer acts as a fused feature extractor to capture intra- and inter-sequence temporal dependencies. The standard Hyena [31] matches or exceeds Transformers via alternating implicit long convolutions and gating, formulated as:

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{v}, \tag{A6}$$

where $\mathbf{v}$ is the input sequence, $\mathbf{y}$ is the output, and $\mathbf{H}$ is a data- and kernel-parameterized matrix. It projects $\mathbf{X} \in \mathbb{R}^{L \times d_{\text{in}}}$ (L: unified sequence length; d_{in}: input dimension) to $(U+1) \cdot d_{\text{in}}$, splitting into U gating sequences for implicit convolution/gating. Thus:

$$\mathbf{H} = D_N S_N * \cdots * D_2 S_2 * D_1 S_1, \tag{A7}$$

where D_N is the gating matrix and S_N is a Toeplitz matrix from an implicit parameterized kernel.

To reduce complexity and overhead for dynamic link prediction, we retain Hyena's convolutional gating while simplifying with Conv1d. The Hyena layer computation is:

$$\mathbf{Y}_u^t = H(\mathbf{X}_u^t), \tag{A8}$$

where $\mathbf{Y}_u^t$ is node u's final embedding at t, and $\mathbf{X}_u^t$ is the fused input. During batch training ($\hat{\mathbf{X}} \in \mathbb{R}^{B \times L \times d_{\text{in}}}$, B: batch size), we transpose $\hat{\mathbf{X}}$ to $\hat{\mathbf{X}}' \in \mathbb{R}^{B \times d_{\text{in}} \times L}$ (Conv1d's batch × channel × sequence format) and apply Conv1d:

$$\mathbf{X} = \text{Conv1d}(\hat{\mathbf{X}}'). \tag{A9}$$

Here, $\hat{\mathbf{X}}'_{b,c,s}$ denotes the b-th sample's c-th channel (embedding dimension) at s-th time step; kernel $\mathbf{K} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}} \times k_{\text{size}}}$ (d_{out}: output dimension; k_{size}: kernel size) maps input channel i to output o via $\mathbf{K}_{o,i,k}$. The Conv1d output $\mathbf{Y} \in \mathbb{R}^{B \times d_{\text{out}} \times L}$ is:

$$Y_{b,o,s} = \sum_{i=1}^{d_{\text{in}}} \sum_{k=1}^{k_{\text{size}}} \mathbf{K}_{o,i,k} \cdot \hat{\mathbf{X}}'_{b,i,s+k-\lfloor k_{\text{size}}/2 \rfloor} + b_o, \tag{A10}$$

where b_o is the output channel o's bias. Hyena's implementation references open-source repositories[1].

Appendix B.3 Derivation of Convergence Rate

Appendix B.3.1 Convergence Rate of DyG-Hyena

The total loss of DyG-Hyena for CTDG learning is:

$$\begin{aligned}\mathcal{L}(\theta) = {} & \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{X}_u^t, \mathbf{C}_u^t)} \left\| \mathbf{X}_u^t - f_\theta(\mathbf{z}, \mathbf{C}_u^t) \right\|_2^2 \\ & + \beta D_{\text{KL}}\left(q_\phi(\mathbf{z}|\mathbf{X}_u^t, \mathbf{C}_u^t) \parallel p(\mathbf{z}) \right) + \lambda \left\| \mathbf{h} \right\|_2^2,\end{aligned} \tag{A11}$$

where f_θ denotes the Hyena-based decoder.

Under the Polyak-Łojasiewicz (PL) condition $\|\nabla \mathcal{L}(\theta)\|_2^2 \geq 2\mu(\mathcal{L}(\theta) - \mathcal{L}^*)$ and Lipschitz smoothness $\|\nabla \mathcal{L}(\theta_1) - \nabla \mathcal{L}(\theta_2)\|_2 \leq L\|\theta_1 - \theta_2\|_2$, the convergence rate with step size $\eta = 1/L$ is bounded by:

$$\mathcal{L}(\theta_t) - \mathcal{L}^* \leq \left(1 - \frac{\mu}{L}\right)^t \left(\mathcal{L}(\theta_0) - \mathcal{L}^*\right), \tag{A12}$$

which indicates exponential convergence on CTDG data.

[1] https://github.com/Suro-One/Hyena-Hierarchy, https://github.com/HazyResearch/safari

Appendix B.3.2 Acceleration Effect of CVAE on Convergence

With annealed KL divergence, the effective loss is:

$$\tilde{\mathcal{L}}(\theta) = \mathbb{E}_{q_\phi} \left\| \mathbf{X}_u^t - f_\theta(\mathbf{z}, \mathbf{C}_u^t) \right\|_2^2 + \lambda \left\| \mathbf{h} \right\|_2^2 . \tag{A13}$$

The maximum eigenvalue of the Hessian matrix satisfies:

$$\lambda_{\max}\left(\nabla^2 \tilde{\mathcal{L}}(\theta)\right) \le L_{\text{Hyena}} + \frac{\sigma_z^2}{B}, \tag{A14}$$

where σ_z^2 denotes the variance of latent variable $\mathbf{z}$ and B denotes batch size.

The expected gradient norm decays as:

$$\mathbb{E}\left[\left\| \nabla \tilde{\mathcal{L}}(\theta_t) \right\|_2\right] \le \rho^t \left\| \nabla \tilde{\mathcal{L}}(\theta_0) \right\|_2 , \tag{A15}$$

where $\rho = 1 - \frac{\mu}{L_{\text{Hyena}} + \sigma_z^2/B} < 1 - \frac{\mu}{L}$. Thus, the CVAE module accelerates convergence for irregular CTDG sequences.

Appendix C Detailed Experimental Settings

Appendix C.1 Datasets

We used 12 widely adopted continuous-time dynamic graph datasets [6,7,10,12],which were first curated and released by EdgeBank [7] and are publicly available[2]:

- **Wikipedia** [4] is a bipartite interaction graph containing Wikipedia page edits over one month. Nodes represent encyclopedia users and pages, links denote users' edits to pages at specific timestamps, and the features of each link correspond to the 172-dimensional Linguistic Inquiry and Word Count (LIWC) framework [47]. This dataset additionally includes node dynamic labels indicating whether a user is prohibited from editing.
- **Reddit** [4] is a bipartite graph recording a month's worth of user posts on subreddits. Nodes are divided into users and subreddits, and links are time-stamped post submissions. Like Wikipedia, each link in Reddit contains a 172-dimensional LIWC feature, and the dataset also includes node dynamic labels.
- **MOOC** [4] is a bipartite interaction network comprising online educational interactions. Nodes are students and course resources, links are records of students' access to course resources, and each link contains a four-dimensional feature.
- **LastFM** [4] is a bipartite interaction network containing records of users' music listening behavior over one month. Nodes are users and songs, and links represent users' listening activities.
- **Enron** [48] is a 3-year dataset of ENRON employee communications. Nodes represent employees, and links denote communication interactions.
- **Social Evo.** [49] is a mobile phone proximity network consisting of eight months of daily communication activities among members of an undergraduate dormitory group. Nodes represent undergraduate users, links represent communication processes, and each link has a two-dimensional feature.
- **UCI** [50] is an online communication network for college students. Nodes are college students, and links represent communication processes.
- **Flights** [51] is a dynamic flight network recording the evolution of air traffic over a period of time. Nodes are airports, and links represent time-stamped flights. Each link has a feature representing the cumulative number of flights between the two airports by the current timestamp that day.

[2] https://zenodo.org/records/dynamic-graphs

- **Can. Parl.** [7] is a Canadian Parliamentarian Interactive network documenting the interactions of Canadian Parliamentarians from 2006 to 2019. Each node represents a member of Parliament from a constituency, links represent two members voting together on a motion, and the weight of each link denotes the cumulative number of times one member has voted in alignment with the other to date.
- **US Legis.** [7] is a social network for members of the United States Senate. Nodes represent senators, links denote their social interaction trajectories, and link weights represent the number of bills co-sponsored between members.
- **UN Trade** [7] is a record of food and agricultural trade between 181 countries over 30 years. Nodes represent countries, links denote trade between countries, and their weights represent the sum of standardized agricultural import and export values between two specific countries.
- **Contact** [7] documents the evolution of physical proximity among approximately 700 college students. Each node represents a student, links indicate that they are in close physical proximity to each other, and link weights quantify the degree of closeness.

We present the statistics of the above datasets in Table A2, where #N&L Feat denotes the feature dimensions of nodes or links. Consistent with our experimental setup, we adopt the statistical data reported in [6].

Table A2: Statistics of the datasets.

Datasets	Domains	#Nodes	#Links	#N&L Feat	Bipartite	Duration	Unique Steps	Time Granularity
Wikipedia	Social	9227	157474	-&172	Ture	1 mounth	152757	Unix timestamps
Reddit	Social	10984	672447	-&172	Ture	1 mounth	669065	Unix timestamps
MOOC	Interaction	7144	411749	-&4	Ture	14 mounths	345600	Unix timestamps
LastFM	Interaction	1980	1293103	-&-	Ture	1 mounth	1283614	Unix timestamps
Enron	Social	184	125235	-&-	False	3 years	22632	Unix timestamps
Social Evo.	Proximity	74	2099519	-&2	False	8 mounths	565932	Unix timestamps
UCI	Social	1899	59835	-&-	False	196 days	58911	Unix timestamps
Flights	Transport	13169	1927145	-&1	False	4 mounths	122	Days
Can. Parl.	Politics	734	74478	-&1	False	14 years	14	Years
US Legis.	Politics	225	60396	-&1	False	12 congresses	12	Congresses
UN Trade	Economics	255	507497	-&1	False	32 years	32	Years
UN Vote	Politics	201	1035742	-&1	False	72 years	72	Years
Contact	Proximity	692	2426479	-&1	False	1 mounth	8064	5 min

Appendix C.2 Baselines

The twelve baselines we selected are described as follows:

- **JODIE** [4] is an RNN-based model designed for user-item bipartite interaction graphs. It updates node representations via two coupled RNNs and predicts future node representations through a projection layer.
- **DyRep** [41] is built on an RNN architecture, updating node states sequentially. It also introduces a time-parameterized attention module to aggregate temporal information.
- **TGAT** [17] was the first to introduce the self-attention mechanism into dynamic graph learning. It adopts a one-hop neighbor sampling and aggregation framework, taking node, edge, and time-encoded features of sampled one-hop neighbors as inputs to the self-attention layer, and iteratively computes attention-based node embeddings.
- **TGN** [20, 22] is a memory-based model that maintains a memory buffer for each node to represent its current state. During message passing, each node's memory is updated interactively via message functions, message aggregators, and a memory updater.

- **CAWN** [21] is also based on RNN. First, it extracts multiple causal anonymous walks (CAWs) for each node and generates relative node identifiers. It encodes these identifiers using an RNN and aggregates the encodings to obtain the final node representations.
- **EdgeBank** [7] is a memory-based heuristic method that stores observed edges in memory. If the queried node pair exists in the memory, it predicts a positive link; otherwise, it predicts a negative link. Thus, it is only applicable to transductive dynamic link prediction.
- **TCL** [23] obtains sequences of one-hop neighbor interactions via a short-term sampling framework, following a breadth-first extraction logic. It then uses a Transformer to aggregate graph topological and temporal information for node representation learning, and introduces a cross-attention mechanism to model interaction relationships.
- **GraphMixer** [19] adopts a short-term sampling framework to obtain one-hop neighbor information. It constructs node encoders, link encoders, and link predictors, all based on a simple MLP-Mixer. It has also demonstrated the effectiveness of parameter-fixed temporal encodings.
- **DyGFormer** [6] is a Transformer-based dynamic graph learning model. It aggregates node, edge, temporal, and neighbor co-occurrence information of one-hop neighbors and concatenates them into a sequence. To capture temporal dependencies of the sequence using a Transformer while mitigating the model's inherent computational overhead, patching technology is adopted to shorten the sequence length.
- **DyG-Mamba** [10,29] is a SOTA solution that balances performance and efficiency. It replaces the sequence's time-dependent learning layer with the SSM-based Mamba model, using time differences as control signals for the SSM. This effectively improves the model's utilization of temporal information and achieves strong performance.
- **TPNet** [26] is a temporal GNN for dynamic link prediction. It unifies relative encoding as temporal walk matrix functions (with time-decayed design to capture structural-temporal dependencies), adopts random feature propagation instead of explicit neighbor sampling, and supports efficient, generalizable transductive/inductive learning.
- **FreeDyG** [12] is a recent state-of-the-art (SOTA) method. It extends neighbor co-occurrence encoding to Node Interaction Frequency (NIF) Encoding and uses a frequency-enhanced MLP-Mixer layer to aggregate temporal embeddings. It adopts a short-term dependency sampling framework to obtain neighbor sequences.
- **RepeatMixer** [24] is a dynamic graph model for link prediction, solving traditional methods' oversight of historical repeat interactions (focusing only on recent neighbors). It adopts repeat-aware neighbor sampling (defining first-order/high-order repeat nodes, selecting neighbors via sliding window), uses an MLP encoder for feature-temporal fusion, and a time-aware aggregation mechanism (Pearson Correlation Coefficient weighting).
- **ScaDyG** [13] is a new paradigm for large-scale dynamic graph learning. It achieves efficient and effective dynamic graph representation learning via three core operations: time-aware topology redefinition, dynamic temporal encoding, and hypernetwork-driven message aggregation. Dataset reconstruction requires specific preprocessing steps.

