BEYOND PAIR STATISTICS: RELATIONAL EXTRAPOLATION FOR TEMPORAL GRAPH LINK PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal link prediction aims to infer future interactions from evolving node histories in temporal graphs. Existing methods are broadly query-agnostic or query-aware, with the latter often achieving stronger overall performance. However, separating repeated and first-seen queries reveals that this advantage concentrates on repeated interactions and can weaken or even reverse on first-seen ones. We identify pair statistics, such as edge recurrence and neighbor co-occurrence, as a key source of this behavior. These statistics provide highly predictive shortcuts when supported by historical interactions, but become unavailable or non-discriminative beyond their support. We call this *Pair-Statistic Support Bias*, under which overall evaluation can conflate pattern exploitation with relational generalization. To address this limitation, we propose STATE, a simple yet effective query-aware approach for link prediction on temporal graphs. We design a stable temporal node memory and provide a tracking-stability analysis; building on it, we propose relational extrapolation encoding for structure learning. Specifically, each source-side historical neighbor serves as a witness and is compared with the targets before aggregation, enabling potential relations to be inferred beyond predefined patterns and generating relational tokens. We then aggregate these relational tokens into node representations. We further provide a unified view: recurrence is directly exposed through witness identity, co-occurrence is indirectly mediated through temporal memories shaped by historical interactions, and unsupported pairs rely on memory-based extrapolation. Across ten temporal graph benchmarks and eleven baselines, STATE achieves up to 17.93% relative MRR improvement over the strongest baseline.

1 INTRODUCTION

Temporal link prediction is a fundamental problem in dynamic systems, including social networks (Feng et al., 2025), financial networks (Shamsi et al., 2025), and road networks (Kumar et al., 2019). Most methods encode the historical interactions of the two query endpoints, differing mainly in whether each history is modeled independently or conditioned on the other endpoint. Query-agnostic methods (Kumar et al., 2019; Xu et al., 2020; Cong et al., 2023; Rossi et al., 2020; Tieu et al., 2025; Raghuvanshi et al., 2026), including GraphMixer (Cong et al., 2023) and L-STEP (Tieu et al., 2025), encode the two endpoint histories independently. Their representations are therefore reused across candidate targets, leaving target-specific signals to the final decoder. Query-aware methods (Yu et al., 2023; Lu et al., 2024; Yu et al., 2025; Tian et al., 2024; Liu et al., 2025; Yi et al., 2025a), in contrast, introduce target information before the final decoder. One group (Luo & Li, 2022; Yu et al., 2023; Lu et al., 2024; Yu et al., 2025; Liu et al., 2025), including DyGFormer (Yu et al., 2023) and SALoM (Liu et al., 2025), augments neighbor representations with query-pair statistics such as neighbor co-occurrence. Another group (Karmim et al., 2024; Yi et al., 2025a), including CRAFT (Yi et al., 2025a), uses target-dependent weighting to aggregate historical neighbors. In general, query-aware methods achieve better overall performance.

To identify the source of query-aware gains, we conduct a diagnostic study on UCI and sx-askubuntu, whose repeated-edge ratios are 66% and 38%, respectively. We partition queries into *repeated* and *first-seen* cases according to whether their endpoints have interacted before prediction. We compare the representative query-agnostic GraphMixer (Cong et al., 2023) against three query-aware methods that exploit distinct pair statistics: DyGFormer (Yu et al., 2023), CRAFT (Yi et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2025a), and SALoM (Liu et al., 2025). Specifically, they use neighbor co-occurrence, repeat frequency, and identity matching, respectively. Fig. 1 and Appendix Fig. 5 reveal two key observations.

Observation I: Existing query-aware mechanisms favor pattern exploitation. On repeat-dominant UCI, statistic-free GraphMixer outperforms the three query-aware methods by an absolute 12.4%–14.3% MRR on first-seen queries (purple solid bars), but trails them by 25.3%–27.8% on repeated queries (red solid bars). Because repeated interactions dominate UCI, this large repeated-query advantage largely drives the stronger overall performance of query-aware methods. On the less repetitive sx-askubuntu, GraphMixer gains 22.9%–46.4% on first-seen queries while remaining competitive on repeated ones. Thus, the advantage of query-aware methods is strongest when predefined patterns are present, but does not reliably extend beyond such support.

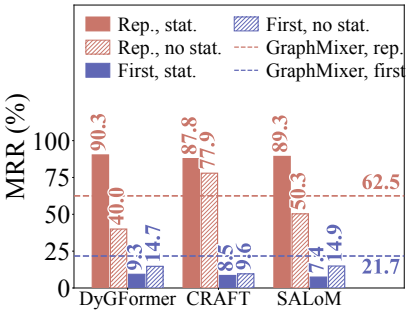


Figure 1: Effect of removing three types of pair statistics on repeated and first-seen queries on UCI.

