S3Former: Sequential, Structural, and Statistical Fusion for Continuous-Time Dynamic Graphs

Ziyan Li[1], Xiangyu Hu[1]*, Tianqing Zhu[2], Wanlun Ma[3], Ke Qin[1]

[1]The Institute of Intelligent Computing, University of Electronic Science and Technology of China, Chengdu 611731, China

[2]City University of Macau

[3]Independent Researcher

Abstract

Continuous-time dynamic graphs model evolving systems with irregularly timed interactions, where predicting future links requires reasoning over temporal evolution, local topology, and historical relation patterns. Existing memory- and sequence-based methods capture temporal dependencies effectively, but organizing historical interactions as node memories or serialized neighbor sequences can weaken the explicit connectivity among candidate-centered historical context nodes. We propose S3Former, a temporal graph framework that jointly models sequential dynamics, shared pair-centered structure, and online statistical memory. For each candidate interaction, S3Former encodes long- and short-term historical sequences, constructs a history-induced shared pair subgraph around the two endpoints, and computes node- and pair-level statistics from past events only. A two-level context-aware fusion mechanism further combines these signals: NodeCCF learns reliable endpoint representations, while PairCCF performs candidate-specific relation correction using explicit pair memory. Experiments on 9 dynamic graph benchmarks show that S3Former achieves the strongest average rank across the four standard transductive and inductive AP/AUC evaluation settings. Ablation studies confirm the effectiveness of shared pair-subgraph modeling, online statistical memory, and hierarchical fusion. The source code is available at `https://github.com/lzy4123/S3Former`.

1 Introduction

Existing continuous-time dynamic graph methods can be broadly grouped into memory-based and sequence-based paradigms. Memory-based models, such as JODIE [7], DyRep [15], and TGN [13], maintain evolving node states through recurrent updates or memory modules over the event stream. Sequence-based models, such as TGAT [19] and DyGFormer [20], construct historical neighbor sequences and use attention or Transformer-style encoders to capture long-range temporal dependencies. These designs have substantially improved temporal modeling for irregular interaction histories. However, serializing historical events may weaken the explicit connectivity among historical context nodes associated with a candidate pair.

This limitation is important because dynamic graph learning should not be reduced to temporal sequence modeling alone. A key insight from recent Graph Transformer studies is that strong graph models need mechanisms to learn both structural-topological features and long-range dependencies [12]. In particular, GraphGPS shows that local message passing and global attention play complementary roles: message passing helps the model aggregate information along real edges and

*Corresponding author: `huxiangyu@uestc.edu.cn`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

learn local structural patterns, while attention captures broader dependencies beyond local neighborhoods. Inspired by this principle, we argue that continuous-time dynamic graph models should also preserve a structural learning mechanism at each prediction time, rather than relying only on serialized historical interactions.

To address this limitation, we propose S3Former (**S**equential, **S**tructural, and **S**tatistical **Former**), a unified framework for continuous-time dynamic link prediction. S3Former combines long-range sequence modeling, short-term dynamics, causal local structural learning, and online statistical memory. For each candidate interaction, S3Former first encodes the long-term and short-term historical sequences of the two endpoints. It then constructs a history-induced shared pair subgraph from past interactions only and applies a lightweight edge-aware GraphSAGE module to learn the local topology around the candidate pair. In parallel, S3Former computes causal node- and pair-level statistics, including activity, recency, repeated interaction strength, and neighborhood overlap, to provide robust relation memory without future information leakage. The sequential channel preserves temporal order and long-range endpoint evolution, the structural channel exposes candidate-conditioned historical connectivity, and the statistical channel retains cumulative activity, recency, repeated-interaction, and overlap evidence. Together, they provide complementary causal summaries of the same pre-query interaction history.

At the core of S3Former is a two-level context-aware collaborative fusion mechanism. NodeCCF adaptively combines sequential, structural, and node-statistical representations to obtain reliable endpoint embeddings. PairCCF further refines the candidate-pair representation using explicit pair memory, enabling relation-specific correction beyond endpoint state modeling. This hierarchical design allows structural and statistical evidence to contribute when informative, while retreating when the corresponding subgraph or pair history is sparse or noisy.

Our main contributions are summarized as follows:

- We identify a structural limitation of sequence-first continuous-time dynamic graph models: serialization captures temporal order but can obscure explicit local connectivity among historical nodes. We propose S3Former, a unified framework that jointly models sequential dynamics, shared pair-centered structure, and causal statistical memory.
- We introduce a history-induced shared pair-subgraph encoder that applies edge-aware message passing to recover local patterns such as common-neighbor bridges, repeated local ties, and triadic closure under strict temporal causality.
- We design online node- and pair-level statistical memory together with a two-level fusion mechanism, NodeCCF and PairCCF, to adaptively coordinate endpoint state modeling and candidate-specific relation correction.
- Experiments on nine representative CTDG benchmarks show that S3Former achieves the best average rank across all four standard transductive and inductive AP/AUC settings. Ablation studies further validate the complementary roles of structural encoding, statistical memory, and hierarchical fusion, with additional appendix analyses on recurrence-heavy institutional-relation datasets.

2 Related Work

2.1 Continuous-Time Dynamic Graph Learning

Dynamic link prediction on continuous-time dynamic graphs aims to infer future interactions from temporally ordered event streams. Early dynamic graph methods often discretize graph evolution into snapshots and apply static graph models or recurrent architectures over the snapshot sequence [14]. While snapshot-based formulations are suitable for coarse-grained temporal evolution, they may lose fine-grained event-level dynamics when interactions occur asynchronously and irregularly.

Event-based continuous-time methods directly model timestamped interactions. A major line of work follows the memory-based paradigm. Models such as JODIE [7], DyRep [15], and TGN [13] maintain evolving node states through recurrent updates or explicit memory modules. These methods are naturally suited to streaming graphs, since node representations can be updated online as new events arrive. However, recursive memory updates may struggle to preserve long-range historical information, especially when interaction histories are long, irregular, or highly heterogeneous.

Another influential line follows the sequence-based paradigm. Methods such as TGAT [19], CAWN [18], TCL [17], GraphMixer [3], and DyGFormer [20] construct historical neighbor sequences and encode them with temporal attention, sequence models, or Transformer-style architectures. These approaches can exploit longer interaction histories more directly than recurrent memory updates and have achieved strong performance on dynamic link prediction benchmarks. In particular, DyGFormer demonstrates the effectiveness of treating continuous-time dynamic graph learning as historical interaction sequence modeling. Our work builds on this sequence-based progress but revisits a key limitation: sequence modeling alone does not explicitly preserve the local connectivity among historical nodes.

2.2 Structural Modeling in Dynamic Graphs

Structural information is central to graph learning, since link formation often depends on local connectivity patterns such as common neighbors, triadic closure, repeated local ties, and neighborhood density [8]. In continuous-time dynamic graphs, existing methods incorporate structure in different ways. Memory-based models often rely on neighborhood aggregation or temporal attention over sampled neighbors, which provides implicit structural signals but does not explicitly reconstruct the local topology associated with a candidate interaction [21]. Random-walk-based methods such as CAWN [18] encode causal anonymous walks and capture richer temporal-structural context, but the resulting structure is represented through sampled walks rather than an induced local subgraph.

Recent methods also introduce specialized structural encodings. DyGFormer [20] and CNE-N [2] use neighbor co-occurrence information to enhance relative structural awareness, SALoM [**?**] incorporates co-occurrence-based structural memory, TPNet [9] projects temporal-walk matrices, and TNCN [**?**] constructs temporal common-neighbor representations. These approaches demonstrate that prior CTDG methods capture structural evidence through temporal walks, neighbor co-occurrence, common-neighbor signals, and related structural encodings. S3Former differs by explicitly modeling historical connectivity among selected candidate-centered context nodes and combining it with sequential and statistical evidence.

S3Former addresses this gap by constructing a causal shared pair subgraph for each candidate interaction. Unlike co-occurrence-only encodings, the structural channel induces historical edges among selected endpoint neighbors and common neighbors using interactions before the prediction time only. This allows lightweight message passing to recover explicit local topology while preserving temporal causality.

2.3 Statistical Memory and Adaptive Fusion

Statistical signals provide complementary evidence for dynamic link prediction. Simple quantities such as node activity, pair frequency, recency, temporal decay, and neighborhood overlap can be highly informative, especially when repeated interactions or stable relations dominate the graph. EdgeBank [11] shows that memory-based historical interaction heuristics can be strong baselines under certain evaluation settings. Other dynamic graph models also use temporal or neighborhood statistics to improve representation learning.

However, statistical signals must be handled carefully in continuous-time prediction. If statistics are computed from global or static tables, they may implicitly include information unavailable at the prediction time, leading to temporal leakage. In addition, statistics tied to fixed node identities or fully observed histories may generalize poorly in inductive settings, where new nodes or sparse-history samples appear. Therefore, statistical memory should be computed online from causal history only and updated after prediction at each timestamp.

Another related issue is how temporal, structural, and statistical signals are fused. Many existing methods combine multiple signals through concatenation, linear projection, attention, or shallow gating. Such fusion strategies often treat different sources of evidence as uniformly reliable across samples. In practice, the usefulness of each signal is context-dependent: structural evidence may be unreliable when the induced subgraph is sparse, while pair statistics may be informative only for repeated or stable relations. SALoM [**?**] highlights the importance of balancing temporal and structural signals through a more careful fusion mechanism, but its fusion is mainly designed for temporal memory and co-occurrence-based structure.

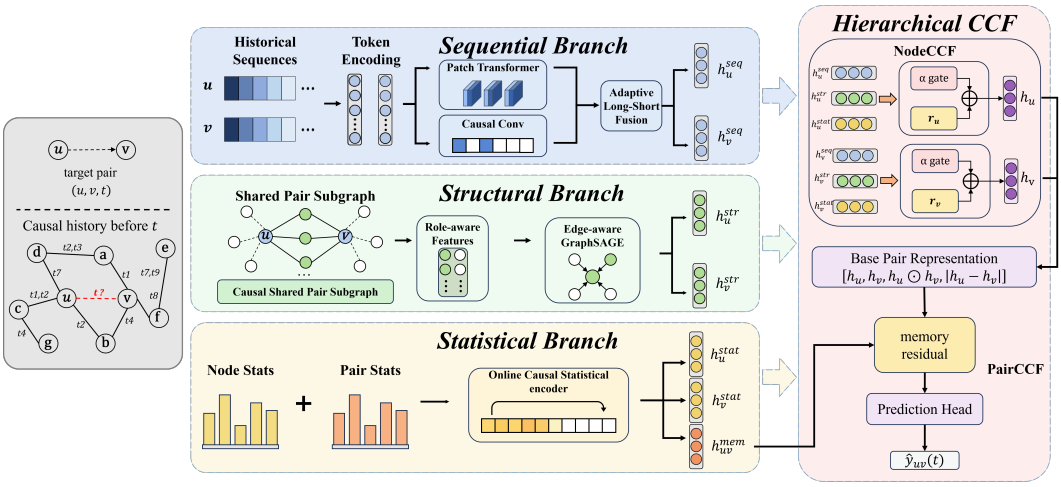


Figure 1: Overall pipeline of S3Former. Sequential, structural, and statistical encoders produce complementary causal representations, which are coordinated by NodeCCF and PairCCF before the final link prediction head.

S3Former instead introduces a two-level context-aware collaborative fusion mechanism. NodeCCF adaptively coordinates sequential, structural, and node-statistical representations at the endpoint level, while PairCCF performs candidate-specific relation correction using causal pair statistics and pair-time context. This two-level design makes the contribution of structural and statistical signals sample-dependent rather than fixed across candidate pairs.

3 Method

3.1 Problem Formulation

We represent a continuous-time dynamic graph as a chronologically ordered interaction stream $\mathcal{G} = \{(u_i, v_i, t_i, \mathbf{e}_i)\}_{i=1}^{N}$, where u_i and v_i are the two endpoints of the i-th interaction, t_i is the timestamp, and $\mathbf{e}_i$ denotes the optional edge feature. Given a candidate interaction (u, v, t), dynamic link prediction estimates whether u and v will interact at time t using only the causal history $\mathcal{G}_{<t} = \{(u_i, v_i, t_i, \mathbf{e}_i) \in \mathcal{G} \mid t_i < t\}$. The prediction function is

$$\hat{y}_{uv}(t) = f_\theta(u, v, t, \mathcal{G}_{<t}), \tag{1}$$

where $\hat{y}_{uv}(t) \in [0, 1]$ is the predicted interaction probability. For both positive and negative candidates, S3Former constructs all historical sequences, subgraphs, and statistical features strictly from $\mathcal{G}_{<t}$, thereby excluding interactions at or after the prediction time.