Appendix C.3 Detailed Experimental Configuration

Appendix C.3.1 Model Configurations

We adopt the baseline model configurations from DyGLib [6], and our reproduced results are consistent with those reported in DyGLib. Thus, we directly use their baseline experimental results for comparison. For DyG-Hyena, we adopt the experimental model configuration as shown in Table A3.

Table A3: Experimental model configuration for DyG-Hyena.

Configuration	Value
Learning rate	0.0001
Optimizer	Adam
Maximum train epochs	100
Random seed	0–4
Batch size	200
Early-stop patience	20
Dropout rate	0.5
Dimension of time encoding	100
Dimension of node/edge encoding	172
Dimension of other encodings	50
FFN hidden dimension of Hyena	2048
Number of Hyena layers	2
CVAE latent dimension	64

Appendix C.3.2 Hyperparameter Settings

In Tables A4 and A5, we report the hyperparameter settings of the Hyena layer across different datasets and tasks. It is worth noting that the unified sequence length L is determined via grid search. We adopt a one-hop full-neighbor aggregation framework, where the aggregation length refers to the number of neighbors selected for aggregation. We perform grid search over the range [32, 2048] and select the optimal value for each dataset. We also report the unified two-hop neighbor sequence length L_2, which is used for constructing cross-stage neighbor intersection (CNI) encoding. Since enumerating all two-hop neighbors incurs substantial computational overhead, we adopt a neighbor sampling strategy.

Table A4: Hyperparameter settings in dynamic link prediction.

Datasets	Unified One-Hop Neighbor Sequence Length L	Hyena Conv1d Kernel Size k	Unified Two-Hop Neighbor Sequence Length L_2
Wikipedia	64	9	100
Reddit	64	9	100
MOOC	256	9	100
LastFM	512	9	256
Enron	512	9	256
Social Evo.	64	9	100
UCI	32	9	100
Flights	256	9	256
Can. Parl.	2048	9	100
US Legis.	274	9	256
UN Trade	256	9	256
Contact	32	9	100

Table A5: Hyperparameter settings in dynamic node classification.

Datasets	Unified One-Hop Neighbor Sequence Length L	Hyena Conv1d Kernel Size k
Wikipedia	64	9
Reddit	128	9

Appendix C.3.3 Negative Sampling Strategies in Link Prediction

We implement three negative sampling strategies for link prediction: random, historical, and inductive negative sampling, following the settings in [6]. The dataset is split into three subsets for training, validation, and testing, with each subset corresponding to a distinct time period. Random negative sampling involves randomly selecting node pairs from all possible non-interacting node pairs to form negative samples. Historical negative sampling constructs negative samples using edges that existed before the current step (training/validation/testing) but not within the current step. Inductive negative sampling selects edges that have never appeared prior to the current step to form negative samples.

Appendix C.3.4 Experimental Environment and Memory Usage

Our experiments are conducted on an Ubuntu server equipped with an Intel(R) Xeon(R) Gold 6330 CPU (2.00 GHz) and 112 physical cores. The GPU device is an NVIDIA A800-SXM4-80GB. Our code is implemented in Python 3.12, with model training conducted using PyTorch 2.6.0 on CUDA 12.4. Given the provided hyperparameters (primarily the sequence length L), the GPU memory usage for all datasets does not exceed 80 GB. We present the maximum memory usage statistics for DyG-Hyena during training in Table A6.

Table A6: Peak memory usage of DyG-Hyena.

Task	Dataset	Memory (MB)
Link prediction	Wikipedia	4218
	Reddit	3382
	MOOC	13,724
	LastFM	19,562
	Enron	29,524
	Social Evo.	4302
	UCI	2074
	Flights	10,830
	Can. Parl.	78,048
	US Legis.	12,152
	UN Trade	10,214
	Contact	3682
Node classification	Wikipedia	462
	Reddit	798

Appendix D Detailed Results of Experiments

Appendix D.1 Additional Results for Transductive Dynamic Link Prediction

We report AP scores for transductive dynamic link prediction under random, historical, and inductive negative sampling strategies in Table A12, and ROC-AUC scores for the same setting in Table A13.

Appendix D.2 Additional Results for Inductive Dynamic Link Prediction

We report AP scores for inductive dynamic link prediction under random, historical, and inductive negative sampling strategies in Table A14, and ROC-AUC scores for the same setting in Table A15.

Appendix D.3 Ablation Study

To evaluate the effectiveness of each component in DyG-Hyena across different tasks, we perform ablation experiments on model variants for link prediction and node classification separately. For dynamic link prediction, we design the following variants: (i) DyGFormer w/o patching: As a baseline for fair comparison, we use DyGFormer without adopting its patching technique. (ii) DyG-Hyena: Our proposed full model. (iii) DyG-Hyena w/o CVAE: In dynamic link prediction, relative time features are directly concatenated with other features, bypassing the CVAE layer. (iv) DyG-Hyena w/o CNI: The cross-stage neighbor intersection (CNI) encoding is removed from the fused input. (v) Only Hyena: Both the CVAE layer and CNI encoding are removed to independently evaluate the performance of the Hyena layer. We present ablation results for as many datasets as possible. In addition to performance metrics, we also report epoch time, convergence iterations (to 99% of the final performance), and GPU memory usage. The results are presented in Table A9.

For dynamic node classification, we design the following variants: (i) DyG-Hyena w/o SCN: Without node feature statistical encoding; (ii) DyG-Hyena w/AIF: Following a similar logic to link prediction, we add neighbor co-occurrence encoding and CNI encoding to the fused features. We conduct experiments on the Wikipedia and Reddit datasets, reporting ROC-AUC scores. These results are presented in the main text.

The ablation results indicate that the Hyena layer contributes significantly to efficiency improvement. While the introduction of the CNI and CVAE modules increases per-epoch time cost, they reduce overall convergence iterations and enhance model performance.

Appendix D.4 Evaluation of Generalization Ability

We experimentally demonstrate that DyG-Hyena possesses strong generalization ability. Specifically, we quantitatively evaluate its generalization from two complementary dimensions to avoid the limitations of a single metric. (1) We measure the AP difference between transductive and inductive settings, defined as $\Delta AP_{t,i} = |AP_{tran} - AP_{ind}|$, which quantifies the performance degradation on unseen nodes. A smaller value indicates better generalization to out-of-distribution nodes. All experiments adopt a unified random negative sampling strategy, and the results of four representative SOTA models are presented in Table A7. (2) We evaluate the generalization across diverse temporal granularities. Datasets are divided into fine-grained (Unix timestamps: 7 datasets) and coarse-grained (year/month/day: 5 datasets), and the average AP ranking of each model is calculated for both categories. Steady performance across both categories demonstrates robust cross-granularity generalization, with results shown in Table A8.

From the results in Tables A7 and A8, DyG-Hyena achieves competitive overall performance in terms of generalization ability. Specifically: (i) The one-hop neighbor truncation-padded aggregation framework endows the model with the ability to capture rich neighborhood information, which lays the foundation for generalization to unseen nodes and datasets with different temporal granularities; (ii) The CVAE layer integrated in DyG-Hyena generates latent variables via the reparameterization trick and reconstructs feature representations, which promotes the generation of diverse potential link states. This design further enhances the model's generalization capability when processing unseen data.

Table A7: The AP difference under the random negative sampling and the transduction/induction setting. Bold red values denote the best performance, blue underlined values represent the second-best performance.

Dataset	Graph-Mixer	DyG-Former	DyG-Mamba	DyG-Hyena
Wikipedia	0.60	0.44	0.43	**0.33**
Reddit	2.05	0.38	0.39	**0.32**
MOOC	1.37	0.56	**0.05**	0.20
LastFM	6.50	2.13	0.90	**0.83**
Enron	6.37	2.71	**2.00**	3.05
Social Evo.	**1.51**	1.59	1.54	1.53
UCI	2.06	1.25	1.99	**1.11**
Flights	7.96	1.12	N/A	**1.09**
Can. Parl.	21.13	9.62	**8.15**	9.29
US Legis.	20.03	16.83	**14.14**	17.04
UN Trade	**0.44**	1.91	2.64	1.60
Contact	1.33	0.26	0.26	**0.23**
Avg. Rank	3.33	2.16	2.09	**1.58**

Table A8: Avg. Rank of coarse and fine-grained datasets under transduction settings. Bold red values denote the best performance, blue underlined values represent the second-best performance.

Different Time Granularities	Graphmixer	DyGFormer	DyG-Mamba	DyG-Hyena
Fine-grained	9.29	4.29	2	**1.71**
Coarse-grained	7.25	4.25	2.67	**2.42**

Table A9: Comprehensive ablation experiment. The corresponding data for each model are: epoch time, final convergence round (99%), AP for transductive, GPU usage. All experiments in the same dataset used the same historical neighbor configuration. For each dataset, we have also indicated the maximum number of historical neighbors that were retained (as a hyperparameter).

Dataset	Model	Epoch Time	Number of Epoch	AP for Transductive	GPU Usage (MB)
Wikipedia/64	DyGFormer w/o padding	1 min 11 s	8	99.03	3476
	DyG-Hyena	1 min 27 s	2	99.38	4218
	DyG-Hyena w/o CVAE	1 min 25 s	4	99.24	3994
	DyG-Hyena w/o CNI	1 min 15 s	2	99.18	4130
	Only Hyena	1 min 2 s	10	98.92	4012

(Continued)

Table A9 (continued)

Dataset	Model	Epoch Time	Number of Epoch	AP for Transductive	GPU Usage (MB)
Reddit/64	DyGFormer w/o padding	10 min 37 s	90	99.22	4790
	DyG-Hyena	8 min 49 s	90	99.33	3382
	DyG-Hyena w/o CVAE	8 min 15 s	90	99.08	3247
	DyG-Hyena w/o CNI	8 min 30 s	90	98.72	3379
	Only Hyena	7 min 50 s	90	98.44	3245
MOOC/256	DyGFormer w/o padding	8 min 2 s	60	87.52	25,096
	DyG-Hyena	7 min 34 s	40	96.92	13,724
	DyG-Hyena w/o CVAE	7 min 23 s	50	95.78	13,436
	DyG-Hyena w/o CNI	6 min 15 s	40	89.05	13,522
	Only Hyena	5 min 43 s	60	86.34	12,032
UCI/32	DyGFormer w/o padding	27 s	10	95.79	1704
	DyG-Hyena	36 s	10	97.44	1996
	DyG-Hyena w/o CVAE	35 s	20	96.12	1923
	DyG-Hyena w/o CNI	36 s	10	96.60	1901
	Only Hyena	24 s	10	94.97	1644

Table A10: AP of transductive dynamic link prediction under different hardware conditions.