Observation II: Much of this exploitation advantage arises from pair-statistic shortcuts. Across both datasets, removing the corresponding statistic reduces repeated-query MRR by an absolute 9.0%–50.3% (hatched bars), while largely preserving first-seen performance and improving it by up to 7.5%. Pair statistics therefore provide highly predictive evidence within their predefined structural support, but their benefit does not reliably transfer beyond it. The improvement in first-seen prediction after removing these statistics further suggests that excessive reliance on such shortcuts may bias optimization toward repeated patterns.

Together, these observations reveal *Pair-Statistic Support Bias*: statistic-dependent query-aware models benefit disproportionately when predefined historical patterns are present. Consequently, overall evaluation can conflate pattern exploitation with relational generalization, masking weaker inference beyond pair-statistic support.

To address this bias, we make three main contributions toward relational generalization beyond pair-statistic support. First, we propose STATE, a simple yet effective query-aware temporal link predictor that shifts query awareness from predefined pattern lookup to learnable relational encoding. We design a stable temporal node memory for relational comparison and provide a tracking-stability analysis for its update. Second, we propose a relational extrapolation encoding for structural learning, in which each historical neighbor serves as a witness and is compared with the candidate targets before aggregation, allowing target-dependent relational evidence to be constructed beyond predefined patterns and generating relational tokens for each neighbor. We then aggregate the resulting relational tokens and features via a lightweight aggregator for node representations. Third, we provide a unified view of pair statistics and relational extrapolation: recurrence is directly exposed through witness identity, co-occurrence is mediated through temporal memories shaped by previous interactions, and unsupported pairs rely on learned memory-based extrapolation. We formalize recurrence and co-occurrence as fixed instances of a common witness–target matching operator, linking predefined pair-statistic matching to learned relational inference. Across ten datasets, STATE achieves up to 17.93% relative MRR improvement against eleven baselines, while improving first-seen prediction without sacrificing repeated-link performance. **Together, theoretical and empirical analyses show that a single learned witness–target relation is sufficient to unify predefined pattern exploitation and relational extrapolation, remaining effective both within and beyond pair-statistic support.**

2 Related Work

We organize the literature on link prediction over temporal graphs (Wang et al., 2025a; Ur Rahman et al., 2025; Lachi et al., 2025; Sun et al., 2025; Li et al., 2025) by whether target information enters neighborhood modeling before final link scoring.

Query-Agnostic Modeling for Temporal Link Prediction. Given a query $(s, d; t)$, query-agnostic methods (Zuo et al., 2018; Kumar et al., 2019; Sankar et al., 2020; Jin et al., 2022; Wang

et al., 2021b; Zhang et al., 2022; 2023; Bai et al., 2023; Pareja et al., 2020; Trivedi et al., 2019; Xu et al., 2024a; Heeg & Scholtes, 2024; Li et al., 2023; Wang et al., 2021a; Zhang et al., 2025a; Zou et al., 2026; Xu et al., 2024b; Wang et al., 2025b; Wu et al., 2025; 2024; Sun & Gong, 2026; Zhang et al., 2025b; Li et al., 2024; Zhao & Zhang, 2024; Zhou et al., 2025; Tieu et al., 2025; Castelli et al., 2026; Yang et al., 2026; Shi et al., 2026; Lai et al., 2026; Wang et al., 2026; Mo et al., 2026; Raghuvanshi et al., 2026) construct temporal representations of s and d from their respective histories without explicitly conditioning one endpoint's neighborhood representation on the other endpoint. GraphMixer (Cong et al., 2023) and TGAT (Xu et al., 2020), for example, encode neighbors using edge features and temporal information, while L-STEP (Tieu et al., 2025) models target-independent structural and temporal signals in the frequency domain. These embeddings can be reused across candidate pairs, which is computationally attractive, but target-dependent discrimination is deferred to the decoder rather than expressed at the level of individual historical neighbors.

Query-Aware Modeling for Temporal Link Prediction. Query-aware methods (Luo & Li, 2022; Yu et al., 2023; Lu et al., 2024; Cheng et al., 2025; Tian et al., 2024; Zou et al., 2024; Yu et al., 2025; Yi et al., 2025a; Liu et al., 2025) introduce target context before final scoring. Existing approaches do so mainly through neighbor-token construction or target-conditioned aggregation. The former injects explicit pair-dependent statistics into tokens, including co-occurrence encodings in DyGFormer (Yu et al., 2023) and SALoM (Liu et al., 2025). The latter uses the target to reweight historical neighbors during aggregation, as in CRAFT (Yi et al., 2025a) and SLATE (Karmim et al., 2024). These methods incorporate target information into neighborhood modeling. In particular, explicit statistic encodings are tied to predefined historical patterns, while target-conditioned weighting primarily changes which observed neighbors are emphasized. Our design instead makes the learned witness–target relation itself an explicit token component before pooling.