3.2 Overview of S3Former

S3Former follows a candidate-centered design. For each candidate interaction (u, v, t), it extracts historical interaction sequences of the two endpoints, induces a causal shared pair subgraph around the candidate pair, and computes online node- and pair-level statistics from $\mathcal{G}_{<t}$. These components correspond to three complementary signals: sequential dynamics, explicit local topology, and statistical memory. Figure 1 summarizes how these signals are encoded, fused, and transformed into a link prediction score.

The sequential channel captures long-range temporal evolution and recent short-term dynamics. The structural channel encodes explicit local connectivity patterns by inducing a historical subgraph around the candidate pair. The statistical channel provides lightweight causal memory, including node activity, pairwise interaction strength, recency, temporal decay, and neighborhood overlap. Instead of simply concatenating these signals, S3Former first uses NodeCCF to compute a structural reliability gate from the endpoint sequence state, structural state, statistical state, and subgraph quality descriptors. PairCCF then computes a scalar retreat gate from the base pair representation and causal pair memory, allowing pair-level correction to intervene only when the pair context is reliable.

3.3 Temporal, Structural, and Statistical Encoding

Multi-timescale sequential encoding. For node u, S3Former collects its historical one-hop interaction sequence before time t, denoted by $\mathcal{S}_u^t$. Each historical interaction is encoded with neighbor features, edge features, and relative-time features. To model long histories efficiently, S3Former partitions the historical sequence into temporal patches and feeds the patch sequence into a Transformer encoder [16]. In parallel, the most recent W valid interactions are encoded by a causal temporal convolution to capture short-term local dynamics. The detailed token construction, time encoding, and patching procedure are provided in Appendix A.

The final sequential representation treats the long-term state as the backbone and the short-term state as an adaptive correction:

$$\begin{aligned} \mathbf{h}_u^{\mathrm{seq}} &= \mathbf{h}_u^{\mathrm{long}} + \eta_u\big(\mathbf{g}_u^{\mathrm{short}} \odot \mathbf{W}_s \mathbf{h}_u^{\mathrm{short}}\big), \\ \mathbf{g}_u^{\mathrm{short}} &= \sigma\big(\mathbf{W}_g(\mathbf{h}_u^{\mathrm{long}} \oplus \mathbf{W}_s \mathbf{h}_u^{\mathrm{short}}) + \mathbf{b}_g\big), \\ \eta_u &= \sigma\big(\mathrm{MLP}_\eta(\mathbf{h}_u^{\mathrm{long}} \oplus \mathbf{W}_s \mathbf{h}_u^{\mathrm{short}})\big). \end{aligned} \tag{2}$$

where $\mathbf{h}_u^{\mathrm{long}}$ and $\mathbf{h}_u^{\mathrm{short}}$ denote long- and short-term temporal representations, $\mathbf{W}_s$ is a learnable projection into the node-feature dimension, η_u is a sample-level coefficient, and $\mathbf{g}_u^{\mathrm{short}}$ is a dimension-wise gate parameterized by $\mathbf{W}_g$ and $\mathbf{b}_g$. The same procedure is applied to node v, yielding $\mathbf{h}_v^{\mathrm{seq}}$.

Causal shared pair subgraph encoding. To encode local topology, S3Former constructs a causal shared pair subgraph for each candidate (u, v, t). Unlike co-occurrence encodings that summarize whether historical neighbors overlap, the shared pair subgraph explicitly includes historical edges among selected pair-centered nodes before t. This enables a lightweight edge-aware GNN to model how historical anchors are connected, rather than treating them only as isolated sequence tokens.

Let $\mathcal{N}_u^{<t}$ and $\mathcal{N}_v^{<t}$ denote the historical one-hop neighborhoods of u and v before time t. We select recent endpoint anchors $\mathcal{R}_u^t$ and $\mathcal{R}_v^t$ according to causal recency-frequency scores, and select common-neighbor anchors $\mathcal{C}_{uv}^t$ from $\mathcal{N}_u^{<t} \cap \mathcal{N}_v^{<t}$. The selection scores and budgets are described in Appendix A. The induced node set and edge set are defined as

$$\mathcal{V}_{uv}^t = \{u, v\} \cup \mathcal{R}_u^t \cup \mathcal{R}_v^t \cup \mathcal{C}_{uv}^t, \qquad \mathcal{E}_{uv}^t = \{(a, b) \mid a, b \in \mathcal{V}_{uv}^t,\ \exists \tau < t,\ (a, b, \tau, \mathbf{e}_{ab}) \in \mathcal{G}_{<t}\}. \tag{3}$$

Unlike co-occurrence encoding, which only counts neighbor appearances, this construction explicitly includes historical edges among the selected nodes. Each node is assigned a role-aware feature indicating whether it is a source, destination, one-sided neighbor, or common neighbor. Each edge is assigned causal temporal attributes such as historical count, decayed strength, latest interaction gap, direction, and aggregated edge features.

We encode the shared pair subgraph with a lightweight message-passing module built on GraphSAGE-style aggregation [5], where neighbor messages are modulated by causal edge attributes. We then read the structural representations from the two center nodes:

$$\mathbf{h}_u^{\mathrm{str}}, \mathbf{h}_v^{\mathrm{str}} = \mathrm{Readout}\big(\mathrm{EdgeSAGE}(\mathcal{V}_{uv}^t, \mathcal{E}_{uv}^t)\big). \tag{4}$$

The purpose of this module is not to introduce a new message-passing operator. Rather, it provides a causal mechanism for constructing and encoding the induced historical topology around a candidate pair, which is not explicitly available in serialized endpoint histories or co-occurrence counts. We also compute structural quality descriptors $\mathbf{q}_u^{\mathrm{str}}$ and $\mathbf{q}_v^{\mathrm{str}}$, including subgraph size, edge density, valid anchor ratio, and common-neighbor coverage. These descriptors help the fusion module decide whether structural evidence is reliable.

Causal statistical memory. S3Former uses a compact causal statistical memory to summarize historical activity and relation strength. For each endpoint, we compute low-dimensional node descriptors such as historical activity, temporal recency, and neighborhood diversity. For each candidate pair, we compute pair descriptors that summarize prior interaction strength, temporal decay, and shared-neighborhood evidence. All descriptors are computed online from $\mathcal{G}_{<t}$ and updated only after the prediction at time t is made. Endpoint descriptors are encoded as $\mathbf{h}_u^{\mathrm{stat}}$ and $\mathbf{h}_v^{\mathrm{stat}}$ by a small MLP. The pair statistical representation is obtained by

$$\mathbf{h}_{uv}^{\mathrm{mem}} = \mathrm{MLP}_{\mathrm{stat}}(\boldsymbol{\psi}_{uv}(t)), \tag{5}$$

where $\boldsymbol{\psi}_{uv}(t)$ denotes the causal pair-statistical vector. Detailed definitions are provided in Appendix A.

3.4 Fusion and Prediction

The three channels provide complementary but not equally reliable evidence across different nodes and candidate pairs. The sequential channel captures temporal evolution, the structural channel recovers causal local topology, and the statistical channel summarizes explicit historical interaction patterns. Instead of using a fixed concatenation head, S3Former adopts a two-level context-aware collaborative fusion mechanism. NodeCCF calibrates endpoint-level representations, while PairCCF performs candidate-specific relation correction. All fusion contexts are computed from $\mathcal{G}_{<t}$.

Node-level context-aware fusion. Before fusion, the sequential, structural, and statistical node representations are projected into a common hidden dimension d. For node u, NodeCCF forms a causal node context

$$\mathbf{c}_u^{\text{node}} = \mathbf{h}_u^{\text{seq}} \oplus \mathbf{h}_u^{\text{str}} \oplus \mathbf{h}_u^{\text{stat}} \oplus \mathbf{q}_u^{\text{str}}, \tag{6}$$

where $\mathbf{q}_u^{\text{str}}$ denotes structural quality descriptors, including valid anchor ratio, subgraph size, edge density, and common-neighbor coverage. Based on this context, NodeCCF computes a dimension-wise structural reliability gate and a residual correction:

$$\boldsymbol{\alpha}_u = \sigma\big(\text{MLP}_\alpha(\mathbf{c}_u^{\text{node}})\big), \qquad \mathbf{r}_u = \text{MLP}_r(\mathbf{c}_u^{\text{node}}). \tag{7}$$

The structural representation is then calibrated as

$$\widetilde{\mathbf{h}}_u^{\text{str}} = \boldsymbol{\alpha}_u \odot \mathbf{h}_u^{\text{str}} + \mathbf{r}_u. \tag{8}$$

Finally, the endpoint representation is obtained by residual fusion:

$$\mathbf{h}_u = \text{LN}\Big(\mathbf{h}_u^{\text{seq}} + \text{MLP}_{\text{node}}(\mathbf{h}_u^{\text{seq}} \oplus \widetilde{\mathbf{h}}_u^{\text{str}} \oplus \mathbf{h}_u^{\text{stat}})\Big). \tag{9}$$

The same NodeCCF parameters are shared for the source and destination endpoints, yielding $\mathbf{h}_v$ analogously. Consequently, sparse or noisy structural context has a limited effect on the endpoint representation.

Pair-level context-aware correction. After endpoint-level fusion, we construct a standard pair representation:

$$\mathbf{h}_{uv}^{\text{base}} = \mathbf{h}_u \oplus \mathbf{h}_v \oplus (\mathbf{h}_u \odot \mathbf{h}_v) \oplus |\mathbf{h}_u - \mathbf{h}_v|. \tag{10}$$

Endpoint representations alone may not fully capture candidate-specific relation patterns, such as repeated interactions, strong recency effects, or shared-neighborhood evidence. PairCCF therefore builds a pair context

$$\mathbf{c}_{uv}^{\text{pair}} = \mathbf{h}_{uv}^{\text{base}} \oplus \mathbf{h}_{uv}^{\text{mem}} \oplus \mathbf{p}_{uv}(t), \tag{11}$$

where $\mathbf{h}_{uv}^{\text{mem}}$ is the encoded causal pair-statistical memory, and $\mathbf{p}_{uv}(t)$ contains pair-time and reliability descriptors computed from $\mathcal{G}_{<t}$. PairCCF predicts a correction vector and a scalar retreat gate:

$$\Delta\mathbf{h}_{uv}^{\text{pair}} = \text{MLP}_\Delta(\mathbf{c}_{uv}^{\text{pair}}), \qquad \gamma_{uv} = \sigma\big(\text{MLP}_\gamma(\mathbf{c}_{uv}^{\text{pair}})\big). \tag{12}$$

The final pair representation is

$$\mathbf{h}_{uv}^{\text{pair}} = \text{LN}\big(\mathbf{h}_{uv}^{\text{base}} + \gamma_{uv}\Delta\mathbf{h}_{uv}^{\text{pair}}\big). \tag{13}$$

This formulation makes PairCCF a retreatable correction mechanism rather than a rigid prediction head. When pair-level context is unreliable, a small γ_{uv} keeps the representation close to the endpoint-based predictor; when pair memory is informative, the correction can intervene more strongly.

Prediction objective. The final link probability is computed by

$$\ell_{uv}(t) = f_{\text{pred}}(\mathbf{h}_{uv}^{\text{pair}}), \qquad \hat{y}_{uv}(t) = \sigma(\ell_{uv}(t)). \tag{14}$$

During training, each observed interaction (u, v, t) is treated as a positive sample. We construct a negative candidate (u, v^-, t) by corrupting the destination endpoint while keeping u and t fixed, and reject sampled negatives that form observed positive interactions at the same timestamp. For both positive and negative candidates, all historical sequences, shared pair subgraphs, and statistical features are constructed strictly from $\mathcal{G}_{<t}$. The model is optimized with binary cross-entropy.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate S3Former on nine public CTDG benchmarks from the DyGLib and EdgeBank benchmark suites [11, 20]: Wikipedia, Reddit, MOOC, LastFM, Enron, Social Evo., UCI, Flights, and Contact. We follow the standard chronological 70%/15%/15% train/validation/test split and evaluate both transductive and inductive link prediction. Recurrence-heavy institutional datasets, including CanParl, USLegis, UNTrade, and UNVote, are analyzed separately in Appendix B. Dataset statistics and protocol details are provided in Appendix A.

Baselines. We compare S3Former with a wide range of state-of-the-art dynamic graph learning models, including JODIE, DyRep, TGAT, TGN, CAWN, EdgeBank, TCL, GraphMixer, NAT [10], DyGFormer, CNE-N, SALoM, and TPNet [9]. These baselines cover memory-based, attention-based, sequence-based, and structure-aware methods for dynamic graph link prediction. The TPNet entries in Table 1 are published five-run means under the matched conventional random-negative protocol; they are not local reproductions or TGB results.