Model	Wikipedia	Reddit	MOOC	UCI
DyG-Hyena (Main Experiment)	99.38	99.33	96.92	97.44
DyG-Hyena (RTX 3090)	99.38	99.30	96.88	97.42

Table A11: MRR of transductive dynamic link prediction on extended datasets.

Model	tgbl-wiki	tgbl-coin
DyGFormer	0.798	0.752
DyG-Hyena	0.802	0.758

Table A12: AP for transductive dynamic link prediction with random, historical, and inductive negative sampling strategies. Bold red values denote the best performance, blue underlined values represent the second-best performance, and N/A means unavailable data or unused method.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
hist	Wikipedia	96.50 ± 0.14	94.86 ± 0.06	96.94 ± 0.06	98.45 ± 0.06	98.76 ± 0.03	90.37 ± 0.00	96.47 ± 0.16	97.25 ± 0.03	99.03 ± 0.02	99.15 ± 0.02	99.32 ± 0.03	99.16 ± 0.02	99.26 ± 0.01	N/A	**99.38 ± 0.01**
	Reddit	98.31 ± 0.14	98.22 ± 0.04	98.52 ± 0.02	98.63 ± 0.06	99.11 ± 0.01	94.86 ± 0.00	97.53 ± 0.02	97.31 ± 0.01	99.22 ± 0.01	99.32 ± 0.01	99.27 ± 0.00	99.22 ± 0.01	**99.48 ± 0.01**	N/A	99.33 ± 0.01
	MOOC	80.23 ± 2.44	81.97 ± 0.49	85.84 ± 0.15	89.15 ± 1.60	80.15 ± 0.25	57.97 ± 0.00	82.38 ± 0.24	82.78 ± 0.15	87.52 ± 0.49	89.21 ± 0.08	96.39 ± 0.09	92.76 ± 0.10	89.61 ± 0.19	90.10 ± 1.20	**96.92 ± 0.33**
	LastFM	70.85 ± 2.13	71.92 ± 2.21	73.42 ± 0.21	77.07 ± 3.97	86.99 ± 0.06	79.29 ± 0.00	67.27 ± 2.16	75.61 ± 0.24	93.00 ± 0.12	93.35 ± 0.20	94.50 ± 0.08	94.14 ± 0.06	92.15 ± 0.16	N/A	**95.58 ± 0.01**
	Enron	84.77 ± 0.30	82.38 ± 3.36	71.12 ± 0.97	86.53 ± 1.11	89.56 ± 0.09	83.53 ± 0.00	79.70 ± 0.71	82.25 ± 0.16	92.47 ± 0.12	92.65 ± 0.12	**92.90 ± 0.17**	92.66 ± 0.07	92.51 ± 0.05	N/A	92.72 ± 0.11
	Social Evo.	89.89 ± 0.55	88.87 ± 0.30	93.16 ± 0.17	93.57 ± 0.17	84.96 ± 0.09	74.95 ± 0.00	93.13 ± 0.16	93.37 ± 0.07	94.73 ± 0.01	94.77 ± 0.01	94.73 ± 0.02	N/A	94.91 ± 0.01	N/A	**94.97 ± 0.01**
	UCI	89.43 ± 1.09	65.14 ± 2.30	79.63 ± 0.70	92.34 ± 1.04	95.18 ± 0.06	76.20 ± 0.00	89.57 ± 1.63	93.25 ± 0.57	95.79 ± 0.17	95.91 ± 0.15	97.35 ± 0.04	96.74 ± 0.08	96.28 ± 0.11	95.60 ± 0.20	**97.44 ± 0.03**
	Flights	95.60 ± 1.73	95.29 ± 0.72	94.03 ± 0.18	97.95 ± 0.14	98.51 ± 0.01	89.35 ± 0.00	91.23 ± 0.02	90.99 ± 0.05	98.92 ± 0.01	98.95 ± 0.05	98.93 ± 0.02	N/A	N/A	N/A	**98.96 ± 0.01**
	Can. Parl.	69.26 ± 0.31	66.54 ± 2.76	70.73 ± 0.72	70.88 ± 2.34	69.82 ± 2.34	64.55 ± 0.00	68.67 ± 2.67	77.04 ± 0.46	97.36 ± 0.45	**99.57 ± 0.08**	90.28 ± 0.37	N/A	N/A	N/A	98.99 ± 0.42
	US Legis.	75.05 ± 1.52	75.34 ± 0.39	68.52 ± 3.16	75.99 ± 0.58	70.58 ± 0.48	58.39 ± 0.00	69.59 ± 0.48	70.74 ± 1.02	71.11 ± 0.59	71.75 ± 0.26	**80.58 ± 0.23**	N/A	N/A	N/A	71.34 ± 0.80
	UN Trade	64.94 ± 0.31	63.21 ± 0.93	61.47 ± 0.18	65.03 ± 1.37	65.39 ± 0.12	60.41 ± 0.00	62.21 ± 0.03	62.61 ± 0.27	66.46 ± 1.29	67.50 ± 0.24	**87.24 ± 0.65**	N/A	N/A	N/A	68.62 ± 0.11
	Contact	95.31 ± 1.33	95.98 ± 0.15	96.28 ± 0.09	96.89 ± 0.56	90.26 ± 0.28	92.58 ± 0.00	92.44 ± 0.12	91.92 ± 0.03	98.29 ± 0.01	98.43 ± 0.12	98.66 ± 0.01	N/A	N/A	N/A	**98.70 ± 0.01**
	Avg. Rank	8.08	8.92	8.58	5.50	7.42	11.50	10.58	8.08	4.17	2.75	2.17	N/A	N/A	N/A	**1.67**
hist	Wikipedia	83.01 ± 0.66	79.93 ± 0.56	87.38 ± 0.22	86.86 ± 0.33	71.21 ± 1.67	73.35 ± 0.00	89.05 ± 0.39	90.90 ± 0.10	82.23 ± 2.54	81.77 ± 1.20	81.55 ± 4.10	90.20 ± 1.04	**91.59 ± 0.57**	N/A	83.93 ± 1.93
	Reddit	80.03 ± 0.36	79.83 ± 0.31	79.55 ± 0.20	81.22 ± 0.61	80.82 ± 0.45	73.59 ± 0.00	77.14 ± 0.16	78.44 ± 0.18	81.57 ± 0.67	81.20 ± 1.52	81.02 ± 1.31	83.02 ± 1.20	**85.67 ± 1.01**	N/A	83.42 ± 0.16
	MOOC	78.94 ± 1.25	75.60 ± 1.12	82.19 ± 0.62	87.06 ± 1.93	74.05 ± 0.95	60.71 ± 0.00	77.06 ± 0.41	77.77 ± 0.92	85.85 ± 0.66	85.89 ± 0.94	92.69 ± 0.95	92.19 ± 0.58	86.71 ± 0.81	N/A	**93.36 ± 0.90**
	LastFM	74.35 ± 3.81	74.92 ± 2.46	71.59 ± 0.24	76.87 ± 4.64	69.86 ± 0.43	73.03 ± 0.00	59.30 ± 2.31	72.47 ± 0.49	81.57 ± 0.48	83.02 ± 0.16	87.74 ± 0.50	86.73 ± 0.34	79.71 ± 0.51	N/A	**87.91 ± 0.36**
	Enron	69.85 ± 2.70	71.19 ± 2.76	64.07 ± 1.05	73.91 ± 1.76	64.73 ± 0.36	76.53 ± 0.00	70.66 ± 0.39	77.98 ± 0.92	75.63 ± 0.73	77.77 ± 1.32	80.79 ± 1.68	**87.38 ± 0.18**	78.87 ± 0.82	N/A	78.99 ± 2.12
	Social Evo.	87.44 ± 6.78	93.29 ± 0.43	95.01 ± 0.44	94.45 ± 0.56	85.53 ± 0.38	80.57 ± 0.00	94.74 ± 0.31	94.93 ± 0.31	97.38 ± 0.14	97.35 ± 0.52	96.80 ± 0.41	N/A	97.79 ± 0.23	N/A	**97.83 ± 0.35**
	UCI	75.24 ± 5.80	55.10 ± 3.14	68.27 ± 1.37	80.43 ± 2.12	65.30 ± 0.43	65.50 ± 0.00	80.25 ± 2.74	84.11 ± 1.35	82.17 ± 0.82	81.03 ± 1.09	86.34 ± 0.80	87.23 ± 0.23	86.10 ± 1.19	N/A	**90.21 ± 0.25**
	Flights	66.48 ± 2.59	67.61 ± 0.99	**72.38 ± 0.18**	66.70 ± 1.64	64.72 ± 0.97	70.53 ± 0.00	70.68 ± 0.24	71.47 ± 0.26	66.59 ± 0.49	67.80 ± 0.21	69.10 ± 1.27	N/A	N/A	N/A	67.33 ± 0.22
	Can. Parl.	51.79 ± 0.63	63.31 ± 1.23	67.13 ± 0.84	68.42 ± 3.07	66.53 ± 2.77	63.84 ± 0.00	65.93 ± 3.00	74.34 ± 0.87	97.00 ± 0.31	**99.77 ± 0.12**	86.61 ± 3.57	N/A	N/A	N/A	99.05 ± 0.98

(Continued)

Table A12 (continued)