Beyond temporal link prediction, recent work (Feldman et al., 2026; Tian et al., 2026; Chmura et al., 2026) has studied node-affinity prediction, dynamic graph anomaly detection, and unified infrastructure for node- and graph-level temporal graph learning.

3 PRELIMINARIES

Definition 1 (Temporal Graph)**.** *A temporal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ is constructed by a sequence of temporal interaction events $\{\delta(t_1), \delta(t_2), \cdots\}$ ordered by timestamps, where $\mathcal{V}$ is the node set, $\mathcal{E}$ is the edge set, and $\mathcal{X}$ is the attribute matrix. Each temporal interaction event is a tuple $\delta(t) = (v_i, v_j, \boldsymbol{e}_{ij}(t), t)$ representing a temporal edge, where v_i and v_j are nodes, $\boldsymbol{e}_{ij}(t)$ denotes edge features, and t is the timestamp at which $\delta(t)$ happens.*

Problem statement. Given a temporal graph $\mathcal{G}$ and a query $(s, d; t)$, link prediction estimates whether source node s and destination node d will interact at time t. This requires learning node embeddings $\mathbf{z}_s(t), \mathbf{z}_d(t)$ from historical interactions before t, and using them to score the likelihood of the query in the ranking setting.

4 THE PROPOSED STATE

We propose STATE, a query-aware temporal link predictor that performs relational extrapolation without explicit pair statistics (Fig. 2). The key idea is that relational evidence in observed interactions can be extrapolated to potential links that have not been explicitly observed. STATE implements this principle with three components: stable temporal node memory, relational extrapolation encoding, and lightweight temporal neighbor aggregation.

4.1 STABLE TEMPORAL NODE MEMORY

Relational extrapolation requires comparing a node with the candidate targets, even when their interaction has never been observed. The comparison therefore requires temporal node states that summarize prior interactions while remaining stable under short-term event fluctuations. We maintain a smoothed memory for each node to provide such states. Formally, each node v maintains a memory vector $\mathbf{m}_v(t)$, where $\mathbf{m}_v(0)$ is Gaussian-initialized and reset at each epoch. For prediction at time t, only the pre-query memory $\mathbf{m}_v(t^-)$ is used ($t^- < t$); events at time t are applied

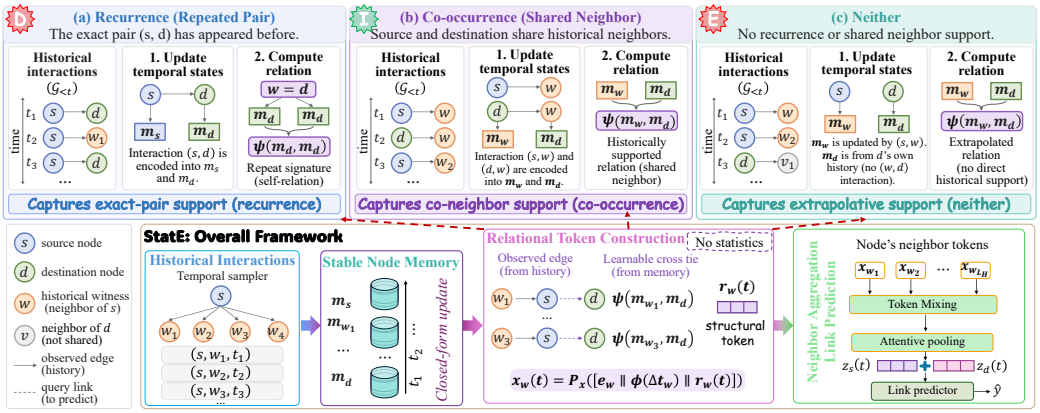


Figure 2: Top: StatE's relational modeling across three representative structural regimes. Bottom: the overall StatE framework, consisting of stable temporal memory, relational token construction, temporal neighbor aggregation, and temporal link prediction.

after scoring. Let t_v^- denote the latest memory-update time of v. We define the retention factor $\rho_v(t) = \exp[-\lambda(t - t_v^-)/3600]$, where timestamps are in seconds, $/3600$ converts elapsed time to hours, and $\lambda > 0$ is the decay rate. Given the new representation $\mathbf{z}_v(t)$, the memory is updated as

$$\widetilde{\mathbf{z}}_v(t) = (1-\beta)\mathbf{z}_v(t) + \beta \mathbf{m}_v(t^-), \;\; \mathbf{m}_v(t) = \rho_v(t)\mathbf{m}_v(t^-) + \big(1-\rho_v(t)\big)\widetilde{\mathbf{z}}_v(t), \tag{1}$$

where $\beta \in [0, 1)$ is a momentum coefficient. For short update intervals, $\rho_v(t)$ remains close to one and limits abrupt memory changes; after longer inactivity, a smaller retention factor assigns more weight to the new representation. The momentum term further smooths event-level variation. The resulting memory is used as the temporal state for structure learning.