Evaluation metrics. We report Average Precision (AP) and Area Under the ROC Curve (AUC) for both transductive and inductive tasks. For each dataset, we report Trans-AP, Trans-AUC, Ind-AP, and Ind-AUC, corresponding to AP and AUC under the transductive and inductive settings.

Implementation details. We use Adam [6] with a learning rate of 1×10^{-4} and binary cross-entropy loss. Model selection is performed using validation AP, and the best validation checkpoint is used for testing. Main experiments are repeated over five random seeds, and we report the mean and standard deviation. Dataset-specific batch sizes, sequence lengths, patch sizes, and structural sampling budgets are provided in Appendix A.

4.2 Main Results

We present the main results of S3Former on the nine main benchmark datasets. Table 1 reports AP and AUC-ROC results for both transductive and inductive link prediction under the same evaluation protocol.

Across the nine main benchmarks, S3Former achieves the best average rank in all four transductive and inductive AP/AUC settings, indicating consistent performance across datasets rather than gains concentrated on a small subset. The gains are visible on datasets where either local topology or repeated interaction patterns provide useful but non-trivial evidence. Additional recurrence diagnostics on CanParl, USLegis, UNTrade, and UNVote are provided in Appendix B; we analyze them separately as a boundary condition rather than mixing them with the main interaction-rich benchmarks.

4.3 Ablation Study

We perform ablation studies to evaluate the contribution of each component in S3Former. Specifically, we investigate three questions: (1) whether the sequential, structural, and statistical channels provide complementary information; (2) whether the two-level context-aware fusion mechanism is more effective than flat fusion; and (3) whether PairCCF benefits from its adaptive and retreatable correction design. Unless otherwise specified, we report transductive AP (%) on representative datasets.

4.3.1 Component Ablation

We first analyze the contribution of the three information channels. We evaluate three single-channel variants, `seq_only`, `struct_only`, and `stat_only`, and three two-channel variants, `seq+str`, `seq+stat`, and `str+stat`.

As shown in Table 2, the sequential-only backbone is insufficient on all representative datasets. For example, on Wikipedia, `seq_only` achieves 94.60% AP, while adding the structural channel improves the performance to 99.37%, and adding the statistical channel improves it to 99.25%. A similar trend is observed on Reddit, where `seq+str` improves AP from 93.98% to 99.64%. These results show that both explicit local topology and causal statistical memory provide information not captured by the sequential backbone alone.

Table 1: Main results on nine interaction-rich CTDG benchmarks. We report five-run means. Standard deviations for the baseline comparison excluding TPNet are provided in Appendix C; TPNet values are published protocol-matched five-run means [9].

Metrics	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	NAT	DyGFormer	CNE-N	SALoM	TPNet	S3Former
Trans-AP	Wikipedia	96.50	94.86	96.94	98.28	98.76	90.37	96.47	97.25	97.50	99.03	98.61	99.03	99.32	**99.43**
	Reddit	98.31	98.22	98.52	98.47	99.11	94.86	97.53	97.31	99.10	99.22	99.26	99.27	99.27	**99.68**
	MOOC	80.23	81.97	85.84	93.21	80.15	57.97	82.38	82.78	87.21	87.52	90.16	92.42	**96.39**	93.03
	LastFM	70.85	71.92	73.42	84.36	86.99	79.29	67.27	75.61	88.57	93.00	92.60	93.14	94.50	**96.60**
	Enron	84.77	82.38	71.12	91.51	89.56	83.53	79.70	82.25	90.81	92.47	92.13	94.08	92.90	**97.49**
	Social Evo.	89.89	88.87	93.16	89.83	84.96	74.95	93.13	93.37	91.23	94.73	94.50	94.73	94.73	**96.45**
	UCI	89.43	65.14	79.63	92.94	95.18	76.20	89.57	93.25	94.26	95.79	95.64	96.36	**97.35**	97.25
	Flights	95.60	95.29	94.03	97.94	98.51	89.35	91.23	90.99	97.66	98.91	98.73	98.94	98.93	**99.05**
	Contact	95.31	95.98	96.28	95.00	90.26	92.58	92.44	91.92	97.25	98.29	98.28	98.53	98.66	**98.82**
	Avg. rank ↓	10.44	11.33	9.78	7.67	8.67	12.56	11.44	10.33	7.22	4.17	5.00	2.89	2.17	**1.33**
Trans-AUC	Wikipedia	96.33	94.37	96.67	98.01	98.54	90.78	95.84	96.92	96.72	98.91	98.40	98.87	99.30	**99.45**
	Reddit	98.31	98.17	98.47	98.32	99.01	95.37	97.42	97.17	99.02	99.15	99.19	99.20	99.22	**99.62**
	MOOC	83.81	85.03	87.11	93.56	80.38	60.86	83.12	84.01	88.38	87.91	91.42	92.52	**97.17**	93.71
	LastFM	70.49	71.16	71.59	82.66	85.92	83.77	64.06	73.53	86.94	93.05	92.21	92.32	94.39	**96.70**
	Enron	87.96	84.89	68.89	90.99	90.45	87.05	75.74	84.38	92.02	93.33	92.77	95.11	93.98	**98.07**
	Social Evo.	92.05	90.76	94.76	90.36	87.34	81.60	94.84	95.23	93.22	96.30	96.20	96.36	96.43	**97.23**
	UCI	90.44	68.77	78.53	92.17	93.87	77.30	87.82	91.81	93.02	94.49	94.32	95.53	**96.79**	96.72
	Flights	96.21	95.95	94.13	97.99	98.45	90.23	91.21	91.13	97.32	98.93	98.74	98.99	99.00	**99.12**
	Contact	96.66	96.48	96.95	95.37	89.99	94.34	94.15	93.94	97.64	98.53	98.62	98.69	98.91	**98.94**
	Avg. rank ↓	10.11	11.11	9.89	8.00	8.78	12.44	11.67	10.44	7.00	4.33	4.89	3.22	1.89	**1.22**
Ind-AP	Wikipedia	94.82	92.43	96.22	97.49	98.24	–	96.22	96.65	95.40	98.59	97.76	98.49	98.91	**99.10**
	Reddit	96.50	96.09	97.09	97.26	98.62	–	94.09	95.26	98.56	98.84	98.82	98.93	98.86	**99.37**
	MOOC	79.63	81.07	85.50	91.86	81.42	–	80.60	81.41	83.59	86.96	88.71	90.53	**95.07**	93.27
	LastFM	81.61	83.02	78.63	87.18	89.42	–	73.53	82.11	86.87	94.23	94.00	94.56	95.36	**95.89**
	Enron	80.72	74.55	67.05	84.53	86.35	–	76.14	75.88	89.03	89.76	87.59	91.67	90.34	**95.24**
	Social Evo.	91.96	90.04	91.41	82.85	79.94	–	91.55	91.86	91.22	93.14	92.70	92.84	93.24	**94.73**
	UCI	79.86	57.48	79.54	82.04	92.73	–	87.36	91.19	87.30	94.54	93.58	94.36	95.74	**95.91**
	Flights	94.74	92.88	88.73	95.03	97.06	–	83.41	83.03	96.59	97.79	97.34	97.85	97.97	**98.10**
	Contact	94.34	92.18	95.87	88.56	89.55	–	91.11	90.59	96.79	98.03	97.58	97.79	**98.39**	98.05
	Avg. rank ↓	9.89	11.00	9.94	8.44	7.78	–	10.61	9.89	7.89	3.78	5.22	3.33	2.00	**1.22**
Ind-AUC	Wikipedia	94.33	91.49	95.90	97.08	98.03	–	95.57	96.30	94.74	98.48	97.45	98.26	98.90	**99.10**
	Reddit	96.52	96.05	96.98	96.94	98.42	–	93.80	94.97	97.99	98.71	98.69	98.85	98.73	**99.35**
	MOOC	83.16	84.03	86.84	92.02	81.86	–	81.43	82.77	86.13	87.62	89.94	90.09	**95.55**	93.97
	LastFM	81.13	82.24	76.99	85.58	87.82	–	70.84	80.37	83.07	94.08	93.62	93.77	95.36	**96.17**
	Enron	81.96	76.34	64.63	83.58	87.02	–	72.33	76.51	89.92	90.69	88.24	92.57	90.21	**96.45**
	Social Evo.	93.70	91.18	93.41	82.04	84.73	–	93.71	94.09	92.11	95.29	94.99	95.03	95.47	**97.12**
	UCI	78.80	58.08	77.64	86.48	90.40	–	84.49	89.30	83.81	92.63	91.31	92.17	94.40	**94.68**
	Flights	95.21	93.56	88.64	95.92	96.86	–	82.48	82.27	96.36	97.80	97.20	97.95	98.05	**98.17**
	Contact	95.37	91.89	96.53	88.87	89.07	–	93.05	92.83	96.67	98.30	97.91	97.98	**98.51**	98.37
	Avg. rank ↓	9.67	10.89	9.78	8.44	8.00	–	10.89	9.78	8.00	3.56	5.22	3.44	2.11	**1.22**

Note. Best results are in bold and second-best results are underlined. Average ranks are computed across the nine datasets; lower is better.

Table 2: Component ablation for S3Former on representative datasets. We report transductive AP (%).

Dataset	`seq_only`	`struct_only`	`stat_only`	`seq+str`	`seq+stat`	`str+stat`	Full
Wikipedia	94.60	99.33	99.20	99.37	99.25	99.38	99.43
MOOC	78.66	88.91	83.90	90.64	87.30	91.50	93.03
Reddit	93.98	99.21	98.60	99.64	99.57	99.55	99.68
LastFM	65.82	95.25	92.60	96.30	94.10	96.10	96.60

The comparison between `seq+str` and `seq+stat` further indicates that the relative importance of structural and statistical information is dataset-dependent. On Wikipedia and Reddit, the structural channel is particularly effective, supporting the narrower motivation that serialization can weaken explicit local connectivity among candidate-centered historical context nodes. On datasets with repeated user-item or pairwise interaction patterns, statistical memory provides complementary evidence by capturing relation strength, recency, and long-term stability.

Figure 2 shows that direct concatenation and removing both CCF modules underperform the full model; detailed discussion is provided in Appendix F.

Removing either the adaptive coefficient or residual correction degrades performance, supporting the retreatable PairCCF design; detailed results are provided in Appendix G.

4.4 Structural Mechanism Analysis

We use an induced-topology diagnostic to quantify historical edges among selected pair-centered nodes that are not explicitly exposed by endpoint-history serialization. For a candidate pair (u, v, t), let $\mathcal{E}_{uv}^{t}$ be the causal induced edge set among the selected pair-centered nodes, and let $\mathcal{E}_{uv}^{t,\mathrm{star}}$ denote

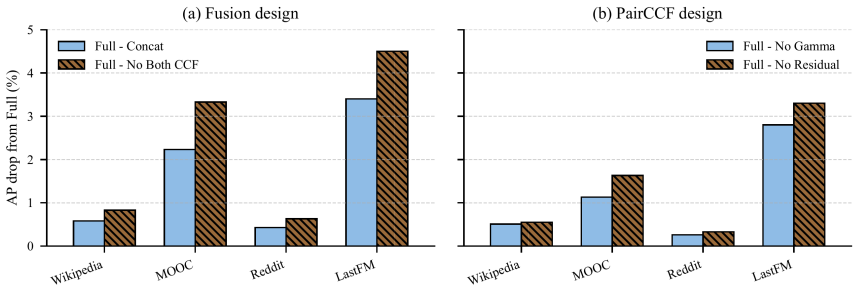


Figure 2: CCF ablation measured by AP drop from the full S3Former. Larger drops indicate stronger contribution.

(a) Structural isolation

Variant	Wikipedia	MOOC	LastFM	UCI
No topology	99.25	85.72	94.10	96.36
+ CN/Jaccard	99.28	87.60	94.35	96.50
+ Binary induced edges	99.35	90.80	94.95	96.82
+ Edge-aware EdgeSAGE	99.43	93.03	95.50	97.05
S3Former full	99.43	93.03	96.60	97.25

(b) Topology-bucket gain

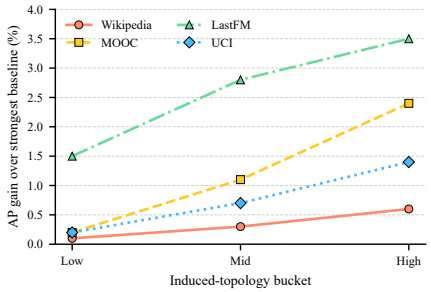


Table 3: Structural mechanism diagnostics. Left: structural isolation under the same sequence-statistical backbone. Right: topology-bucketed AP gain over the strongest baseline.

the endpoint-star edges directly involving u or v in the serialized histories. We define

$$\Delta_{\text{ind}}(u, v, t) = \frac{|\mathcal{E}_{uv}^{t} \setminus \mathcal{E}_{uv}^{t,\text{star}}|}{\max(1, |\mathcal{E}_{uv}^{t}|)}.$$

This diagnostic does not imply that sequence, walk, or co-occurrence models cannot learn structural correlations; it only measures the amount of induced-edge information absent from serialized endpoint histories.