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
	US Legis.	51.71 ± 5.76	86.88 ± 2.25	62.14 ± 6.60	74.00 ± 7.57	68.82 ± 8.23	63.22 ± 0.00	80.53 ± 3.95	81.65 ± 1.02	85.30 ± 3.88	82.15 ± 1.02	**94.55 ± 0.62**	N/A	N/A	N/A	<u>86.92 ± 2.92</u>
	UN Trade	61.39 ± 1.83	59.19 ± 1.07	55.74 ± 0.91	58.44 ± 5.51	55.71 ± 0.38	<u>81.32 ± 0.00</u>	55.90 ± 1.17	57.05 ± 1.22	64.41 ± 1.40	65.10 ± 0.02	**85.22 ± 1.22**	N/A	N/A	N/A	64.62 ± 1.44
	Contact	95.31 ± 2.13	96.39 ± 0.20	96.05 ± 0.52	93.05 ± 2.35	84.16 ± 0.49	88.81 ± 0.00	93.86 ± 0.21	93.36 ± 0.41	97.57 ± 0.06	97.61 ± 0.04	**98.02 ± 0.15**	N/A	N/A	N/A	<u>97.80 ± 0.02</u>
	Avg. Rank	8.67	7.92	7.33	6.33	10.42	8.92	7.75	5.75	4.67	4.08	<u>3.17</u>	N/A	N/A	N/A	**2.50**
ind	Wikipedia	75.65 ± 0.79	70.21 ± 1.58	87.00 ± 0.16	85.62 ± 0.44	74.06 ± 2.62	80.63 ± 0.00	86.76 ± 0.72	88.59 ± 0.17	78.29 ± 5.38	79.86 ± 2.18	79.35 ± 5.52	<u>88.86 ± 0.97</u>	**90.05 ± 0.79**	N/A	82.83 ± 3.46
	Reddit	86.98 ± 0.16	86.30 ± 0.26	89.59 ± 0.24	88.10 ± 0.24	**91.67 ± 0.24**	85.48 ± 0.00	87.45 ± 0.29	85.26 ± 0.11	91.11 ± 0.40	91.15 ± 0.54	88.19 ± 0.33	91.11 ± 0.73	90.74 ± 0.17	N/A	<u>91.26 ± 0.58</u>
	MOOC	65.23 ± 2.19	61.66 ± 0.95	75.95 ± 0.64	77.50 ± 2.91	73.51 ± 0.94	49.43 ± 0.00	74.65 ± 0.54	74.27 ± 0.92	81.24 ± 0.69	81.11 ± 0.63	<u>88.18 ± 0.97</u>	81.94 ± 0.95	83.11 ± 1.28	N/A	**88.42 ± 0.12**
	LastFM	62.67 ± 4.49	64.41 ± 2.70	71.13 ± 0.17	65.95 ± 5.98	67.48 ± 0.77	75.49 ± 0.00	58.21 ± 0.89	68.12 ± 0.33	73.97 ± 0.50	73.63 ± 0.54	<u>77.99 ± 1.30</u>	75.46 ± 0.78	72.19 ± 0.24	N/A	**78.21 ± 1.38**
	Enron	68.96 ± 0.98	67.79 ± 1.53	63.94 ± 1.36	70.89 ± 2.72	75.15 ± 0.58	73.89 ± 0.00	71.29 ± 0.32	75.01 ± 0.79	77.41 ± 0.89	80.86 ± 1.24	75.36 ± 1.81	**83.17 ± 0.50**	77.81 ± 0.65	N/A	77.69 ± 1.04
	Social Evo.	89.82 ± 4.11	93.28 ± 0.48	94.84 ± 0.44	95.13 ± 0.56	88.32 ± 0.27	83.69 ± 0.00	94.90 ± 0.36	94.72 ± 0.33	97.68 ± 0.10	<u>97.68 ± 0.42</u>	97.35 ± 0.32	N/A	97.57 ± 0.15	N/A	**97.82 ± 0.45**
	UCI	65.99 ± 1.40	54.79 ± 1.76	68.67 ± 0.84	70.94 ± 0.71	64.61 ± 0.48	57.43 ± 0.00	76.01 ± 1.11	80.10 ± 0.51	72.25 ± 1.71	71.95 ± 2.51	77.26 ± 1.57	**84.20 ± 0.34**	<u>82.35 ± 0.73</u>	N/A	82.06 ± 0.95
	Flights	69.07 ± 4.02	70.57 ± 1.82	<u>75.48 ± 0.26</u>	71.09 ± 2.72	69.18 ± 1.52	**81.08 ± 0.00**	74.62 ± 0.18	74.87 ± 0.21	70.92 ± 1.78	73.79 ± 5.69	64.78 ± 1.50	N/A	N/A	N/A	71.12 ± 2.44
	Can. Parl.	69.26 ± 0.31	66.54 ± 2.76	70.73 ± 0.72	70.88 ± 2.34	69.82 ± 2.34	64.55 ± 0.00	68.67 ± 2.67	77.04 ± 0.46	97.36 ± 0.45	<u>98.32 ± 0.34</u>	85.59 ± 3.08	N/A	N/A	N/A	**98.99 ± 0.42**
	US Legis.	50.27 ± 5.13	83.44 ± 1.16	61.91 ± 5.82	67.57 ± 6.47	65.81 ± 8.52	64.74 ± 0.00	78.15 ± 3.34	79.63 ± 0.84	81.25 ± 3.62	81.67 ± 2.16	**91.05 ± 1.21**	N/A	N/A	N/A	<u>84.41 ± 1.36</u>
	UN Trade	60.42 ± 1.48	60.19 ± 1.24	60.61 ± 1.24	61.04 ± 6.01	62.54 ± 0.67	<u>72.97 ± 0.00</u>	61.06 ± 1.74	60.15 ± 1.29	55.79 ± 1.02	58.89 ± 0.98	**86.61 ± 0.99**	N/A	N/A	N/A	60.22 ± 2.44
	Contact	93.43 ± 1.78	94.18 ± 0.10	94.35 ± 0.48	90.18 ± 3.28	89.31 ± 0.27	85.20 ± 0.00	91.35 ± 0.21	90.87 ± 0.35	95.01 ± 0.15	94.63 ± 0.06	**95.84 ± 0.32**	N/A	N/A	N/A	<u>95.20 ± 0.18</u>
	Avg. Rank	10.75	10.50	7.33	7.58	8.83	8.33	7.92	7.50	5.83	5.25	<u>4.17</u>	N/A	N/A	N/A	**3.25**

Table A13: ROC-AUC for transductive dynamic link prediction with random, historical, and inductive negative sampling strategies. Bold red values denote the best performance, blue underlined values represent the second-best performance, and N/A means unavailable data or unused method.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
rnd	Wikipedia	96.33 ± 0.07	94.37 ± 0.09	96.67 ± 0.07	98.37 ± 0.07	98.54 ± 0.04	90.78 ± 0.00	95.84 ± 0.18	96.92 ± 0.03	98.91 ± 0.02	99.08 ± 0.02	99.30 ± 0.02	99.04 ± 0.01	**99.41 ± 0.01**	N/A	<u>99.39 ± 0.01</u>
	Reddit	98.31 ± 0.05	98.17 ± 0.05	98.47 ± 0.02	98.60 ± 0.06	99.01 ± 0.01	95.37 ± 0.00	97.42 ± 0.02	97.17 ± 0.02	99.15 ± 0.01	99.27 ± 0.01	99.22 ± 0.00	99.15 ± 0.01	**99.50 ± 0.01**	N/A	<u>99.29 ± 0.01</u>
	MOOC	83.81 ± 2.09	85.03 ± 0.58	87.11 ± 0.19	91.21 ± 1.15	80.38 ± 0.26	60.86 ± 0.00	83.12 ± 0.18	84.01 ± 0.17	87.91 ± 0.58	89.58 ± 0.12	<u>97.17 ± 0.08</u>	93.60 ± 0.39	89.93 ± 0.35	N/A	**97.34 ± 1.17**
	LastFM	70.49 ± 1.66	71.16 ± 1.89	71.59 ± 0.18	78.47 ± 2.94	85.92 ± 0.10	83.77 ± 0.00	64.06 ± 1.16	73.53 ± 0.12	93.05 ± 0.10	93.31 ± 0.18	<u>94.39 ± 0.04</u>	94.27 ± 0.02	93.42 ± 0.15	N/A	**95.69 ± 0.01**
	Enron	87.96 ± 0.52	84.89 ± 3.00	68.89 ± 1.10	88.32 ± 0.99	90.45 ± 0.14	87.05 ± 0.00	75.74 ± 0.72	84.38 ± 0.21	93.33 ± 0.13	93.34 ± 0.23	<u>93.98 ± 0.26</u>	93.47 ± 0.17	94.01 ± 0.11	N/A	**94.23 ± 0.10**
	Social Evo.	92.05 ± 0.46	90.76 ± 0.21	94.76 ± 0.16	95.39 ± 0.17	87.34 ± 0.08	81.60 ± 0.00	94.84 ± 0.17	95.23 ± 0.07	96.30 ± 0.01	96.38 ± 0.02	96.43 ± 0.02	N/A	**96.59 ± 0.04**	N/A	<u>96.46 ± 0.08</u>
	UCI	90.44 ± 0.49	68.77 ± 2.34	78.53 ± 0.74	92.03 ± 1.13	93.87 ± 0.08	77.30 ± 0.00	87.82 ± 1.36	91.81 ± 0.67	94.77 ± 0.18	95.32 ± 0.18	<u>96.79 ± 0.05</u>	93.54 ± 0.14	95.00 ± 0.21	N/A	**96.87 ± 0.06**
	Flights	96.21 ± 1.42	95.95 ± 0.62	94.13 ± 0.17	98.22 ± 0.13	98.45 ± 0.01	90.23 ± 0.00	91.21 ± 0.02	91.13 ± 0.01	98.93 ± 0.01	98.98 ± 0.05	<u>99.00 ± 0.02</u>	N/A	N/A	N/A	**99.01 ± 0.01**
	Can. Parl.	78.21 ± 0.23	73.35 ± 3.67	75.69 ± 0.78	76.99 ± 1.80	75.70 ± 3.27	64.14 ± 0.00	72.46 ± 3.23	83.17 ± 0.53	97.76 ± 0.41	**99.69 ± 0.06**	92.05 ± 0.34	N/A	N/A	N/A	<u>99.45 ± 0.12</u>
	US Legis.	82.85 ± 1.07	82.28 ± 0.32	75.84 ± 1.99	<u>83.34 ± 0.43</u>	77.16 ± 0.39	62.57 ± 0.00	76.27 ± 0.63	76.96 ± 0.79	77.90 ± 0.58	79.03 ± 0.26	**86.49 ± 0.18**	N/A	N/A	N/A	77.48 ± 0.41
	UN Trade	69.62 ± 0.44	67.44 ± 0.83	64.01 ± 0.12	69.10 ± 1.67	68.54 ± 0.18	66.75 ± 0.00	64.72 ± 0.05	65.52 ± 0.51	70.20 ± 1.44	71.41 ± 0.21	**89.17 ± 0.46**	N/A	N/A	N/A	<u>72.41 ± 0.18</u>
	Contact	96.66 ± 0.89	96.48 ± 0.14	96.95 ± 0.08	97.54 ± 0.35	89.99 ± 0.34	94.34 ± 0.00	94.15 ± 0.09	93.94 ± 0.02	98.53 ± 0.01	98.68 ± 0.02	**98.91 ± 0.01**	N/A	N/A	N/A	<u>98.74 ± 0.01</u>
	Avg. Rank	8.00	9.42	9.50	5.92	7.83	11.16	10.67	9.16	4.75	3.50	<u>2.33</u>	N/A	N/A	N/A	**2.00**
hist	Wikipedia	80.77 ± 0.73	77.74 ± 0.33	82.87 ± 0.22	82.74 ± 0.32	67.84 ± 0.64	77.27 ± 0.00	85.76 ± 0.46	**87.68 ± 0.17**	78.80 ± 1.95	78.99 ± 1.24	79.89 ± 2.47	<u>85.32 ± 0.70</u>	82.78 ± 0.30	N/A	82.18 ± 1.02
	Reddit	80.52 ± 0.32	80.15 ± 0.18	79.33 ± 0.16	81.11 ± 0.19	80.27 ± 0.30	78.58 ± 0.00	76.49 ± 0.16	77.80 ± 0.12	80.54 ± 0.29	81.71 ± 0.49	81.87 ± 0.49	<u>81.95 ± 0.56</u>	**85.92 ± 0.10**	N/A	81.43 ± 0.34
	MOOC	82.75 ± 0.83	81.06 ± 0.94	80.81 ± 0.67	88.00 ± 1.80	71.57 ± 1.07	61.90 ± 0.00	72.09 ± 0.56	76.68 ± 1.40	87.04 ± 0.35	87.91 ± 0.93	<u>93.45 ± 0.67</u>	92.09 ± 0.32	88.32 ± 0.99	N/A	**94.01 ± 0.74**
	LastFM	75.22 ± 2.36	74.65 ± 1.98	64.27 ± 0.26	77.97 ± 3.04	67.88 ± 0.24	78.09 ± 0.00	47.24 ± 3.13	64.21 ± 0.73	78.78 ± 0.35	79.82 ± 0.27	**84.64 ± 0.45**	<u>82.35 ± 0.39</u>	73.53 ± 0.12	N/A	81.83 ± 0.44
	Enron	75.39 ± 2.37	74.69 ± 3.55	61.85 ± 1.43	77.09 ± 2.22	65.10 ± 0.34	79.59 ± 0.00	67.95 ± 0.88	75.27 ± 1.14	76.55 ± 0.52	77.73 ± 0.61	<u>81.16 ± 1.28</u>	**84.33 ± 0.13**	75.74 ± 0.72	N/A	79.60 ± 2.07
	Social Evo.	90.06 ± 3.15	93.12 ± 0.34	93.08 ± 0.59	94.71 ± 0.53	87.43 ± 0.15	85.81 ± 0.00	93.44 ± 0.68	94.39 ± 0.31	97.28 ± 0.07	97.27 ± 0.30	97.22 ± 0.30	N/A	**97.42 ± 0.02**	N/A	<u>97.38 ± 0.04</u>
	UCI	78.64 ± 3.50	57.91 ± 3.12	58.89 ± 1.57	77.25 ± 2.68	57.86 ± 0.15	69.56 ± 0.00	72.25 ± 3.46	77.54 ± 2.02	76.97 ± 0.24	75.43 ± 1.99	<u>80.42 ± 0.64</u>	78.85 ± 0.43	80.38 ± 0.26	N/A	**85.99 ± 0.42**
	Flights	68.97 ± 1.87	69.43 ± 0.90	<u>72.20 ± 0.16</u>	68.39 ± 0.95	66.11 ± 0.71	**74.64 ± 0.00**	70.57 ± 0.18	70.37 ± 0.23	68.09 ± 0.43	68.98 ± 0.81	71.82 ± 0.82	N/A	N/A	N/A	68.11 ± 0.22
	Can. Parl.	62.44 ± 1.11	70.16 ± 1.70	70.86 ± 0.94	73.23 ± 3.08	72.06 ± 3.94	63.04 ± 0.00	69.95 ± 3.70	79.03 ± 1.01	97.61 ± 0.40	**99.82 ± 0.10**	86.39 ± 3.73	N/A	N/A	N/A	<u>99.60 ± 0.34</u>