Table 3 isolates the structural channel under the same sequence-statistical backbone and selected node set. CN/Jaccard features provide only mild gains, while adding binary induced edges and edge-aware EdgeSAGE yields larger improvements, indicating that explicit historical edges contain information beyond overlap statistics. The gains also increase from low- to high-topology buckets, supporting the intended role of explicit induced-edge modeling when candidate-centered histories contain richer local topology.

4.5 Robustness under Historical Negative Sampling

The random negative sampler used in the main DyGLib-style benchmark [11, 20] may reward historical pair recurrence, especially for models with explicit pair-level memory. To examine whether S3Former relies only on this shortcut, we additionally evaluate representative datasets under historical negative sampling. In this protocol, negative candidates are sampled from historically observed pairs before the prediction time, making direct pair-frequency signals less discriminative than under random negative sampling [1, 11]. We evaluate DyGFormer, SALoM, and S3Former using the same chronological split, historical negative sampler, and five random seeds.

Table 4 shows that historical negatives substantially reduce the absolute performance compared with the random-negative results in Table 1. Nevertheless, S3Former remains competitive and achieves the best AP on all four datasets, although the margins are much smaller on Wikipedia and LastFM. The reduced margins indicate that performance under random negative sampling should be interpreted with care. Appendix E provides an extended eight-dataset, four-metric evaluation together with a targeted MOOC ablation.

Table 4: Robustness under historical negative sampling, reported as transductive AP (%, mean ± std).

Dataset	DyGFormer	SALoM	S3Former
Wikipedia	82.23 ± 2.54	83.25 ± 0.22	$\mathbf{85.73 \pm 3.21}$
UCI	82.17 ± 0.82	66.27 ± 1.37	$\mathbf{83.15 \pm 1.09}$
Enron	75.63 ± 0.73	62.07 ± 1.05	$\mathbf{78.32 \pm 0.56}$
LastFM	81.57 ± 0.48	69.59 ± 0.24	$\mathbf{81.77 \pm 0.55}$

Table 5: Ranking-based evaluation on TGB-v2 datasets. (a) Test MRR comparison with representative baselines from the official TGB leaderboard. (b) Detailed single-run ranking metrics of S3Former.

(a) Official TGB leaderboard comparison (Test MRR)

Method	Wiki-v2	Review-v2	Coin-v2	Flight-v2
TGN	0.396 ± 0.060	0.349 ± 0.020	0.586 ± 0.037	0.705 ± 0.020
DyRep	0.050 ± 0.017	0.220 ± 0.030	0.452 ± 0.046	0.556 ± 0.014
GraphMixer	0.118 ± 0.002	$\mathbf{0.521 \pm 0.015}$	–	–
DyGFormer	0.798 ± 0.004	0.224 ± 0.015	0.752 ± 0.004	–
TNCN	0.718 ± 0.001	0.377 ± 0.010	0.762 ± 0.004	0.820 ± 0.004
TPNet	$\mathbf{0.827 \pm 0.001}$	–	$\mathbf{0.832 \pm 0.001}$	0.884 ± 0.001
S3Former	0.7489	0.4089	0.7988	**0.8923**

(b) Detailed single-run S3Former ranking metrics

Dataset	Val MRR	Test MRR	H@1	H@3	H@10
`tgbl-wiki-v2`	0.7654	0.7489	0.6570	0.7907	0.8399
`tgbl-review-v2`	0.3547	0.4089	0.1691	0.3426	0.6295
`tgbl-coin-v2`	0.7849	0.7988	0.6886	0.8397	0.9415
`tgbl-flight-v2`	0.9030	0.8923	0.8408	0.9321	0.9802

Source: Baseline MRR values in panel (a) are taken from the official TGB leaderboard under the corresponding dataset versions (`https://tgb.complexdatalab.com/docs/leader_linkprop/`, accessed November 4, 2026); S3Former results are our single-run measurements.

4.6 Ranking-based Evaluation and Scalability

To complement the saturated random-negative AP/AUC evaluation, we further evaluate S3Former on four TGB-v2 link-prediction benchmarks using the official candidate sets and evaluator [**?**]. Table 5(a) provides broader context by reporting representative baseline results from the official TGB leaderboard under the corresponding dataset versions, whereas Table 5(b) reports our detailed single-run MRR and Hits@K measurements. S3Former is competitive across these datasets, including the highest displayed Test MRR on `tgbl-flight-v2`, but it does not consistently outperform all leaderboard baselines. We therefore make no claim of an official leaderboard rank or universal state of the art.

Scalability was measured on one RTX 4090 with one-time index/cache construction excluded; full wall-clock results and matched local TPNet timing runs are provided in Appendix I. These matched local runs are distinct from the official leaderboard accuracy values in Table 5(a). Wiki has similar per-epoch cost, while S3Former is faster on Review and slower on Coin; therefore, no uniform efficiency superiority is claimed.

5 Conclusion

In this paper, we introduced S3Former, a unified framework for continuous-time dynamic link prediction that integrates long-term sequential dynamics, shared local topology, and causal statistical memory. By proposing a novel causal shared pair-subgraph mechanism, S3Former addresses the loss of explicit local connectivity that can arise when historical interactions are serialized. Moreover, our two-level hierarchical fusion (NodeCCF and PairCCF) effectively combines endpoint state modeling with candidate-specific relation correction, while preserving strict temporal causality.

Across nine representative dynamic graph benchmarks, S3Former achieves the best average rank across all four standard transductive and inductive AP/AUC settings. Ablation studies and mechanism analyses support the complementary roles of the structural and statistical channels relative to the sequence-first backbone.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant No. 62401123. We thank Dr. Qiang Dong for helpful discussions on the theoretical analysis.

Competing Interests. The authors declare no competing interests.

References

[1] Ziyue Chen, Tongya Zheng, and Mingli Song. Curriculum negative mining for temporal networks. *Neural Networks*, 191:107858, 2025.

[2] Ke Cheng, Peng Linzhi, Junchen Ye, Leilei Sun, and Bowen Du. Co-neighbor encoding schema: A light-cost structure encoding method for dynamic link prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 421–432, 2024.

[3] Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? 2023.

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[5] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. volume 30, 2017.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.

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

[8] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, 2003.

[9] Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. volume 37, pages 141153–141182, 2024.

[10] Yuhong Luo and Pan Li. Neighborhood-aware scalable temporal network representation learning. In *Learning on Graphs Conference*, pages 1–1. PMLR, 2022.

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

[12] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in neural information processing systems*, 35:14501–14515, 2022.

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

[14] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th international conference on web search and data mining*, pages 519–527, 2020.

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

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

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

[18] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. 2021.

[19] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. 2020.

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

[21] Tao Zou, Yuhao Mao, Junchen Ye, and Bowen Du. Repeat-aware neighbor sampling for dynamic graph learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4722–4733, 2024.

A Dataset and Experimental Details

A.1 Dataset Statistics

We evaluate S3Former on nine interaction-rich continuous-time dynamic graph benchmarks in the main paper: Wikipedia, Reddit, MOOC, LastFM, Enron, Social Evo., UCI, Flights, and Contact. We additionally analyze CanParl, USLegis, UNTrade, and UNVote in Appendix B as recurrence-heavy institutional datasets. These datasets exhibit strong historical pair-recurrence effects under random negative sampling, and are therefore analyzed separately as diagnostic boundary cases rather than included in the aggregate main comparison.

All datasets are adopted from public dynamic link prediction benchmarks used in DyGLib and EdgeBank [11, 20]. Each dataset is represented as a chronological interaction stream

$$\mathcal{G} = \{(u_i, v_i, t_i, e_i)\}_{i=1}^{N},$$

where u_i and v_i denote the endpoints, t_i denotes the timestamp, and e_i denotes the optional edge feature. Table 6 summarizes the dataset statistics. "N/A" indicates that the dataset does not contain the corresponding raw node or edge features.

Table 6: Statistics of the dynamic graph datasets used in the main experiments and appendix analysis.

Dataset	#Nodes	#Interactions	#Time Steps	Node Feat.	Edge Feat.	Domain	Usage
Wikipedia	9,227	157,474	152,757	N/A	172	User–page interaction	Main
Reddit	10,984	672,447	669,065	N/A	172	User–post interaction	Main
MOOC	7,144	411,749	345,600	N/A	4	Online learning	Main
LastFM	1,980	1,293,103	1,283,614	N/A	N/A	User–song interaction	Main
Enron	184	125,235	22,632	N/A	N/A	Email communication	Main
Social Evo.	74	2,099,519	565,932	N/A	2	Physical proximity	Main
UCI	1,899	59,835	58,911	N/A	N/A	Online communication	Main
Flights	13,169	1,927,145	122	N/A	1	Air traffic network	Main
CanParl	734	74,478	14	N/A	1	Parliamentary voting	Diagnostic
Contact	692	2,426,279	8,064	N/A	1	Physical proximity	Main
USLegis	225	60,396	12	N/A	1	Legislative co-sponsorship	Diagnostic
UNTrade	255	507,497	32	N/A	1	International trade	Diagnostic
UNVote	201	1,035,742	72	N/A	1	UN voting relation	Diagnostic

The datasets cover diverse domains, including user-item interaction, online communication, email networks, physical proximity, transportation, parliamentary voting, legislative co-sponsorship, international trade, and UN voting relations. Wikipedia, Reddit, MOOC, and LastFM are bipartite interaction datasets. Enron, UCI, Social Evo., and Contact describe communication or proximity networks. Flights represents temporal air traffic. CanParl, USLegis, UNTrade, and UNVote describe political or institutional relations with longer temporal spans.

A.2 Evaluation Protocol

We follow the standard chronological evaluation protocol used in DyGLib [20]. For each dataset, interactions are sorted by timestamp and split into training, validation, and testing sets with a ratio of 70%/15%/15%. Model selection is performed on the validation split, and the checkpoint with the best validation AP is used for test evaluation.

We evaluate dynamic link prediction under both transductive and inductive settings. In the transductive setting, the model predicts future links among nodes observed during training. In the inductive setting, test interactions involve at least one endpoint unseen during training, requiring the model to generalize to newly observed nodes or sparse-history cases.

The main results use random negative sampling following the standard DyGLib protocol [20]. For each positive interaction (u, v, t), we construct a negative candidate (u, v^-, t) by corrupting the destination endpoint while keeping the source node and timestamp fixed. The sampled negative endpoint is rejected if it forms an observed positive interaction with u at the same timestamp. We report Average Precision (AP) and AUC-ROC for both transductive and inductive link prediction.

For every positive and negative candidate, S3Former constructs historical sequences, shared pair subgraphs, and statistical features strictly from $\mathcal{G}_{<t}$. Interactions at the current timestamp are not

Table 7: Main hyperparameters of S3Former. Dataset-specific configurations are selected on the validation split.

Dataset	Runs	Batch Size	Seq. Length	Patch Size	Recent K	Common K	Max GNN Neigh.
Wikipedia	5	200	32	1	16	8	20
Reddit	5	200	64	2	16	8	20
MOOC	5	200	256	8	16	8	20
LastFM	5	200	512	16	16	8	20
Enron	5	64	256	8	24	12	30
Social Evo.	5	200	32	1	16	8	20
UCI	5	200	32	1	16	8	20
Flights	5	200	256	8	16	8	20
CanParl	5	200	2048	64	16	8	20
Contact	5	200	32	1	16	8	20

used when making predictions at time t, which prevents current-event leakage. Learned-model robustness analysis with historical negatives is provided in Appendix E; the pair-memory diagnostic in Appendix B answers a separate question about recurrence-heavy datasets.

A.3 Implementation Details

S3Former is implemented based on the DyGLib training pipeline [20]. Unless otherwise specified, we train the model using Adam [6] with a learning rate of 1×10^{-4}, binary cross-entropy loss, and early stopping based on validation AP. Each main experiment is repeated with five random seeds, and we report the mean and standard deviation. Experiments are conducted on a single NVIDIA GeForce RTX 4090 D GPU with 24GB memory.

Table 7 summarizes the main dataset-specific hyperparameters. Following the sequence-length configurations commonly used in DyGLib-style evaluations, datasets with short interaction histories use shorter input sequences, while long-history datasets such as LastFM and CanParl use larger sequence lengths and patch sizes. For most datasets, we use a batch size of 200, recent anchor budget $K_r = 16$, common-neighbor budget $K_c = 8$, and maximum GNN neighbor budget 20. Enron uses a smaller batch size and larger structural budgets selected on the validation split due to its denser communication patterns.

For baseline methods, we follow the configurations reported in DyGLib and SALoM whenever available. When official results under the same protocol are available, we report the published numbers; otherwise, we reproduce the baseline using the released implementation and the same chronological split as S3Former.

A.4 Additional Method Details

This subsection gives the implementation-level details needed to reproduce the three channels and the two-stage fusion module. We keep only equations that specify model behavior; descriptor lists and engineering choices are summarized in tables. Unless otherwise stated, every quantity is computed from the causal history $\mathcal{G}_{<t}$ and is updated only after all candidates at timestamp t have been scored.

A.4.1 Sequential Encoding Details

For an endpoint u and prediction time t, we retrieve its historical one-hop sequence $\mathcal{S}_u^t$ from $\mathcal{G}_{<t}$. Each valid interaction token combines the historical neighbor representation, optional edge feature, and a relative-time encoding. The relative-time encoder is shared across endpoints and uses learnable sinusoidal bases initialized at multiple temporal scales.

S3Former uses a two-scale temporal encoder. The full valid history is partitioned into fixed-size temporal patches and processed by a Transformer encoder to obtain the long-term representation. In parallel, the most recent W interactions are processed by a causal temporal convolution to capture short-range dynamics. The long-term representation is used as the backbone, while the short-term representation is injected through the adaptive correction defined in Eq. (2) of the main paper. Source and destination endpoints share the same sequential encoder.

Table 8: Causal statistical descriptors used by S3Former. All quantities are computed from $\mathcal{G}_{<t}$ and updated online after timestamp-level scoring.

Group	Descriptors	Role in S3Former
Endpoint activity	count, degree, latest gap	node activeness and neighborhood diversity
Endpoint decay	time-decayed activity	recency-aware endpoint state
Pair frequency	history indicator, historical count	repeated relation evidence
Pair recency	latest pair gap, multi-scale decay	recency-aware pair strength
Overlap	common neighbors, Jaccard overlap	shared-neighborhood evidence
Stability	long-term count, burstiness ratio	stable versus bursty relation behavior
Pair-time context	endpoint gaps, shared-activation gap	candidate-specific temporal context

A.4.2 Shared Pair-Subgraph Construction and Encoding Details

For a candidate (u, v, t), the structural channel constructs a shared pair-centered subgraph rather than two independent endpoint subgraphs. This design exposes edges among selected historical anchors around the candidate pair. We first collect historical neighborhoods $\mathcal{N}_u^{<t}$ and $\mathcal{N}_v^{<t}$. A historical neighbor $c \in \mathcal{N}_u^{<t}$ is ranked by

$$s_u^t(c) = a_r \exp\big(-\lambda_r \Delta_{u,c}^{<t}\big) + a_f \log\big(1 + \mathrm{cnt}_{u,c}^{<t}\big)\,, \tag{15}$$

where $\Delta_{u,c}^{<t}$ is the latest-interaction gap between u and c, and $\mathrm{cnt}_{u,c}^{<t}$ is their historical interaction count. Recent endpoint anchors are selected by $s_u^t(\cdot)$ and $s_v^t(\cdot)$. Common-neighbor anchors are selected from $\mathcal{N}_u^{<t} \cap \mathcal{N}_v^{<t}$ by the summed score $s_u^t(c) + s_v^t(c)$. Ties are broken deterministically by latest timestamp and node index. If no common neighbor exists, the construction falls back to recent endpoint anchors.

Given the selected node set $\mathcal{V}_{uv}^t$, the structural channel induces all historical edges among these nodes before t. Each induced edge (a, b) is assigned causal edge attributes

$$\boldsymbol{\chi}_{ab}^t = \big[\mathrm{cnt}_{ab}^{<t}, \mathrm{decay}_{ab}^{<t}, \Delta_{ab}^{<t}, \mathrm{dir}_{ab}^{<t}, \bar{\mathbf{e}}_{ab}^{<t}\big]\,, \tag{16}$$

which encode historical count, time-decayed strength, latest gap, direction, and aggregated edge features. These attributes are used as edge gates in a lightweight GraphSAGE update:

$$\mathbf{h}_a^{(\ell+1)} = \sigma\Big(\mathbf{W}_s^{(\ell)}\mathbf{h}_a^{(\ell)} + \mathbf{W}_n^{(\ell)}\,\mathrm{Mean}_{b\in\mathcal{N}_{uv}^t(a)}\Big[\rho_\chi^{(\ell)}(\boldsymbol{\chi}_{ab}^t) \odot \mathbf{W}_m^{(\ell)}\mathbf{h}_b^{(\ell)}\Big]\Big)\,, \tag{17}$$

where $\rho_\chi(\cdot)$ is a sigmoid edge gate. The structural representations $\mathbf{h}_u^{\mathrm{str}}$ and $\mathbf{h}_v^{\mathrm{str}}$ are read from the two center nodes. We also compute structural quality descriptors, including valid anchor ratio, subgraph size, edge density, and common-neighbor coverage. These descriptors are not used as labels; they only condition the fusion gates.

A.4.3 Causal Statistical Memory Details

The statistical channel provides explicit, non-parametric historical descriptors before they are projected by MLPs. It contains endpoint-level descriptors $\boldsymbol{\phi}_u(t)$ and pair-level descriptors $\boldsymbol{\psi}_{uv}(t)$. Table 8 summarizes the descriptor groups.

For a pair (u, v), the multi-scale decayed strength is maintained as

$$D_{uv,k}^{<t} = \sum_{\tau_i<t:(u_i,v_i)=(u,v)} \exp(-\lambda_k(t-\tau_i))\,, \tag{18}$$

with several decay rates λ_k. Endpoint decay uses the same accumulator over interactions incident to the endpoint. The complete pair-statistical vector is formed by concatenating frequency, recency, decay, stability, overlap, and pair-time descriptors. Endpoint descriptors are encoded as $\mathbf{h}_u^{\mathrm{stat}}$ and $\mathbf{h}_v^{\mathrm{stat}}$; pair descriptors are encoded as $\mathbf{h}_{uv}^{\mathrm{mem}}$. No node-specific or pair-specific trainable table is introduced, which keeps the statistical channel applicable to inductive nodes after their causal histories become available.

Table 9: Implementation details of NodeCCF and PairCCF.

Module	Quantity	Input dimension	Output dimension
NodeCCF	$\mathbf{c}_u^{\mathrm{node}}$	$3d + d_q$	–
NodeCCF	$\boldsymbol{\alpha}_u^{\mathrm{node}}$	$3d + d_q$	d
NodeCCF	$\mathbf{r}_u^{\mathrm{node}}$	$3d + d_q$	d
NodeCCF	$\mathrm{MLP}_{\mathrm{node}}$	$3d$	d
PairCCF	$\mathbf{h}_{uv}^{\mathrm{base}}$	–	$4d$
PairCCF	$\mathbf{c}_{uv}^{\mathrm{pair}}$	$4d + d_m + d_p$	–
PairCCF	$\Delta\mathbf{h}_{uv}^{\mathrm{pair}}$	$4d + d_m + d_p$	$4d$
PairCCF	γ_{uv}	$4d + d_m + d_p$	1

A.4.4 Online Update Protocol

To avoid current-timestamp leakage, S3Former processes events in timestamp groups. For each timestamp t, it first constructs positive and negative candidates, queries sequence buffers, structural indices, and statistical memories built from $\mathcal{G}_{<t}$, and computes all logits/losses. Only after this scoring step are the positive events at timestamp t inserted into the online buffers and memories. This protocol is used for training, validation, and testing.

A.4.5 Context-aware Collaborative Fusion Details

The main equations of NodeCCF and PairCCF are provided in Section 3.4. Table 9 records the dimensions needed for reproduction. Before fusion, sequential, structural, and statistical node representations are projected to the same hidden dimension d. Here d_q is the dimension of structural quality descriptors, d_m is the dimension of encoded pair memory $\mathbf{h}_{uv}^{\mathrm{mem}}$, and d_p is the dimension of the pair-time and reliability descriptor $\mathbf{p}_{uv}(t)$.

NodeCCF parameters are shared between source and destination endpoints. PairCCF parameters are shared across candidate pairs and timestamps. All linear layers in the fusion modules use Xavier initialization [4] with zero biases. The correction branch, statistical encoders, structural encoder, and sequential encoder are trained jointly by the link prediction objective.

A.4.6 Training Objective

For each observed interaction (u, v, t), we use (u, v, t) as a positive candidate and sample one destination-corrupted negative candidate (u, v^-, t) unless otherwise specified. The negative endpoint is rejected if it forms an observed positive interaction with u at the same timestamp. Both positive and negative candidates query the same causal history $\mathcal{G}_{<t}$. The model is trained with binary cross-entropy, and model selection uses validation AP.

B Pair-Memory Diagnostic on Stable Institutional-Relation Datasets

In the main paper, we report results on nine interaction-rich dynamic graph benchmarks. We analyze CanParl together with USLegis, UNTrade, and UNVote in this appendix. These datasets describe long-span institutional relations, such as parliamentary voting, legislative co-sponsorship, international trade, and UN voting agreement. In such settings, repeated pairwise relations can be a dominant signal, and random transductive evaluation may reward simple historical pair memory.

B.1 Motivation

The purpose of this analysis is to examine whether long-span institutional-relation datasets contain a strong historical pair-memory shortcut under random negative sampling. A model may obtain high performance in such settings not necessarily because it learns more general temporal or structural patterns, but because positive test pairs are more likely to have appeared repeatedly in the past than randomly sampled negative pairs.

This issue is particularly relevant to S3Former because our method explicitly includes causal pair-level statistical memory. We therefore diagnose the extent to which a non-parametric pair-memory

Table 10: Pair-memory shortcut diagnostic under random negative sampling. Non-parametric baselines use only causal historical pair counts or decayed pair strength.

Dataset	Pair mode	Freq. AP	Freq. AUC	Decay AP	Decay AUC
CanParl	Undirected	78.80	64.06	84.32	66.20
USLegis	Undirected	67.35	64.51	91.75	79.53
UNTrade	Directed	92.35	91.30	96.29	95.55
UNVote	Undirected	82.73	84.49	98.30	99.15

heuristic can already explain performance on these datasets. This analysis does not imply that learned models exploit the shortcut in the same way. Rather, it characterizes a property of the evaluation protocol and the dataset mechanism.

B.2 Non-Parametric Pair-Frequency Baseline

We consider a simple frequency-based baseline that does not train any parameters. For a candidate interaction (u, v, t), the score is defined as

$$s_{\text{freq}}(u, v, t) = \log(1 + c_{uv}^{<t}),$$

where $c_{uv}^{<t}$ denotes the number of times the pair (u, v) has appeared before time t. This baseline does not use node features, edge features, GNN message passing, Transformer encoders, or learnable parameters. It only uses causal historical pair frequency.

We also use a decayed pair-memory score based on causal historical interaction strength. For CanParl, USLegis, and UNVote, we count pair memory over unordered node pairs because the relations are treated as undirected. For UNTrade, we count pair memory over ordered node pairs because the trade relation is directed.

B.3 Evaluation Protocol

We scan the complete event stream in chronological order. During validation and test evaluation, the pair history is not reset at the beginning of each split; instead, it is accumulated from the start of the full event stream. This matches the standard continuous-time setting where historical interactions before the prediction time are available.

To avoid current-timestamp leakage, all candidates at the same timestamp are scored before any positive edge at that timestamp is inserted into the history. After scoring is finished for timestamp t, the positive interactions at t are used to update the pair-frequency memory. Thus, the score for (u, v, t) only depends on interactions with timestamp strictly smaller than t.

For each positive interaction (u, v, t), we construct a random negative candidate (u, v^-, t) by corrupting the destination endpoint while keeping the source node and timestamp fixed. The negative endpoint is rejected if it forms a positive interaction with u at the same timestamp. The filtering set is constructed using all positive edges at the current timestamp in the full dataset.

B.4 Results

Table 10 reports the performance of the non-parametric pair-memory baselines under random negative sampling on the test split.