(Continued)

Table A13 (continued)

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
	US Legis.	67.47 ± 6.40	91.44 ± 1.18	73.47 ± 5.25	83.53 ± 4.53	78.62 ± 7.46	67.41 ± 0.00	83.97 ± 3.71	85.17 ± 0.70	90.77 ± 1.96	88.36 ± 1.78	**96.28 ± 0.44**	N/A	N/A	N/A	89.84 ± 2.12
	UN Trade	68.92 ± 1.40	64.36 ± 1.40	60.37 ± 0.68	63.93 ± 5.41	63.09 ± 0.74	86.61 ± 0.00	61.43 ± 1.04	63.20 ± 1.54	73.86 ± 1.13	71.41 ± 0.21	**88.90 ± 1.00**	N/A	N/A	N/A	64.99 ± 1.33
	Contact	96.35 ± 0.92	96.00 ± 0.23	95.39 ± 0.43	93.76 ± 1.29	83.06 ± 0.32	92.17 ± 0.00	93.34 ± 0.19	93.14 ± 0.34	97.17 ± 0.05	97.27 ± 0.16	97.73 ± 0.11	N/A	N/A	N/A	**97.85 ± 0.08**
	Avg. Rank	7.16	7.58	8.33	5.91	10.25	8.25	8.50	7.08	5.08	4.25	**2.42**	N/A	N/A	N/A	3.16
ind	Wikipedia	70.96 ± 0.78	67.36 ± 0.96	81.93 ± 0.22	80.97 ± 0.31	70.95 ± 0.95	81.73 ± 0.00	82.19 ± 0.48	**84.28 ± 0.30**	75.09 ± 3.70	75.64 ± 2.42	75.36 ± 3.41	83.67 ± 0.89	82.74 ± 0.32	N/A	76.54 ± 1.91
	Reddit	83.51 ± 0.15	82.90 ± 0.31	87.13 ± 0.20	84.56 ± 0.24	**88.04 ± 0.29**	85.93 ± 0.00	84.67 ± 0.29	82.21 ± 0.13	86.23 ± 0.51	86.35 ± 0.52	81.64 ± 0.42	86.45 ± 1.03	84.38 ± 0.21	N/A	85.94 ± 0.26
	MOOC	66.63 ± 2.30	63.26 ± 1.01	73.18 ± 0.33	77.44 ± 2.86	70.32 ± 1.43	48.18 ± 0.00	70.36 ± 0.37	72.45 ± 0.72	80.76 ± 0.76	81.08 ± 0.82	89.07 ± 0.63	81.94 ± 0.95	78.47 ± 0.94	N/A	**89.28 ± 1.47**
	LastFM	61.32 ± 3.49	62.15 ± 2.12	63.99 ± 0.21	65.46 ± 4.27	67.92 ± 0.44	**77.37 ± 0.00**	46.93 ± 2.59	60.22 ± 0.32	69.25 ± 0.36	68.74 ± 0.55	72.48 ± 0.90	67.73 ± 0.96	72.30 ± 0.59	N/A	72.51 ± 3.47
	Enron	70.92 ± 1.05	68.73 ± 1.34	60.45 ± 2.12	71.34 ± 2.46	75.17 ± 0.50	75.00 ± 0.00	67.64 ± 0.86	71.53 ± 0.85	74.07 ± 0.64	75.47 ± 1.41	75.44 ± 1.38	**80.05 ± 0.53**	77.27 ± 0.61	N/A	77.38 ± 1.37
	Social Evo.	90.01 ± 3.19	93.07 ± 0.38	92.94 ± 0.61	95.24 ± 0.56	89.93 ± 0.15	87.88 ± 0.00	93.44 ± 0.72	94.22 ± 0.32	97.51 ± 0.06	97.37 ± 0.64	97.59 ± 0.25	N/A	**98.47 ± 0.02**	N/A	97.84 ± 0.53
	UCI	64.14 ± 1.26	54.25 ± 2.01	60.80 ± 1.01	64.11 ± 1.04	58.06 ± 0.26	58.03 ± 0.00	70.05 ± 1.86	74.59 ± 0.74	65.96 ± 1.18	66.83 ± 2.83	70.85 ± 0.96	**76.56 ± 0.30**	75.39 ± 0.57	N/A	76.01 ± 0.15
	Flights	69.99 ± 3.10	71.13 ± 1.55	73.47 ± 0.18	71.63 ± 1.72	69.70 ± 0.75	**81.10 ± 0.00**	72.54 ± 0.19	72.21 ± 0.21	69.53 ± 1.17	71.16 ± 3.24	64.21 ± 1.30	N/A	N/A	N/A	70.21 ± 2.67
	Can. Parl.	52.88 ± 0.80	63.53 ± 0.65	72.47 ± 1.18	69.57 ± 2.81	72.93 ± 1.78	61.41 ± 0.00	69.47 ± 2.12	70.52 ± 0.94	96.70 ± 0.59	**99.56 ± 0.21**	85.05 ± 2.71	N/A	N/A	N/A	96.87 ± 1.33
	US Legis.	59.05 ± 5.52	89.44 ± 0.71	71.62 ± 5.42	78.12 ± 4.46	76.45 ± 7.02	68.66 ± 0.00	82.54 ± 3.91	84.22 ± 0.91	87.96 ± 1.80	86.06 ± 2.27	**94.48 ± 0.50**	N/A	N/A	N/A	89.55 ± 1.12
	UN Trade	66.82 ± 1.27	65.60 ± 1.28	66.13 ± 0.78	66.37 ± 5.39	71.73 ± 0.74	74.20 ± 0.00	67.80 ± 1.21	66.53 ± 1.22	62.56 ± 1.51	67.60 ± 0.64	**89.56 ± 0.87**	N/A	N/A	N/A	70.20 ± 0.30
	Contact	94.47 ± 1.08	94.23 ± 0.18	94.10 ± 0.41	91.64 ± 1.72	87.68 ± 0.24	85.87 ± 0.00	91.23 ± 0.19	90.96 ± 0.27	95.01 ± 0.15	95.68 ± 0.20	**95.93 ± 0.23**	N/A	N/A	N/A	95.09 ± 0.08
	Avg. Rank	9.33	8.41	7.25	7.25	7.66	7.58	7.33	6.83	6.08	4.33	4.50	N/A	N/A	N/A	**2.58**

Table A14: AP for inductive dynamic link prediction with random, historical, and inductive negative sampling strategies. Bold red values denote the best performance, blue underlined values represent the second-best performance, and N/A means unavailable data or unused method.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
rnd	Wikipedia	94.82 ± 0.20	92.43 ± 0.37	96.22 ± 0.07	97.83 ± 0.04	98.24 ± 0.03	N/A	96.22 ± 0.17	96.65 ± 0.02	98.59 ± 0.03	98.77 ± 0.03	98.91 ± 0.01	98.70 ± 0.05	98.97 ± 0.01	N/A	**99.05 ± 0.01**
	Reddit	96.50 ± 0.13	96.09 ± 0.11	97.09 ± 0.04	97.50 ± 0.07	98.62 ± 0.01	N/A	94.09 ± 0.07	95.26 ± 0.02	98.84 ± 0.02	98.97 ± 0.01	98.86 ± 0.01	98.85 ± 0.01	98.91 ± 0.01	N/A	**99.01 ± 0.01**
	MOOC	79.63 ± 1.92	81.07 ± 0.44	85.50 ± 0.19	89.04 ± 1.17	81.42 ± 0.24	N/A	80.60 ± 0.22	81.41 ± 0.21	86.96 ± 0.43	88.64 ± 0.08	95.07 ± 0.26	93.05 ± 0.12	87.75 ± 0.62	N/A	**95.96 ± 0.46**
	LastFM	81.61 ± 3.82	83.02 ± 1.48	78.63 ± 0.31	81.45 ± 4.29	89.42 ± 0.07	N/A	73.53 ± 1.66	82.11 ± 0.42	94.23 ± 0.09	94.42 ± 0.21	95.36 ± 0.11	94.98 ± 0.12	94.89 ± 0.01	N/A	**96.41 ± 0.01**
	Enron	80.72 ± 1.39	74.55 ± 3.95	67.05 ± 1.51	77.94 ± 1.02	86.35 ± 0.51	N/A	76.14 ± 0.79	75.88 ± 0.48	89.76 ± 0.34	89.67 ± 0.27	90.34 ± 0.28	87.97 ± 0.29	89.69 ± 0.17	N/A	**90.56 ± 0.27**
	Social Evo.	91.96 ± 0.48	90.04 ± 0.47	91.41 ± 0.16	90.77 ± 0.86	79.94 ± 0.18	N/A	91.55 ± 0.09	91.86 ± 0.06	93.14 ± 0.04	93.13 ± 0.05	93.24 ± 0.07	N/A	**94.76 ± 0.05**	N/A	93.44 ± 0.01
	UCI	79.86 ± 1.48	57.48 ± 1.87	79.54 ± 0.48	88.12 ± 2.05	92.73 ± 0.06	N/A	87.36 ± 2.03	91.19 ± 0.42	94.54 ± 0.12	94.76 ± 0.19	95.74 ± 0.05	95.04 ± 0.12	94.85 ± 0.10	N/A	**95.85 ± 0.23**
	Flights	94.74 ± 0.37	92.88 ± 0.73	88.73 ± 0.33	95.03 ± 0.60	97.06 ± 0.02	N/A	83.41 ± 0.07	83.03 ± 0.05	97.79 ± 0.02	97.85 ± 0.22	97.97 ± 0.04	N/A	N/A	N/A	**98.01 ± 0.01**
	Can. Parl.	53.92 ± 0.94	54.02 ± 0.76	55.18 ± 0.79	54.10 ± 0.93	55.80 ± 0.69	N/A	54.30 ± 0.66	55.91 ± 0.82	87.74 ± 0.71	**93.46 ± 2.62**	68.09 ± 1.55	N/A	N/A	N/A	89.70 ± 0.54
	US Legis.	54.93 ± 2.29	57.28 ± 0.71	51.00 ± 3.11	58.63 ± 0.37	53.17 ± 1.20	N/A	52.59 ± 0.97	50.71 ± 0.76	54.28 ± 2.87	55.95 ± 1.16	**61.71 ± 0.84**	N/A	N/A	N/A	54.30 ± 1.10
	UN Trade	59.65 ± 0.77	57.02 ± 0.69	61.03 ± 0.18	58.31 ± 3.15	65.24 ± 0.21	N/A	62.21 ± 0.12	62.17 ± 0.31	64.55 ± 0.62	70.55 ± 0.04	**86.53 ± 0.29**	N/A	N/A	N/A	67.02 ± 0.21
	Contact	94.34 ± 1.45	92.18 ± 0.41	95.87 ± 0.11	93.82 ± 0.99	89.55 ± 0.30	N/A	91.11 ± 0.12	90.59 ± 0.05	98.03 ± 0.02	98.16 ± 0.03	**98.39 ± 0.02**	N/A	N/A	N/A	98.22 ± 0.01
	Avg. Rank	9.25	10.33	8.42	7.17	6.83	N/A	10.67	8.42	4.50	3.58	2.17	N/A	N/A	N/A	**1.75**
hist	Wikipedia	68.69 ± 0.39	62.18 ± 1.27	84.17 ± 0.22	81.76 ± 0.32	67.27 ± 1.63	N/A	82.20 ± 2.18	**87.60 ± 0.30**	71.42 ± 4.43	N/A	71.28 ± 4.33	84.11 ± 1.88	82.78 ± 0.30	N/A	74.48 ± 2.12
	Reddit	62.34 ± 0.54	61.60 ± 0.72	63.47 ± 0.36	64.85 ± 0.85	63.67 ± 0.41	N/A	60.83 ± 0.25	64.50 ± 0.26	65.37 ± 0.60	N/A	62.15 ± 1.72	**66.14 ± 1.40**	66.02 ± 0.41	N/A	66.10 ± 0.34
	MOOC	63.22 ± 1.55	62.93 ± 1.24	76.73 ± 0.29	77.07 ± 3.41	74.68 ± 0.68	N/A	74.27 ± 0.53	74.00 ± 0.97	80.82 ± 0.30	N/A	81.85 ± 1.60	83.19 ± 1.72	81.63 ± 0.33	N/A	**86.15 ± 0.44**
	LastFM	70.39 ± 4.31	71.45 ± 1.76	76.27 ± 0.25	66.65 ± 6.11	71.33 ± 0.47	N/A	65.78 ± 0.65	76.42 ± 0.22	76.35 ± 0.52	N/A	**82.27 ± 1.22**	81.12 ± 0.30	77.28 ± 0.21	N/A	81.68 ± 0.46
	Enron	65.86 ± 3.71	62.08 ± 2.27	61.40 ± 1.31	62.91 ± 1.16	60.70 ± 0.36	N/A	67.11 ± 0.62	72.37 ± 1.37	67.07 ± 0.62	N/A	74.60 ± 1.35	**82.86 ± 0.47**	73.01 ± 0.88	N/A	72.92 ± 3.20
	Social Evo.	88.51 ± 0.87	88.72 ± 1.10	93.97 ± 0.54	90.66 ± 1.62	79.83 ± 0.38	N/A	94.10 ± 0.31	94.01 ± 0.47	96.82 ± 0.16	N/A	96.38 ± 0.18	N/A	96.69 ± 0.14	N/A	**97.01 ± 0.17**
	UCI	63.11 ± 2.27	52.47 ± 2.06	70.52 ± 0.93	70.78 ± 0.78	64.54 ± 0.47	N/A	76.71 ± 1.00	81.66 ± 0.49	72.13 ± 1.87	N/A	78.48 ± 1.18	**85.52 ± 0.16**	82.35 ± 0.39	N/A	82.61 ± 0.14
	Flights	61.01 ± 1.65	62.83 ± 1.31	64.72 ± 0.36	59.31 ± 1.43	56.82 ± 0.57	N/A	64.50 ± 0.25	**65.28 ± 0.24**	57.11 ± 0.21	N/A	54.67 ± 0.76	N/A	N/A	N/A	58.33 ± 1.35
	Can. Parl.	52.60 ± 0.88	52.28 ± 0.31	56.72 ± 0.47	54.42 ± 0.77	57.14 ± 0.07	N/A	55.71 ± 0.74	55.84 ± 0.73	87.40 ± 0.85	N/A	68.97 ± 1.60	N/A	N/A	N/A	**88.16 ± 0.41**