We further report the historical coverage gap:

$$\Delta_{\text{hist}} = \mathbb{P}(c_{uv}^{<t} > 0 \mid y = 1) - \mathbb{P}(c_{uv}^{<t} > 0 \mid y = 0).$$

A positive and large value indicates that positive samples are substantially more likely than random negative samples to have appeared in the historical event stream. The coverage gaps are $+23.95\%$ on CanParl, $+28.79\%$ on USLegis, $+46.95\%$ on UNTrade, and $+33.12\%$ on UNVote.

B.5 Discussion

The results indicate that random transductive evaluation on these long-span institutional datasets is highly sensitive to historical pair repetition. On UNTrade and UNVote, the decayed pair-memory

baseline already achieves very high AP and AUC, suggesting that recent repeated relations explain much of the test signal. CanParl shows a milder but still visible recurrence effect.

This diagnostic should not be interpreted as evidence that a particular neural model relies only on pair frequency. Instead, it shows that the evaluation protocol itself rewards historical pair memory in these datasets. Therefore, high random-transductive performance on USLegis, UNTrade, and UNVote should be interpreted carefully: it does not automatically imply stronger temporal generalization or better structural reasoning.

This observation motivates our decision to report these datasets separately from the nine main benchmarks. The main paper focuses on interaction-rich datasets where temporal evolution, local structure, and online statistics are all relevant. The stable institutional-relation datasets are used here to analyze a boundary condition of dynamic link prediction evaluation, where long-term pair recurrence can dominate other signals.

Appendix analysis on stable institutional-relation datasets further highlights a boundary condition where long-term pair recurrence can dominate local structural signals. Future work may explore how to better calibrate structural recovery and pair-memory priors under such stable-relation regimes while preserving computational efficiency.

C Full Results with Standard Deviations

This appendix reports standard deviations for the baseline comparison excluding TPNet. TPNet is reported in Table 1 using published five-run means. Accordingly, the average ranks below are computed over the baseline set excluding TPNet and therefore differ from those in Table 1.

Table 11: Full results with standard deviations on the nine main benchmark datasets. Results are reported as mean ± standard deviation over five runs, following the same layout as Table 1.

Metrics	Datasets	JODIE	DyRep	TGAT	TGN	CAWN	EdgeBank	TCL	GraphMixer	NAT	DyGFormer	CNE-N	SALoM	S3Former
Trans-AP	Wikipedia	96.50 ± 0.14	94.86 ± 0.06	96.94 ± 0.06	98.28 ± 0.06	98.76 ± 0.03	90.37 ± 0.00	96.47 ± 0.16	97.25 ± 0.03	97.50 ± 0.04	99.03 ± 0.02	98.61 ± 0.04	99.03 ± 0.02	99.43 ± 0.01
	Reddit	98.31 ± 0.14	98.22 ± 0.04	98.52 ± 0.02	98.47 ± 0.06	99.11 ± 0.01	94.86 ± 0.00	97.53 ± 0.02	97.31 ± 0.01	99.10 ± 0.21	99.22 ± 0.01	99.26 ± 0.01	99.27 ± 0.03	99.68 ± 0.02
	MOOC	80.23 ± 2.44	81.97 ± 0.49	85.84 ± 0.15	93.21 ± 1.51	80.15 ± 0.25	57.97 ± 0.00	82.38 ± 0.24	82.78 ± 0.15	87.21 ± 0.63	87.52 ± 0.49	90.16 ± 0.07	92.42 ± 0.96	93.03 ± 0.37
	LastFM	70.85 ± 2.13	71.92 ± 2.21	73.42 ± 0.21	84.36 ± 3.97	86.99 ± 0.06	79.29 ± 0.00	67.27 ± 2.16	75.61 ± 0.24	88.57 ± 1.76	93.00 ± 0.12	92.60 ± 0.03	93.14 ± 0.35	96.60 ± 0.21
	Enron	84.77 ± 0.30	82.38 ± 3.36	71.12 ± 0.97	91.51 ± 1.11	89.56 ± 0.09	83.53 ± 0.00	79.70 ± 0.71	82.25 ± 0.16	90.81 ± 0.31	92.47 ± 0.12	92.13 ± 0.06	94.08 ± 0.40	97.49 ± 0.04
	Social Evo.	89.89 ± 0.55	88.87 ± 0.30	93.16 ± 0.17	89.83 ± 0.17	84.96 ± 0.09	74.95 ± 0.00	93.13 ± 0.16	93.37 ± 0.07	91.23 ± 0.37	94.73 ± 0.01	94.50 ± 0.04	94.73 ± 0.07	96.45 ± 0.02
	UCI	89.43 ± 1.09	65.14 ± 2.30	79.63 ± 0.70	92.94 ± 1.04	95.18 ± 0.06	76.20 ± 0.00	89.57 ± 1.63	93.25 ± 0.57	94.26 ± 0.37	95.79 ± 0.17	95.64 ± 0.11	96.36 ± 0.05	97.25 ± 0.15
	Flights	95.60 ± 1.73	95.29 ± 0.72	94.03 ± 0.18	97.94 ± 0.14	98.51 ± 0.01	89.35 ± 0.00	91.23 ± 0.02	90.99 ± 0.05	97.66 ± 0.80	98.91 ± 0.01	98.73 ± 0.01	98.94 ± 0.06	99.05 ± 0.02
	Contact	95.31 ± 1.33	95.98 ± 0.15	96.28 ± 0.09	95.00 ± 0.56	90.26 ± 0.28	92.58 ± 0.00	92.44 ± 0.12	91.92 ± 0.03	97.25 ± 0.33	98.29 ± 0.01	98.28 ± 0.01	98.53 ± 0.41	98.82 ± 0.14
	Avg. rank ↓	9.4	10.3	8.8	6.7	7.7	11.6	10.4	9.3	6.2	3.2	4.0	2.2	**1.1**
Trans-AUC	Wikipedia	96.33 ± 0.07	94.37 ± 0.09	96.67 ± 0.07	98.01 ± 0.07	98.54 ± 0.04	90.78 ± 0.00	95.84 ± 0.18	96.92 ± 0.03	96.72 ± 0.21	98.91 ± 0.02	98.40 ± 0.06	98.87 ± 0.03	99.45 ± 0.02
	Reddit	98.31 ± 0.05	98.17 ± 0.05	98.47 ± 0.02	98.32 ± 0.06	99.01 ± 0.01	95.37 ± 0.00	97.42 ± 0.02	97.17 ± 0.02	99.02 ± 0.10	99.15 ± 0.01	99.19 ± 0.01	99.20 ± 0.03	99.62 ± 0.03
	MOOC	83.81 ± 2.09	85.03 ± 0.58	87.11 ± 0.19	93.56 ± 1.01	80.38 ± 0.26	60.86 ± 0.00	83.12 ± 0.18	84.01 ± 0.17	88.38 ± 0.71	87.91 ± 0.58	91.42 ± 0.09	92.52 ± 0.93	93.71 ± 0.52
	LastFM	70.49 ± 1.66	71.16 ± 1.89	71.59 ± 0.18	82.66 ± 2.94	85.92 ± 0.10	83.77 ± 0.00	64.06 ± 1.16	73.53 ± 0.12	86.94 ± 2.29	93.05 ± 0.10	92.21 ± 0.03	92.32 ± 0.29	96.70 ± 0.11
	Enron	87.96 ± 0.52	84.89 ± 3.00	68.89 ± 1.10	90.99 ± 0.99	90.45 ± 0.14	87.05 ± 0.00	75.74 ± 0.72	84.38 ± 0.21	92.02 ± 0.32	93.33 ± 0.13	92.77 ± 0.10	95.11 ± 0.08	98.07 ± 0.15
	Social Evo.	92.05 ± 0.46	90.76 ± 0.21	94.76 ± 0.16	90.36 ± 0.17	87.34 ± 0.08	81.60 ± 0.00	94.84 ± 0.17	95.23 ± 0.07	93.22 ± 0.13	96.30 ± 0.01	96.20 ± 0.03	96.36 ± 0.08	97.23 ± 0.01
	UCI	90.44 ± 0.49	68.77 ± 2.34	78.53 ± 0.74	92.17 ± 1.13	93.87 ± 0.08	77.30 ± 0.00	87.82 ± 1.36	91.81 ± 0.67	93.02 ± 0.48	94.49 ± 0.26	94.32 ± 0.16	95.53 ± 0.07	96.72 ± 0.25
	Flights	96.21 ± 1.42	95.95 ± 0.62	94.13 ± 0.17	97.99 ± 0.13	98.45 ± 0.01	90.23 ± 0.00	91.21 ± 0.02	91.13 ± 0.01	97.32 ± 0.34	98.93 ± 0.01	98.74 ± 0.01	98.99 ± 0.11	99.12 ± 0.02
	Contact	96.66 ± 0.89	96.48 ± 0.14	96.95 ± 0.08	95.37 ± 0.35	89.99 ± 0.34	94.34 ± 0.00	94.15 ± 0.09	93.94 ± 0.02	97.64 ± 0.58	98.53 ± 0.01	98.62 ± 0.01	98.69 ± 0.27	98.94 ± 0.33
	Avg. rank ↓	9.1	10.1	8.9	7.0	7.8	11.4	10.7	9.4	6.0	3.3	3.9	2.3	**1.0**
Ind-AP	Wikipedia	94.82 ± 0.20	92.43 ± 0.37	96.22 ± 0.07	97.49 ± 0.04	98.24 ± 0.03	–	96.22 ± 0.17	96.65 ± 0.02	95.40 ± 0.04	98.59 ± 0.03	97.76 ± 0.06	98.49 ± 0.05	99.10 ± 0.02
	Reddit	96.50 ± 0.13	96.09 ± 0.11	97.09 ± 0.04	97.26 ± 0.07	98.62 ± 0.01	–	94.09 ± 0.07	95.26 ± 0.02	98.56 ± 0.21	98.84 ± 0.02	98.82 ± 0.03	98.93 ± 0.09	99.37 ± 0.03
	MOOC	79.63 ± 1.92	81.07 ± 0.44	85.50 ± 0.19	91.86 ± 1.05	81.42 ± 0.24	–	80.60 ± 0.22	81.41 ± 0.21	83.59 ± 1.58	86.96 ± 0.43	88.71 ± 0.04	90.53 ± 1.06	93.27 ± 0.58
	LastFM	81.61 ± 3.82	83.02 ± 1.48	78.63 ± 0.31	87.18 ± 4.29	89.42 ± 0.07	–	73.53 ± 1.66	82.11 ± 0.42	86.87 ± 1.95	94.23 ± 0.09	94.00 ± 0.05	94.56 ± 0.38	95.89 ± 0.07
	Enron	80.72 ± 1.39	74.55 ± 3.95	67.05 ± 1.51	84.53 ± 1.02	86.35 ± 0.51	–	76.14 ± 0.79	75.88 ± 0.48	89.03 ± 0.83	89.76 ± 0.34	87.59 ± 0.09	91.67 ± 0.08	95.24 ± 0.23
	Social Evo.	91.96 ± 0.48	90.04 ± 0.47	91.41 ± 0.16	82.85 ± 0.86	79.94 ± 0.18	–	91.55 ± 0.09	91.86 ± 0.06	91.22 ± 0.32	93.14 ± 0.04	92.70 ± 0.07	92.84 ± 0.36	94.73 ± 0.04
	UCI	79.86 ± 1.48	57.48 ± 1.87	79.54 ± 0.48	82.04 ± 2.05	92.73 ± 0.06	–	87.36 ± 2.03	91.19 ± 0.42	87.30 ± 0.15	94.54 ± 0.12	93.58 ± 0.03	94.36 ± 0.14	95.91 ± 0.13
	Flights	94.74 ± 0.37	92.88 ± 0.73	88.73 ± 0.33	95.03 ± 0.60	97.06 ± 0.02	–	83.41 ± 0.07	83.03 ± 0.05	96.59 ± 1.67	97.79 ± 0.02	97.34 ± 0.01	97.85 ± 0.12	98.10 ± 0.10
	Contact	94.34 ± 1.45	92.18 ± 0.41	95.87 ± 0.11	88.56 ± 0.99	89.55 ± 0.30	–	91.11 ± 0.12	90.59 ± 0.05	96.79 ± 0.37	98.03 ± 0.02	97.58 ± 0.01	97.79 ± 0.83	98.05 ± 0.06
	Avg. rank ↓	8.9	10.0	8.9	7.4	6.8	–	9.6	8.9	6.9	2.8	4.2	2.6	**1.0**
Ind-AUC	Wikipedia	94.33 ± 0.27	91.49 ± 0.45	95.90 ± 0.09	97.08 ± 0.03	98.03 ± 0.04	–	95.57 ± 0.20	96.30 ± 0.04	94.74 ± 0.44	98.48 ± 0.03	97.45 ± 0.05	98.26 ± 0.05	99.10 ± 0.04
	Reddit	96.52 ± 0.13	96.05 ± 0.12	96.98 ± 0.04	96.94 ± 0.07	98.42 ± 0.02	–	93.80 ± 0.07	94.97 ± 0.05	97.99 ± 0.52	98.71 ± 0.01	98.69 ± 0.03	98.85 ± 0.09	99.35 ± 0.07
	MOOC	83.16 ± 1.30	84.03 ± 0.49	86.84 ± 0.17	92.02 ± 0.91	81.86 ± 0.25	–	81.43 ± 0.19	82.77 ± 0.24	86.13 ± 3.55	87.62 ± 0.51	89.94 ± 0.09	90.09 ± 0.99	93.97 ± 0.21
	LastFM	81.13 ± 3.39	82.24 ± 1.51	76.99 ± 0.29	85.58 ± 3.15	87.82 ± 0.12	–	70.84 ± 0.85	80.37 ± 0.18	83.07 ± 2.32	94.08 ± 0.08	93.62 ± 0.04	93.77 ± 0.38	96.17 ± 0.12
	Enron	81.96 ± 1.34	76.34 ± 4.20	64.63 ± 1.74	83.58 ± 1.11	87.02 ± 0.50	–	72.33 ± 0.99	76.51 ± 0.71	89.92 ± 0.72	90.69 ± 0.26	88.24 ± 0.07	92.57 ± 0.18	96.45 ± 0.32
	Social Evo.	93.70 ± 0.29	91.18 ± 0.49	93.41 ± 0.19	82.04 ± 0.59	84.73 ± 0.27	–	93.71 ± 0.18	94.09 ± 0.07	92.11 ± 0.07	95.29 ± 0.03	94.99 ± 0.03	95.03 ± 0.17	97.12 ± 0.03
	UCI	78.80 ± 0.94	58.08 ± 1.81	77.64 ± 0.38	86.48 ± 2.29	90.40 ± 0.11	–	84.49 ± 1.82	89.30 ± 0.57	83.81 ± 1.28	92.63 ± 0.13	91.31 ± 0.11	92.17 ± 0.17	94.68 ± 0.17
	Flights	95.21 ± 0.32	93.56 ± 0.70	88.64 ± 0.35	95.92 ± 0.43	96.86 ± 0.02	–	82.48 ± 0.01	82.27 ± 0.07	96.36 ± 1.51	97.80 ± 0.02	97.20 ± 0.01	97.95 ± 0.27	98.17 ± 0.02
	Contact	95.37 ± 0.92	91.89 ± 0.38	96.53 ± 0.10	88.87 ± 0.75	89.07 ± 0.34	–	93.05 ± 0.09	92.83 ± 0.05	96.67 ± 0.45	98.30 ± 0.02	97.91 ± 0.01	97.98 ± 0.67	98.37 ± 0.01
	Avg. rank ↓	8.7	9.9	8.8	7.4	7.0	–	9.9	8.8	7.0	2.7	4.2	2.7	**1.0**