(Continued)

Table A14 (continued)

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
	US Legis.	52.94 ± 2.11	62.10 ± 1.41	51.83 ± 3.95	61.18 ± 1.10	55.56 ± 1.71	N/A	53.87 ± 1.41	52.03 ± 1.02	56.31 ± 3.46	N/A	**66.95 ± 1.81**	N/A	N/A	N/A	56.40 ± 3.35
	UN Trade	55.46 ± 1.19	55.49 ± 0.84	55.28 ± 0.71	52.80 ± 3.19	55.00 ± 0.38	N/A	55.76 ± 1.05	54.94 ± 0.97	53.20 ± 1.07	N/A	**78.83 ± 0.53**	N/A	N/A	N/A	53.48 ± 2.66
	Contact	90.42 ± 2.34	89.22 ± 0.66	**94.15 ± 0.45**	88.13 ± 1.50	74.20 ± 0.80	N/A	90.44 ± 0.17	89.91 ± 0.36	93.56 ± 0.52	N/A	93.56 ± 0.57	N/A	N/A	N/A	93.58 ± 0.30
	Avg. Rank	8.42	8.42	6.17	7.42	8.58	N/A	6.58	5.58	5.58	N/A	4.33	N/A	N/A	N/A	**3.42**
ind	Wikipedia	68.70 ± 0.39	62.19 ± 1.28	84.17 ± 0.22	81.77 ± 0.32	67.24 ± 1.63	N/A	82.20 ± 2.18	**87.60 ± 0.29**	71.42 ± 4.43	71.14 ± 2.44	67.95 ± 2.77	84.10 ± 1.88	87.54 ± 0.26	N/A	74.50 ± 2.15
	Reddit	62.32 ± 0.54	61.58 ± 0.72	63.40 ± 0.36	64.84 ± 0.84	63.65 ± 0.41	N/A	60.81 ± 0.26	64.49 ± 0.25	65.35 ± 0.60	65.30 ± 1.05	62.37 ± 0.83	**66.13 ± 1.40**	64.98 ± 0.20	N/A	65.40 ± 0.63
	MOOC	63.22 ± 1.55	62.92 ± 1.24	76.72 ± 0.30	77.07 ± 3.40	74.69 ± 0.68	N/A	74.28 ± 0.53	73.99 ± 0.97	80.82 ± 0.30	80.75 ± 1.00	84.46 ± 0.88	83.19 ± 1.72	81.41 ± 0.31	N/A	**86.11 ± 0.18**
	LastFM	70.39 ± 4.31	71.45 ± 1.75	76.28 ± 0.25	69.46 ± 4.65	71.33 ± 0.47	N/A	65.78 ± 0.65	76.42 ± 0.22	76.35 ± 0.52	76.76 ± 0.43	77.10 ± 0.78	81.12 ± 0.30	77.01 ± 0.43	N/A	**81.68 ± 0.45**
	Enron	65.86 ± 3.71	62.08 ± 2.27	61.40 ± 1.30	62.90 ± 1.16	60.72 ± 0.36	N/A	67.11 ± 0.62	72.37 ± 1.38	67.07 ± 0.62	68.77 ± 0.60	74.50 ± 1.02	**82.87 ± 0.47**	72.85 ± 0.81	N/A	72.88 ± 2.21
	Social Evo.	88.51 ± 0.87	88.72 ± 1.10	93.97 ± 0.54	90.65 ± 1.62	79.83 ± 0.39	N/A	94.10 ± 0.32	94.01 ± 0.47	96.82 ± 0.17	96.83 ± 0.56	96.76 ± 0.14	N/A	96.91 ± 0.12	N/A	**96.98 ± 0.08**
	UCI	63.16 ± 2.27	52.47 ± 2.09	70.49 ± 0.93	70.73 ± 0.79	64.54 ± 0.47	N/A	76.65 ± 0.99	81.64 ± 0.49	72.13 ± 1.86	72.17 ± 2.20	71.35 ± 0.84	**85.53 ± 0.16**	82.06 ± 0.58	N/A	82.64 ± 0.14
	Flights	61.01 ± 1.66	62.83 ± 1.31	64.72 ± 0.37	59.32 ± 1.45	56.82 ± 0.56	N/A	64.50 ± 0.25	**65.29 ± 0.24**	57.11 ± 0.20	57.76 ± 2.06	53.08 ± 0.87	N/A	N/A	N/A	57.15 ± 2.03
	Can. Parl.	52.58 ± 0.86	52.24 ± 0.28	56.46 ± 0.50	54.18 ± 0.73	57.06 ± 0.08	N/A	55.46 ± 0.69	55.76 ± 0.65	87.22 ± 0.82	**92.68 ± 0.97**	69.11 ± 1.18	N/A	N/A	N/A	88.21 ± 0.45
	US Legis.	52.94 ± 2.11	62.10 ± 1.41	51.83 ± 3.95	61.18 ± 1.10	55.56 ± 1.71	N/A	53.87 ± 1.41	52.03 ± 1.02	56.31 ± 3.46	57.85 ± 0.23	**68.37 ± 1.62**	N/A	N/A	N/A	56.40 ± 3.38
	UN Trade	55.43 ± 1.20	55.42 ± 0.87	55.58 ± 0.68	52.80 ± 3.24	54.97 ± 0.38	N/A	55.66 ± 0.98	54.88 ± 1.01	52.56 ± 1.70	52.81 ± 0.18	**80.70 ± 1.03**	N/A	N/A	N/A	53.46 ± 2.60
	Contact	90.43 ± 2.33	89.22 ± 0.65	**94.14 ± 0.45**	88.12 ± 1.50	74.19 ± 0.81	N/A	90.43 ± 0.17	89.91 ± 0.36	93.55 ± 0.52	94.05 ± 0.32	93.47 ± 0.43	N/A	N/A	N/A	93.59 ± 0.32
	Avg. Rank	9.00	9.17	6.58	8.00	9.33	N/A	7.17	6.00	6.17	5.08	5.25	N/A	N/A	N/A	**3.58**

Table A15: ROC-AUC for inductive dynamic link prediction with random, historical, and inductive negative sampling strategies. Bold red values denote the best performance, blue underlined values represent the second-best performance, and N/A means unavailable data or unused method.