Note. Best results are in bold and second-best results are underlined. Average ranks are computed across the nine datasets; lower is better.

D Formal Causality Property of S3Former

In this section, we formalize the temporal causality property of S3Former. The purpose of this analysis is not to provide a generalization or convergence guarantee, but to show that, under the online construction and update protocol used by S3Former, each prediction only depends on information available before the prediction time.

Let

$$\mathcal{G} = \{(u_i, v_i, t_i, e_i)\}_{i=1}^{N}$$

be a chronologically ordered continuous-time dynamic graph, where u_i and v_i are the endpoints of the i-th interaction, t_i is its timestamp, and e_i denotes the optional edge feature. For a prediction time t, we define the causal history as

$$\mathcal{G}_{<t} = \{(u_i, v_i, t_i, e_i) \in \mathcal{G} \mid t_i < t\}.$$

We also define the information set generated by the causal history as

$$\mathcal{F}_{<t} = \sigma(\mathcal{G}_{<t}).$$

For a candidate interaction (u, v, t), S3Former outputs

$$\hat{y}_{uv}(t) = \sigma(\ell_{uv}(t)),$$

where $\ell_{uv}(t)$ is the final prediction logit. We show that $\hat{y}_{uv}(t)$ is measurable with respect to $\mathcal{F}_{<t}$, meaning that the prediction is fully determined by the causal history and does not use events at or after timestamp t.

Proposition 1 (Temporal causality of S3Former). Assume that: (i) historical sequences, shared pair subgraphs, and statistical features are constructed only from $\mathcal{G}_{<t}$; (ii) online sequence buffers, structural indices, and statistical memories are updated only after all predictions or training losses at timestamp t have been computed; (iii) neural modules and sampling tie-breaking operations are deterministic given model parameters θ; and (iv) model parameters θ are fixed during the scoring of candidates at timestamp t. Then, for any candidate interaction (u, v, t), the prediction $\hat{y}_{uv}(t)$ produced by S3Former is a function of $(u, v, t, \mathcal{G}_{<t}, \theta)$ only. Equivalently,

$$\hat{y}_{uv}(t) \text{ is } \mathcal{F}_{<t}\text{-measurable.}$$

Therefore, no event with timestamp $\tau \geq t$ is used as an input to the prediction.

Proof. We prove the proposition by showing that each input channel of S3Former is constructed from the causal history $\mathcal{G}_{<t}$, and that all subsequent fusion and prediction operations are deterministic mappings applied to these causal inputs.

Sequential channel. For node u, S3Former constructs its historical interaction sequence before time t as

$$S_u^t = \{(u, w_j, \tau_j, e_{uw_j}) \mid \tau_j < t\}.$$

By definition, every interaction in S_u^t belongs to $\mathcal{G}_{<t}$. The same construction is used for node v, yielding S_v^t. Each historical interaction token is encoded using the historical neighbor feature, historical edge feature, and the relative time gap $t - \tau_j$, where $\tau_j < t$. Therefore, the token sequence for node u is fully determined by $(u, t, \mathcal{G}_{<t})$, and the token sequence for node v is fully determined by $(v, t, \mathcal{G}_{<t})$.

The long-term patch encoder, Transformer encoder, short-term causal convolution, and long-short adaptive fusion are deterministic neural mappings given model parameters θ. Hence, the sequential representations can be written as

$$h_u^{seq} = F_{seq}(u, t, \mathcal{G}_{<t}; \theta), \qquad h_v^{seq} = F_{seq}(v, t, \mathcal{G}_{<t}; \theta).$$

Thus, h_u^{seq} and h_v^{seq} are $\mathcal{F}_{<t}$-measurable.

Structural channel. The structural channel constructs historical one-hop neighborhoods

$$\mathcal{N}_u^{<t} = \{w \mid \exists \tau < t,\ (u, w, \tau, e) \in \mathcal{G}_{<t}\},$$

and similarly $\mathcal{N}_v^{<t}$ for node v. Since both neighborhoods are defined using only interactions before t, they are functions of $\mathcal{G}_{<t}$.

For a historical neighbor $c \in \mathcal{N}_u^{<t}$, the recency-frequency score is

$$s_u^t(c) = a_r \exp(-\lambda_r \Delta_{u,c}^{<t}) + a_f \log(1 + \mathrm{cnt}_{u,c}^{<t}),$$

where $\Delta_{u,c}^{<t}$ is the time gap since the latest interaction between u and c before t, and $\mathrm{cnt}_{u,c}^{<t}$ is their historical interaction count before t. Both quantities are computed only from $\mathcal{G}_{<t}$. Therefore, the selected recent anchor sets R_u^t and R_v^t are functions of $\mathcal{G}_{<t}$.

The common-neighbor set is defined as

$$C_{uv}^t = \mathrm{TopK}\left(\mathcal{N}_u^{<t} \cap \mathcal{N}_v^{<t}, s_u^t(c) + s_v^t(c)\right).$$

Because $\mathcal{N}_u^{<t}$, $\mathcal{N}_v^{<t}$, $s_u^t(c)$, and $s_v^t(c)$ are all functions of $\mathcal{G}_{<t}$, the common-neighbor set C_{uv}^t is also determined by $\mathcal{G}_{<t}$.

The node set of the shared pair subgraph is

$$V_{uv}^t = \{u, v\} \cup R_u^t \cup R_v^t \cup C_{uv}^t.$$

The induced edge set is

$$E_{uv}^t = \{(a, b) \mid a, b \in V_{uv}^t,\ \exists \tau < t,\ (a, b, \tau, e_{ab}) \in \mathcal{G}_{<t}\}.$$

Thus, both V_{uv}^t and E_{uv}^t are functions of $\mathcal{G}_{<t}$. The edge attributes used by the edge-aware GraphSAGE module, including historical counts, time-decayed strength, latest interaction gap, direction indicator, and aggregated historical edge features, are also computed only from interactions with timestamps smaller than t. Therefore, the shared pair subgraph and all its node and edge attributes are $\mathcal{F}_{<t}$-measurable.

The edge-aware GraphSAGE encoder and endpoint readout are deterministic neural mappings given θ. Therefore, the structural representations and structural quality descriptors can be written as

$$h_u^{str}, h_v^{str}, q_u^{str}, q_v^{str} = F_{str}(u, v, t, \mathcal{G}_{<t}; \theta).$$

Hence, the structural channel is temporally causal.

Statistical channel. S3Former computes node-level statistics using only historical events. For node u, the node-statistical vector is

$$\phi_u(t) = [\mathrm{cnt}_u^{<t}, \deg_u^{<t}, \mathrm{rec}_u^{<t}, \mathrm{act}_u^{<t}],$$

where $\mathrm{cnt}_u^{<t}$ is the historical interaction count, $\deg_u^{<t}$ is the number of distinct historical neighbors, $\mathrm{rec}_u^{<t}$ is the latest historical interaction gap, and $\mathrm{act}_u^{<t}$ is the time-decayed activity computed from events before t. Each term is a function of $\mathcal{G}_{<t}$. Therefore,

$$h_u^{stat} = F_{node_stat}(u, t, \mathcal{G}_{<t}; \theta),$$

and the same argument applies to h_v^{stat}.

The pair-level statistical vector is

$$\psi_{uv}(t) = \psi_{uv}^{direct}(t) \oplus \psi_{uv}^{stability}(t) \oplus \psi_{uv}^{overlap}(t) \oplus \psi_{uv}^{time}(t).$$

The direct group includes historical pair frequency, latest historical pair-interaction gap, and multi-scale time-decayed pair strength. The stability group measures long-term stability and burstiness using historical pair interactions. The overlap group uses common-neighbor count and Jaccard overlap computed from $\mathcal{N}_u^{<t}$ and $\mathcal{N}_v^{<t}$. The temporal group uses pair-specific time context derived from historical interactions. Therefore, every component of $\psi_{uv}(t)$ is computed from $\mathcal{G}_{<t}$, and

$$h_{uv}^{mem} = F_{pair_stat}(u, v, t, \mathcal{G}_{<t}; \theta).$$

Importantly, the online sequence buffers, structural indices, and statistical memories are updated only after the prediction or training loss at timestamp t has been computed. Thus, the memory state queried for prediction at time t contains only events from $\mathcal{G}_{<t}$, not events at timestamp t or after t.

Fusion and prediction. NodeCCF obtains endpoint representations by applying deterministic gates and residual corrections to the sequential, structural, and statistical representations:

$$h_u = F_{NodeCCF}(h_u^{seq}, h_u^{str}, h_u^{stat}, q_u^{str}; \theta),$$

and similarly for h_v. Since all inputs to NodeCCF are $\mathcal{F}_{<t}$-measurable, the endpoint embeddings h_u and h_v are also $\mathcal{F}_{<t}$-measurable.

PairCCF first constructs the base pair representation

$$h_{uv}^{base} = h_u \oplus h_v \oplus (h_u \odot h_v) \oplus |h_u - h_v|.$$

It then computes pair-level adaptive correction using the causal pair memory h_{uv}^{mem}. Since both h_{uv}^{base} and h_{uv}^{mem} are $\mathcal{F}_{<t}$-measurable, the corrected pair representation h_{uv}^{pair} is also $\mathcal{F}_{<t}$-measurable.

Finally, the prediction head computes

$$\ell_{uv}(t) = f_{pred}(h_{uv}^{pair}), \qquad \hat{y}_{uv}(t) = \sigma(\ell_{uv}(t)).$$

Both operations are deterministic given θ. Therefore, $\hat{y}_{uv}(t)$ is a deterministic function of $(u, v, t, \mathcal{G}_{<t}, \theta)$, and is $\mathcal{F}_{<t}$-measurable.