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
rnd	Wikipedia	94.33 ± 0.27	91.49 ± 0.45	95.90 ± 0.09	97.72 ± 0.03	98.03 ± 0.04	N/A	95.57 ± 0.20	96.30 ± 0.04	98.48 ± 0.03	98.72 ± 0.03	98.90 ± 0.01	98.63 ± 0.02	99.01 ± 0.02	N/A	**99.05 ± 0.01**
	Reddit	96.52 ± 0.13	96.05 ± 0.12	96.98 ± 0.04	97.39 ± 0.07	98.42 ± 0.02	N/A	93.80 ± 0.07	94.97 ± 0.05	98.71 ± 0.01	98.88 ± 0.01	98.68 ± 0.02	98.73 ± 0.03	98.84 ± 0.01	N/A	**98.91 ± 0.01**
	MOOC	83.16 ± 1.30	84.03 ± 0.49	86.84 ± 0.17	91.24 ± 0.99	81.86 ± 0.25	N/A	81.43 ± 0.19	82.77 ± 0.24	87.62 ± 0.51	89.34 ± 0.12	95.55 ± 0.25	94.10 ± 0.07	87.01 ± 0.74	N/A	**96.87 ± 0.38**
	LastFM	81.13 ± 3.39	82.24 ± 1.51	76.99 ± 0.29	82.61 ± 3.15	87.82 ± 0.12	N/A	70.84 ± 0.85	80.37 ± 0.18	94.08 ± 0.08	94.37 ± 0.13	95.36 ± 0.06	95.10 ± 0.03	94.32 ± 0.03	N/A	**96.45 ± 0.01**
	Enron	81.96 ± 1.34	76.34 ± 4.20	64.63 ± 1.74	78.83 ± 1.11	87.02 ± 0.50	N/A	72.33 ± 0.99	76.51 ± 0.71	90.69 ± 0.26	89.76 ± 0.21	90.21 ± 0.49	89.04 ± 0.26	89.51 ± 0.20	N/A	**90.71 ± 0.50**
	Social Evo.	93.70 ± 0.29	91.18 ± 0.49	93.41 ± 0.19	93.43 ± 0.59	84.73 ± 0.27	N/A	93.71 ± 0.18	94.09 ± 0.07	95.29 ± 0.03	95.36 ± 0.04	95.47 ± 0.04	N/A	**96.41 ± 0.07**	N/A	95.45 ± 0.01
	UCI	78.80 ± 0.94	58.08 ± 1.81	77.64 ± 0.38	86.68 ± 2.29	90.40 ± 0.11	N/A	84.49 ± 1.82	89.30 ± 0.57	92.63 ± 0.13	92.70 ± 0.19	94.40 ± 0.03	93.54 ± 0.14	93.01 ± 0.08	N/A	**94.47 ± 0.07**
	Flights	95.21 ± 0.32	93.56 ± 0.70	88.64 ± 0.35	95.92 ± 0.43	96.86 ± 0.02	N/A	82.48 ± 0.01	82.27 ± 0.06	97.80 ± 0.02	**97.98 ± 0.25**	98.05 ± 0.04	N/A	N/A	N/A	97.93 ± 0.01
	Can. Parl.	53.81 ± 1.14	55.27 ± 0.49	56.51 ± 0.75	55.86 ± 0.75	58.83 ± 1.13	N/A	55.83 ± 1.07	58.32 ± 1.08	89.33 ± 0.48	**94.02 ± 3.02**	69.21 ± 1.31	N/A	N/A	N/A	90.84 ± 0.58
	US Legis.	58.12 ± 2.35	61.07 ± 0.56	48.27 ± 3.50	62.38 ± 0.48	51.49 ± 1.13	N/A	50.43 ± 1.48	47.20 ± 0.89	53.21 ± 3.04	57.17 ± 0.20	**65.29 ± 0.61**	N/A	N/A	N/A	50.94 ± 1.12
	UN Trade	62.28 ± 0.50	58.82 ± 0.98	62.72 ± 0.12	59.99 ± 3.50	67.05 ± 0.21	N/A	63.76 ± 0.07	63.48 ± 0.37	67.25 ± 1.05	68.26 ± 0.26	**86.88 ± 0.23**	N/A	N/A	N/A	69.48 ± 0.19
	Contact	95.37 ± 0.92	91.89 ± 0.38	96.53 ± 0.10	94.84 ± 0.75	89.07 ± 0.34	N/A	93.05 ± 0.09	92.83 ± 0.05	98.30 ± 0.02	98.44 ± 0.05	98.51 ± 0.01	N/A	N/A	N/A	**98.53 ± 0.01**
	Avg. Rank	8.83	9.83	9.42	8.17	8.25	N/A	10.08	9.25	5.17	3.83	2.42	N/A	N/A	N/A	**2.00**
hist	Wikipedia	61.86 ± 0.53	57.54 ± 1.09	78.38 ± 0.20	75.75 ± 0.29	62.04 ± 0.65	N/A	79.79 ± 0.96	**82.87 ± 0.21**	68.33 ± 2.82	N/A	67.95 ± 2.77	77.58 ± 1.00	82.08 ± 0.32	N/A	68.21 ± 1.63
	Reddit	61.69 ± 0.39	60.45 ± 0.37	64.43 ± 0.27	64.55 ± 0.50	64.94 ± 0.21	N/A	61.43 ± 0.26	64.27 ± 0.13	64.81 ± 0.25	N/A	62.37 ± 0.83	65.15 ± 0.36	**66.79 ± 0.31**	N/A	65.44 ± 0.84
	MOOC	64.48 ± 1.64	64.23 ± 1.29	74.08 ± 0.27	77.69 ± 3.55	71.68 ± 0.94	N/A	69.82 ± 0.32	72.53 ± 0.84	80.77 ± 0.63	N/A	84.46 ± 0.88	82.68 ± 1.28	81.52 ± 0.37	N/A	**88.10 ± 0.73**
	LastFM	68.44 ± 3.26	68.79 ± 1.08	69.89 ± 0.28	66.99 ± 5.62	67.69 ± 0.24	N/A	55.88 ± 1.85	70.07 ± 0.20	70.73 ± 0.37	N/A	**77.10 ± 0.78**	73.48 ± 0.36	72.63 ± 0.16	N/A	75.33 ± 0.19
	Enron	65.32 ± 3.57	61.50 ± 2.50	57.84 ± 2.18	62.68 ± 1.09	62.25 ± 0.40	N/A	64.06 ± 1.02	68.20 ± 1.62	65.78 ± 0.42	N/A	74.50 ± 1.02	**78.84 ± 0.29**	70.09 ± 0.65	N/A	70.77 ± 1.20
	Social Evo.	88.53 ± 0.55	87.93 ± 1.05	91.87 ± 0.72	92.10 ± 1.22	83.54 ± 0.24	N/A	93.28 ± 0.60	93.62 ± 0.35	96.91 ± 0.09	N/A	96.76 ± 0.14	N/A	96.94 ± 0.17	N/A	**97.01 ± 0.15**
	UCI	60.24 ± 1.94	51.25 ± 2.37	62.32 ± 1.18	62.69 ± 0.90	56.39 ± 0.10	N/A	70.46 ± 1.94	75.98 ± 0.84	65.55 ± 1.01	N/A	71.35 ± 0.84	76.91 ± 0.23	76.01 ± 0.75	N/A	**76.97 ± 1.21**
	Flights	60.72 ± 1.29	61.99 ± 1.39	63.38 ± 0.26	59.66 ± 1.04	56.58 ± 0.44	N/A	**63.48 ± 0.23**	63.30 ± 0.19	56.05 ± 0.21	N/A	53.08 ± 0.87	N/A	N/A	N/A	56.10 ± 0.45
	Can. Parl.	51.62 ± 1.00	52.38 ± 0.46	58.30 ± 0.61	55.64 ± 0.54	60.11 ± 0.48	N/A	57.30 ± 1.03	56.68 ± 1.20	**88.68 ± 0.74**	N/A	68.98 ± 1.21	N/A	N/A	N/A	88.12 ± 0.34

(Continued)

Table A15 (continued)

NSS	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	DyGFormer	DyG-Mamba	TPNet	RepeatMixer	FreeDyG	ScaDyG	DyG-Hyena
	US Legis.	58.12 ± 2.94	67.94 ± 0.98	49.99 ± 4.88	64.87 ± 1.65	54.41 ± 1.31	N/A	52.12 ± 2.13	49.28 ± 0.86	56.57 ± 3.22	N/A	**68.37 ± 1.62**	N/A	N/A	N/A	56.77 ± 3.40
	UN Trade	58.73 ± 1.19	57.90 ± 1.33	59.74 ± 0.59	55.61 ± 3.54	60.95 ± 0.80	N/A	61.12 ± 0.97	59.88 ± 1.17	58.46 ± 1.65	N/A	**80.70 ± 1.03**	N/A	N/A	N/A	58.69 ± 2.46
	Contact	90.80 ± 1.18	88.88 ± 0.68	93.76 ± 0.41	88.84 ± 1.39	74.79 ± 0.37	N/A	90.37 ± 0.16	90.04 ± 0.29	94.14 ± 0.26	N/A	93.47 ± 0.43	N/A	N/A	N/A	**94.20 ± 0.30**
	Avg. Rank	7.08	7.42	5.92	6.33	6.92	N/A	5.33	5.00	4.08	N/A	3.75	N/A	N/A	N/A	**3.33**
ind	Wikipedia	61.87 ± 0.53	57.54 ± 1.09	78.38 ± 0.20	75.76 ± 0.29	62.02 ± 0.65	N/A	79.79 ± 0.96	**82.88 ± 0.21**	68.33 ± 2.82	67.92 ± 2.23	67.96 ± 2.77	77.58 ± 1.00	83.17 ± 0.31	N/A	68.22 ± 1.63
	Reddit	61.69 ± 0.39	60.44 ± 0.37	64.39 ± 0.27	64.55 ± 0.50	64.91 ± 0.21	N/A	61.36 ± 0.26	64.27 ± 0.13	64.80 ± 0.25	64.93 ± 0.89	62.37 ± 0.83	65.15 ± 0.36	64.51 ± 0.19	N/A	**65.40 ± 0.80**
	MOOC	64.48 ± 1.64	64.22 ± 1.29	74.07 ± 0.27	77.68 ± 3.55	71.69 ± 0.94	N/A	69.83 ± 0.32	72.52 ± 0.84	80.77 ± 0.63	81.12 ± 0.63	84.46 ± 0.88	82.68 ± 1.28	75.81 ± 0.69	N/A	**88.10 ± 0.48**
	LastFM	68.44 ± 3.26	68.79 ± 1.08	69.89 ± 0.28	66.99 ± 5.61	67.68 ± 0.24	N/A	55.88 ± 1.85	70.07 ± 0.20	70.73 ± 0.37	70.59 ± 0.57	**77.10 ± 0.78**	73.48 ± 0.36	71.42 ± 0.33	N/A	75.33 ± 0.27
	Enron	65.32 ± 3.57	61.50 ± 2.50	57.83 ± 2.18	62.68 ± 1.09	62.27 ± 0.40	N/A	64.05 ± 1.02	68.19 ± 1.63	65.79 ± 0.42	68.98 ± 1.00	74.50 ± 1.02	**78.85 ± 0.29**	68.79 ± 0.91	N/A	70.38 ± 1.50
	Social Evo.	88.53 ± 0.55	87.93 ± 1.05	91.88 ± 0.72	92.10 ± 1.22	83.54 ± 0.24	N/A	93.28 ± 0.60	93.62 ± 0.35	96.91 ± 0.09	96.65 ± 0.29	96.76 ± 0.14	N/A	96.79 ± 0.17	N/A	**97.01 ± 0.15**
	UCI	60.27 ± 1.94	51.26 ± 2.40	62.29 ± 1.17	62.66 ± 0.91	56.39 ± 0.11	N/A	70.42 ± 1.93	75.97 ± 0.85	65.58 ± 1.00	66.95 ± 2.22	71.37 ± 0.84	76.92 ± 0.23	73.41 ± 0.88	N/A	**76.99 ± 0.32**
	Flights	60.72 ± 1.29	61.99 ± 1.39	63.40 ± 0.26	59.66 ± 1.05	56.58 ± 0.44	N/A	**63.49 ± 0.23**	63.32 ± 0.19	56.58 ± 2.12	N/A	53.05 ± 0.87	N/A	N/A	N/A	57.11 ± 2.03
	Can. Parl.	51.61 ± 0.98	52.35 ± 0.52	58.15 ± 0.62	55.43 ± 0.42	60.01 ± 0.47	N/A	56.88 ± 0.93	56.63 ± 1.09	88.51 ± 0.73	**92.37 ± 0.18**	68.87 ± 1.68	N/A	N/A	N/A	88.56 ± 0.45
	US Legis.	58.12 ± 2.94	67.94 ± 0.98	49.99 ± 4.88	64.87 ± 1.65	54.41 ± 1.31	N/A	52.12 ± 2.13	49.28 ± 0.86	56.57 ± 3.22	57.91 ± 3.41	**68.37 ± 1.62**	N/A	N/A	N/A	56.77 ± 3.40
	UN Trade	58.71 ± 1.20	57.87 ± 1.36	59.98 ± 0.59	55.62 ± 3.59	60.88 ± 0.79	N/A	61.01 ± 0.93	59.71 ± 1.17	57.28 ± 3.06	57.58 ± 0.20	**80.78 ± 1.03**	N/A	N/A	N/A	58.66 ± 2.60
	Contact	90.80 ± 1.18	88.87 ± 0.67	93.76 ± 0.40	88.85 ± 1.39	74.79 ± 0.38	N/A	90.37 ± 0.16	90.04 ± 0.29	94.14 ± 0.26	**94.35 ± 0.29**	93.47 ± 0.43	N/A	N/A	N/A	94.19 ± 0.32
	Avg. Rank	8.83	9.75	7.17	8.00	8.83	N/A	7.25	6.42	5.92	5.00	4.50	N/A	N/A	N/A	**3.33**

Appendix D.5 Comparison Results of Training Time and Memory

Fig. A1 shows the training time and memory usage of Hynea alone and DyGFormer w/o patching in the Can. Parl.

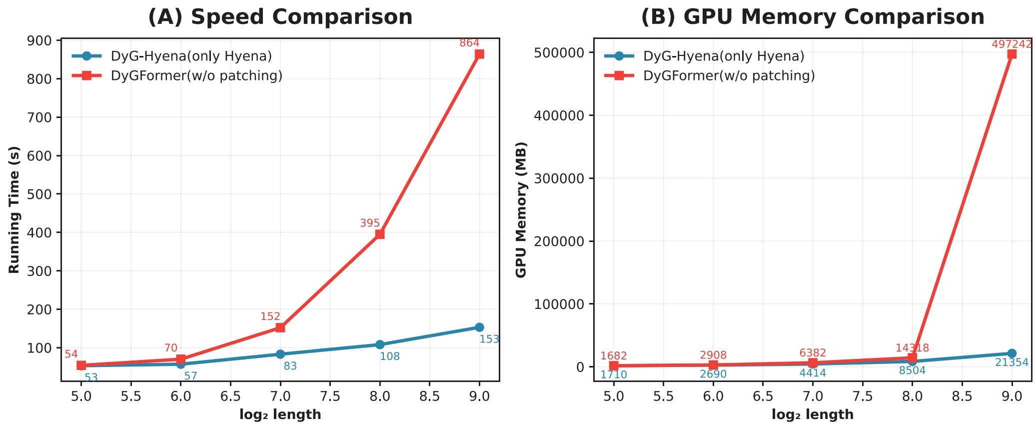


Figure A1: Illustration of the ablation experiment.

Appendix D.6 Additional Hardware Configuration, Benchmarks, Metrics

To enable standardized comparison with the baseline, this section reduces unfair comparisons arising from inconsistent conditions or assessment methods via experiments. Two sets of comparative experiments were conducted: (i) To mitigate unfairness caused by superior hardware in the main text, we re-evaluated the AP of transductive dynamic link prediction on four datasets (Wikipedia, Reddit, MOOC, UCI) using a machine with 56-core Intel(R) Xeon(R) Gold 6132 CPU @ 2.60 GHz and NVIDIA GeForce RTX 3090 24 G as shown in Table 1. (ii) We compared DyG-Hyena with baseline DyGFormer under more extensive datasets (tgbl-wiki [4, 52, 53], tgbl-coin(>600 K nodes, >20 million edges) [52, 53]) and stricter indicators (MRR) [52, 53] as shown in Table 2. Results show hardware conditions do not significantly improve performance, while broader datasets and stricter indicators confirm DyG-Hyena's competitiveness.

References

1. Kazemi SM, Goel R, Jain K, Kobyzev I, Sethi A, Forsyth P, et al. Representation learning for dynamic graphs: A survey. J Mach Learn Res. 2020;21(70):1–73.
2. Yang C. Dynamic graph representation learning with neural networks: A survey. IEEE Access. 2024;12(70):43460–84. doi:10.1109/access.2024.3378111.
3. Song W, Xiao Z, Wang Y, Charlin L, Zhang M, Tang J. Session-based social recommendation via dynamic graph attention networks. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining; 2019 Feb 11–15; Melbourne, Australia. p. 555–63.
4. Kumar S, Zhang X, Leskovec J. Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019 Aug 4–8; Anchorage, AK, USA. p. 1269–78.
5. Hui Y, Wang S, Chekol MW, Rudinac S, Zwetsloot IM. Non-progressive influence maximization in dynamic social networks. arXiv:2412.07402. 2024.
6. Yu L, Sun L, Du B, Lv W. Towards better dynamic graph learning: new architecture and unified library. Adv Neural Inf Process Syst. 2023;36:67686–700.
7. Poursafaei F, Huang S, Pelrine K, Rabbany R. Towards better evaluation for dynamic link prediction. Adv Neural Inf Process Syst. 2022;35:32928–41. doi:10.52202/068431-2386.

8. Rozemberczki B, Scherer P, He Y, Panagopoulos G, Riedel A, Astefanoaei M, et al. Pytorch geometric temporal: spatiotemporal signal processing with neural machine learning models. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management; 2021 Nov 1–5; Online. p. 4564–73.
9. Zhou H, Zheng D, Nisa I, Ioannidis V, Song X, Karypis G. TGL: a general framework for temporal GNN training on billion-scale graphs. arXiv:2203.14883. 2022.
10. Li D, Tan S, Zhang Y, Jin M, Pan S, Okumura M, et al. DyG-mamba: continuous state space modeling on dynamic graphs. arXiv:2408.06966. 2024.
11. Zhang J, Chen J, Yang M, Feng A, Liang S, Shao S, et al. DTGB: a comprehensive benchmark for dynamic text-attributed graphs. arXiv:2406.12072. 2024.
12. Tian Y, Qi Y, Guo F. Frequency enhanced continuous-time dynamic graph model for link prediction. In: Proceedings of the Twelfth International Conference on Learning Representations; 2024 May 7–11; Vienna, Austria.
13. Wu X, Li X, Li RH, Zhao K, Wang G. ScaDyG: a new paradigm for large-scale dynamic graph learning. arXiv:2501.16002. 2025.
14. Karmim Y, Lafon M, Fournier-S'Niehotta R, Thome N. Supra-laplacian encoding for transformer on dynamic graphs. Adv Neural Inf Process Syst. 2024;37:17215–46. doi:10.52202/079017-0547.
15. Chen C, Geng H, Yang N, Yang X, Yan J. EasyDGL: encode, train and interpret for continuous-time dynamic graph learning. IEEE Trans Pattern Anal Mach Intell. 2024;46(12):10845–62. doi:10.1109/tpami.2024.3443110.
16. Sun J, Gu M, Yeh CCM, Fan Y, Chowdhary G, Zhang W. Dynamic graph node classification via time augmentation. In: Proceedings of the 2022 IEEE International Conference on Big Data (Big Data); 2022 Dec 17–20; Osaka, Japan. p. 800–5.
17. Xu D, Ruan C, Korpeoglu E, Kumar S, Achan K. Inductive representation learning on temporal graphs. arXiv:2002.07962. 2020.
18. Yu X, Liu Z, Zhang X, Fang Y. Node-time conditional prompt learning in dynamic graphs. arXiv:2405.13937. 2024.
19. Cong W, Zhang S, Kang J, Yuan B, Wu H, Zhou X, et al. Do we really need complicated model architectures for temporal networks? arXiv:2302.11636. 2023.
20. Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein M. Temporal graph networks for deep learning on dynamic graphs. arXiv:2006.10637.2020.
21. Wang Y, Chang YY, Liu Y, Leskovec J, Li P. Inductive representation learning in temporal networks via causal anonymous walks. arXiv:2101.05974. 2021.
22. Souza A, Mesquita D, Kaski S, Garg V. Provably expressive temporal graph networks. Adv Neural Inf Process Syst. 2022;35:32257–69. doi:10.52202/068431-2337.
23. Wang L, Chang X, Li S, Chu Y, Li H, Zhang W, et al. TCL: transformer-based dynamic graph modelling via contrastive learning. arXiv:2105.07944. 2021.
24. Zou T, Mao Y, Ye J, Du B. Repeat-aware neighbor sampling for dynamic graph learning. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2024 Aug 25–29; Barcelona, Spain. p. 4722–33.
25. Luo Y, Li P. Neighborhood-aware scalable temporal network representation learning. In: Proceedings of the First Learning on Graphs Conference (LoG 2022); 2022 Dec 9–12; Virtual. p. 1.
26. Lu X, Sun L, Zhu T, Lv W. Improving temporal link prediction via temporal walk matrix projection. Adv Neural Inf Process Syst. 2024;37:141153–82. doi:10.52202/079017-4482.
27. Wu Y, Fang Y, Liao L. On the feasibility of simple transformer for dynamic graph modeling. In: Proceedings of the ACM Web Conference 2024; 2024 May 13–17; Singapore. p. 870–80.
28. Chen H, Jiao P, Du M, Guo X, Zhao Z, Jin D, et al. TGFormer: towards temporal graph transformer with auto-correlation mechanism. Pattern Recognit. 2026;170:112053.
29. Ding Z, Li Y, He Y, Norelli A, Wu J, Tresp V, et al. DyGMamba: efficiently modeling long-term temporal dependency on continuous-time dynamic graphs with state space models. arXiv:2408.04713. 2024.

30. Poli M, Massaroli S, Nguyen E, Fu DY, Dao T, Baccus S, et al. Hyena hierarchy: towards larger convolutional language models. In: Proceedings of the 40th International Conference on Machine Learning; 2023 Jul 23–29; Honolulu, HI, USA. p. 28043–78.
31. Fu DY, Epstein EL, Nguyen E, Thomas AW, Zhang M, Dao T, et al. Simple hardware-efficient long convolutions for sequence modeling. In: Proceedings of the 40th International Conference on Machine Learning; 2023 Jul 23–29; Honolulu, HI, USA. p. 10373–91.
32. Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models. Adv Neural Inf Process Syst. 2015;28:1–9.
33. Pandey AP, Varghese AJ, Patil S, Xu M. A comparative study on dynamic graph embedding based on mamba and transformers. arXiv:2412.11293. 2024.
34. You J, Du T, Leskovec J. ROLAND: graph learning framework for dynamic graphs. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2022 Aug 14–18; Washington, DC, USA. p. 2358–66.
35. Cong W, Wu Y, Tian Y, Gu M, Xia Y, Mahdavi M, et al. Dynamic graph representation learning via graph transformer networks. arXiv.2111.10447. 2021.
36. Sankar A, Wu Y, Gou L, Zhang W, Yang H. DySAT: deep neural representation learning on dynamic graphs via self-attention networks. In: Proceedings of the 13th International Conference on Web Search and Data Mining; 2020 Feb 3–7; Houston, TX, USA. p. 519–27.
37. Pareja A, Domeniconi G, Chen J, Ma T, Suzumura T, Kanezashi H, et al. EvolveGCN: evolving graph convolutional networks for dynamic graphs. Proc AAAI Conf Artif Intell. 2020;34(04):5363–70.
38. Xiong J, Zareie A, Sakellariou R. A survey of link prediction in temporal networks. arXiv:2502.21185. 2025.
39. Ma Y, Guo Z, Ren Z, Tang J, Yin D. Streaming graph neural networks. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020 Jul 25–30; Virtual. p. 719–28.
40. Wang X, Lyu D, Li M, Xia Y, Yang Q, Wang X, et al. APAN: asynchronous propagation attention network for real-time temporal graph embedding. In: Proceedings of the 2021 International Conference on Management of Data; 2021 Jun 20–25; Virtual. p. 2628–38.
41. Trivedi R, Farajtabar M, Biswal P, Zha H. DyRep: learning representations over dynamic graphs. In: Proceedings of the International Conference on Learning Representations; 2019 May 6–9; New Orleans, LA, USA.
42. Fu DY, Dao T, Saab KK, Thomas AW, Rudra A, Ré C. Hungry hungry hippos: towards language modeling with state space models. arXiv:2212.14052. 2022.
43. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst. 2017;30:6000–10. doi:10.65215/pc26a033.
44. Moskalev A, Prakash M, Xu J, Cui T, Liao R, Mansi T. Geometric hyena networks for large-scale equivariant learning. arXiv:2505.22560. 2025.
45. Gu A, Dao T. Linear-time sequence modeling with selective state spaces. In: Proceedings of the First Conference on Language Modeling; 2024 Oct 7–9; Philadelphia, PA, USA.
46. Peng B, Alcaide E, Anthony Q, Albalak A, Arcadinho S, Biderman S, et al. RWKV: reinventing rnns for the transformer era. arXiv:2305.13048. 2023.
47. Pennebaker JW, Francis ME, Booth RJ. Linguistic inquiry and word count: lIWC 2001. Mahway Lawrence Erlbaum Assoc. 2001;71(2001):206–29. doi:10.4018/978-1-60960-741-8.ch012.
48. Shetty J, Adibi J. The Enron email dataset database schema and brief statistical report. Inf Sci Inst Tech Rep Univ South Calif. 2004;4(1):120–8.
49. Madan A, Cebrian M, Moturu S, Farrahi K. Sensing the "health state" of a community. IEEE Pervasive Comput. 2011;11(4):36–45. doi:10.1109/mprv.2011.79.
50. Panzarasa P, Opsahl T, Carley KM. Patterns and dynamics of users' behavior and interaction: network analysis of an online community. J Am Soc Inf Sci Technol. 2009;60(5):911–32.
51. Strohmeier M, Olive X, Lübbe J, Schäfer M, Lenders V. Crowdsourced air traffic data from the OpenSky Network 2019–20. Earth Syst Sci Data. 2021;13(2):357–66. doi:10.5194/essd-13-357-2021.

52. Huang S, Poursafaei F, Danovitch J, Fey M, Hu W, Rossi E, et al. Temporal graph benchmark for machine learning on temporal graphs. In: Proceedings of the 37th International Conference on Neural Information Processing Systems; 2023 Dec 10–16; New Orleans, LA, USA.
53. Gastinger J, Huang S, Galkin M, Loghmani E, Parviz A, Poursafaei F, et al. TGB 2.0: a benchmark for learning on temporal knowledge graphs and heterogeneous graphs. In: Proceedings of the 38th International Conference on Neural Information Processing Systems; 2024 Dec 10–15; Vancouver, BC, Canada.