Thus, no event with timestamp $\tau \geq t$ is used by S3Former when predicting (u, v, t). This proves the proposition. □

Corollary 1 (No current-timestamp leakage). Suppose all candidate interactions at timestamp t are scored before any positive event at timestamp t is inserted into the online sequence buffers, structural indices, or statistical memories. Then the prediction for any candidate (u, v, t) does not depend on any other event occurring at the same timestamp t.

Proof. Under the stated update protocol, all candidates at timestamp t query the same historical state constructed from $\mathcal{G}_{<t}$. Events with timestamp t are inserted into the online memories only after all predictions or training losses at timestamp t have been computed. Therefore, for every candidate (u, v, t), the sequential channel, structural channel, statistical channel, NodeCCF, PairCCF, and prediction head receive inputs constructed from $\mathcal{G}_{<t}$ only. By Proposition 1, the prediction is $\mathcal{F}_{<t}$-measurable and cannot depend on other events at timestamp t. □

Remark on negative sampling. The causality property concerns the model input and prediction for a fixed candidate interaction. In the evaluation protocol, same-timestamp positive edges may be used to filter sampled negatives and avoid false negative candidates. This filtering step is part of benchmark construction and does not provide S3Former with additional input features. Once a positive or negative candidate (u, v, t) is fixed, all historical sequences, shared pair subgraphs, and statistical features used by S3Former are constructed strictly from $\mathcal{G}_{<t}$.

E Robustness under Historical Negative Sampling

The four-dataset experiment is reported in main Table 4. We additionally evaluate eight representative datasets under all four main metrics, with DyGFormer and TPNet as temporal-structural comparisons. Table 12 reports the corresponding mean results. Boldface denotes only the largest point estimate in each row/metric; it does not denote statistical significance.

Table 12: Extended historical-negative evaluation over eight datasets and four metrics. Entries are mean Trans-AP/Trans-AUC/Ind-AP/Ind-AUC values (%).

Dataset	DyGFormer	TPNet	S3Former
Wikipedia	82.23/78.80/71.42/68.33	81.55/79.89/71.28/67.95	**85.73/80.21/72.56/69.20**
Reddit	**81.57**/80.54/**65.37**/64.81	81.02/**81.87**/62.15/62.37	81.11/80.35/65.02/**64.97**
MOOC	85.85/87.04/80.82/80.77	92.69/**93.45**/81.85/84.46	**92.74**/93.01/**87.36/87.33**
Enron	75.63/76.55/67.07/65.78	**80.79/81.16/74.60/74.50**	78.32/80.11/68.32/68.50
UCI	82.17/76.97/72.13/65.55	**86.34/80.42/78.48/71.35**	83.15/76.83/72.21/65.81
Flights	66.59/68.09/57.11/56.05	69.10/71.82/54.67/53.08	**71.83/73.56/61.89/61.51**
Social Evo.	97.38/97.28/96.82/96.91	96.80/96.22/96.38/96.76	**98.12/97.52/97.01/97.21**
Contact	97.57/97.17/93.56/94.14	98.02/97.73/93.56/93.47	**98.31/97.89/94.51/94.21**

S3Former has the largest point estimate in 20 of 32 combinations, while TPNet is stronger on Enron and UCI and the methods are close on Reddit. We additionally perform a targeted MOOC check using 61,763 true test interactions and 61,763 historical negatives: Full S3Former obtains 92.70 AP/92.99 AUC, removing the statistical pathway obtains 87.57/87.13, and Sequence+Structure obtains 90.11/90.10. This targeted result supports usefulness beyond the simplest unseen-pair shortcut, but it is a single-dataset diagnostic rather than evidence that every statistical descriptor is universally necessary.

F Fusion Ablation

We next evaluate whether S3Former benefits from hierarchical context-aware fusion. The `concat_fusion` variant replaces NodeCCF and PairCCF with direct feature concatenation, while `no_both_ccf` removes both CCF modules and uses a simplified fusion head. This comparison tests whether the improvement comes merely from adding more feature channels or from adaptive cross-channel coordination.

As shown in Figure 3, both `concat_fusion` and `no_both_ccf` consistently underperform the full model. These results suggest that the gain of S3Former does not simply come from concatenating sequential, structural, and statistical features. Instead, NodeCCF and PairCCF provide useful adaptive coordination: NodeCCF determines how much structural and statistical evidence should participate in endpoint representation learning, while PairCCF performs candidate-specific relation correction using pair memory.

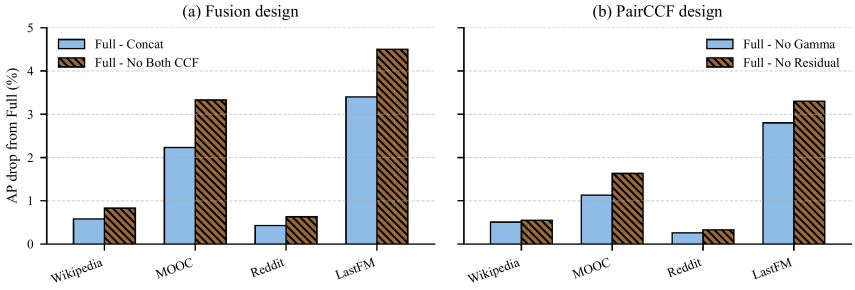


Figure 3: CCF ablation measured by AP drop from the full S3Former. Larger drops indicate stronger contribution.

G PairCCF Ablation

Finally, we analyze the internal design of PairCCF. The `pair_no_gamma` variant removes the adaptive coefficient γ, which controls how strongly pair-level correction intervenes. The `pair_no_res` variant removes the pair-level residual correction term. These variants evaluate whether PairCCF is more than a regular pairwise prediction head and whether its retreatable correction mechanism is useful.

Table 13: PairCCF ablation for S3Former. We report transductive AP (%).

Dataset	Pair No Gamma	Pair No Res	Full
Wikipedia	*98.92*	*98.88*	99.43
MOOC	*91.90*	*91.40*	93.03
Reddit	*99.42*	*99.35*	99.68
LastFM	*93.80*	*93.30*	96.60

As shown in Table 13, removing either component consistently reduces performance. Removing γ prevents the model from deciding when pair-level memory should strongly intervene and when it should retreat to the base endpoint representation. Removing the residual term also reduces performance, indicating that PairCCF provides useful relation-level correction beyond the base pair representation. Overall, these results support the design of PairCCF as an adaptive correction mechanism rather than a rigid prediction head.

H Structural Mechanism Analysis

We use an induced-topology diagnostic to quantify historical edges among selected pair-centered nodes that are not explicitly exposed by endpoint-history serialization. For a candidate pair (u, v, t), let $\mathcal{E}_{uv}^{t}$ be the causal induced edge set among the selected pair-centered nodes, and let $\mathcal{E}_{uv}^{t,\mathrm{star}}$ denote the endpoint-star edges directly involving u or v in the serialized histories. We define

$$\Delta_{\mathrm{ind}}(u, v, t) = \frac{|\mathcal{E}_{uv}^{t} \setminus \mathcal{E}_{uv}^{t,\mathrm{star}}|}{\max(1, |\mathcal{E}_{uv}^{t}|)}.$$

This diagnostic does not imply that sequence, walk, or co-occurrence models cannot learn structural correlations; it only measures the amount of induced-edge information absent from serialized endpoint histories.

(a) Structural isolation

Variant	Wikipedia	MOOC	LastFM	UCI
No topology	99.25	85.72	94.10	96.36
+ CN/Jaccard	99.28	87.60	94.35	96.50
+ Binary induced edges	99.35	90.80	94.95	96.82
+ Edge-aware EdgeSAGE	99.43	93.03	95.50	97.05
S3Former full	99.43	93.03	96.60	97.25

(b) Topology-bucket gain

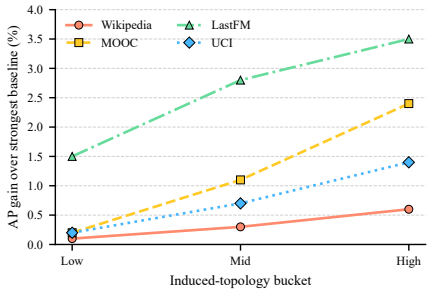


Figure 4: Structural mechanism diagnostics. Left: structural isolation under the same sequence-statistical backbone. Right: topology-bucketed AP gain over the strongest baseline.

The structural-isolation comparison uses the same sequence-statistical backbone and selected node set. CN/Jaccard features provide only mild gains, while adding binary induced edges and edge-aware EdgeSAGE yields larger improvements, indicating that explicit historical edges contain information beyond overlap statistics. The topology buckets show increasing gains from low- to high-topology candidates, supporting the intended role of the structural channel when candidate-centered histories contain richer local topology.

I TGB Wall-Clock Measurements

Scaling shared pair-subgraph extraction to billion-edge dynamic graphs while preserving strict temporal causality also presents an important challenge for advancing real-world dynamic graph systems.

Table 14: TGB wall-clock time in seconds on one RTX 4090. "Train" is per epoch; "–" denotes an unavailable matched TPNet run.

Dataset	S3Former Train	S3Former Val	S3Former Test	TPNet Train
Wiki	32	647	648	33
Review	472	3,684	2,074	1,051
Coin	3,680	2,390	2,435	2,643
Flight	3,566	5,709	5,358	–

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims about S3Former's sequential, structural, statistical, and fusion components. The experimental scope and protocol-specific average-rank claim are supported by Table 1; the boundary condition on recurrence-heavy institutional datasets is discussed in Appendix B.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix B discusses the boundary condition where long-term pair recurrence can dominate structural signals, and Appendix I notes the scalability challenge of shared pair-subgraph extraction for billion-edge dynamic graphs. The main scalability discussion also avoids claiming uniform efficiency superiority.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The formal causality property is provided in Appendix D. Proposition 1 states the assumptions for temporal causality and gives a proof, followed by Corollary 1 on no current-timestamp leakage.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The Method section specifies the model architecture, causal input construction, prediction head, and training objective. The Experiments section and Appendix A specify datasets, chronological splits, negative sampling, metrics, optimizer, model-selection criterion, repeated runs, and dataset-specific hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The datasets used in this work are publicly available, and a cleaned S3Former implementation is publicly available at `https://github.com/lzy4123/S3Former`. The repository documents environment setup, Enron data preparation, training, and evaluation, but it does not yet provide instructions and assets for reproducing the full multi-benchmark experimental suite reported in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The Experiments section specifies the datasets, baselines, metrics, chronological split, optimizer, loss, model-selection criterion, and number of runs. Appendix A provides dataset statistics, negative-sampling details, hardware, and dataset-specific hyperparameters; Section 4.6 specifies the official TGB evaluation protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The conventional AP/AUC experiments are reported over five random seeds, and Appendix C provides mean ± standard deviation over five runs for the locally evaluated baselines and S3Former. The TPNet entries in Table 1 are published five-run means. The extended historical-negative appendix table reports means without uncertainty estimates, and no statistical-significance claim is made for that table. The TGB ranking results in Table 5 are single-run measurements, and no multi-run statistical significance is claimed for them.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A details the hardware configuration used for the main experiments, including one NVIDIA GeForce RTX 4090 D GPU with 24GB of memory. Appendix I also provides detailed measurements of training, validation, and test times, as well as timing results for the TGB evaluations and TPNet runs. These measurements provide representative information on the computational cost required to reproduce the reported experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: This work conforms to the NeurIPS Code of Ethics. The study uses publicly available benchmark datasets for dynamic graph link prediction, does not collect new personal data, does not involve human subjects or crowdsourcing, and does not release high-risk models or datasets.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The current camera-ready manuscript does not include a dedicated discussion covering both potential positive and negative societal impacts of dynamic link prediction. The paper uses public benchmark datasets, but this does not replace the broader-impact analysis requested by this checklist item.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not introduce or release high-risk assets such as pretrained language models, image generators, or newly scraped datasets; the experiments use public continuous-time dynamic graph benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the original publications for the existing datasets and codebases, but the current camera-ready manuscript does not explicitly enumerate the license names and terms of use for all assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: We release a cleaned implementation of S3Former at `https://github.com/lzy4123/S3Former`. The repository documents environment setup, Enron data preparation, training, and evaluation, but documentation for the full multi-benchmark release and an explicit software license are not currently provided.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not describe crowdsourcing experiments or new research with human subjects; it evaluates models on public benchmark datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or new research with human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology described in the paper does not use LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.