Remark 1. *Theorem 1 in Appendix B.1 gives a per-step tracking-error decomposition into propagated error, latent structural drift, and event-level residual. Remark 3 further gives a uniform bound under bounded drift, bounded residuals, and a contraction condition. These results motivate the memory as a stable input to the relational encoder; they do not assume that the latent structural state is exactly observed.*

4.2 Relational Extrapolation Encoding

For each source node s at time t, useful ranking evidence must vary across candidate targets. Query-agnostic methods (Cong et al., 2023; Tieu et al., 2025) encode historical neighbors independently of the target, while statistic-based query-aware methods (Yu et al., 2023; Tian et al., 2024; Liu et al., 2025) often introduce target dependence through predefined patterns including edge recurrence and neighbor co-occurrence. Considering random negative sampling, these statistics can be highly predictive: a repeated positive pair has an explicit recurrence signal and may also exhibit stronger neighborhood overlap, whereas a sampled negative is often first-seen and carries neither pattern. The resulting predictor can therefore benefit from recognizing predefined patterns rather than learning a relation that remains target-dependent beyond their support. This motivates a query-aware mechanism that makes historical evidence target-dependent without tying it to explicit pair-statistic support. We therefore propose relational extrapolation encoding for structure modeling, which infers target-dependent relations from temporal node states before aggregation rather than retrieving pair-specific interactions. Because the relation is inferred from node states, compatibility can still be assessed for witness–target pairs that have never directly interacted. Before formalizing this relational encoding, we first describe the temporal neighbor sampling for constructing the historical context.

Temporal Neighbor Sampling. For each endpoint $v \in \{s, d\}$, we sample multi-hop neighbors using temporal sampler (Yu et al., 2023), encode each neighbor into a relational token, and assemble the resulting tokens with hop decay. To unify one-hop and multi-hop expansion, we initialize $\mathcal{N}_v^{(0)}(t) = \{(v, t)\}$, and recursively define, for $h = 1, \ldots, H$, $\mathcal{N}_v^{(h)}(t) = \bigcup_{(u,t_u)\in\mathcal{N}_v^{(h-1)}(t)} \{(w, t') : (u, w, t') \in \mathcal{E},\ t' < t_u\}$, where we sample K most recent timestamped

neighbors for one hop. We then form the multi-hop neighborhood as $\mathcal{N}_v(t) = \bigcup_{h=1}^{H} \mathcal{N}_v^{(h)}(t)$, and assign each sampled neighbor w a hop label $\{1, \ldots, H\}$. Intuitively, one-hop neighbors provide direct local evidence, whereas higher-hop neighbors contribute complementary structural context beyond immediate interaction.

Relational Extrapolation Encoding. We formulate relational extrapolation encoding by modeling the unresolved relation between each historical witness and the target. Given a query $(s, d; t)$, a source-side historical witness (neighbor) $w \in \mathcal{N}_s(t)$ is already anchored to s through its sampled historical path and the associated edge, temporal, and hop information. For fixed (s, t), this source-side context is invariant across targets (e.g., d), whereas the witness–target relation varies with d and directly affects ranking. We thus model only the target-dependent relation between w and d. Let $\mathbf{m}_v = \mathbf{m}_v(t^-)$ denote the pre-query memory of node v. For each source-side historical witness $w \in \mathcal{N}_s(t)$, we define its relational extrapolation encoding as

$$\mathbf{r}_w(t) = \psi(\mathbf{m}_w, \mathbf{m}_d) = \mathrm{MLP}_\psi(\mathrm{LN}(\mathbf{m}_w - \mathbf{m}_d) + \boldsymbol{\eta}), \tag{2}$$

where MLP_ψ is a multilayer perceptron and $\boldsymbol{\eta}$ is a learnable offset. The difference $\mathbf{m}_w - \mathbf{m}_d$ in equation 2 captures the feature-wise relative temporal state between the witness and target. Layer normalization standardizes this difference across dimensions, reducing sensitivity to its overall scale while retaining its standardized cross-dimensional discrepancy pattern. Thus, ψ does not impose a fixed distance-based similarity rule. Instead, $\mathrm{MLP}_\psi(\cdot)$ learns to map normalized relative-state patterns into relational tokens, allowing different configurations to encode diverse relational compatibility. As the target state $\mathbf{m}_d$ changes, the same historical witness induces different relation encodings, yielding target-dependent evidence for ranking. Since this relation is inferred from temporal node states rather than retrieved from pair-specific observations, it remains well-defined even for witness–target pairs that have never directly interacted, provided that their memories are informative. We apply the same relation operator to destination-side neighbors to encode the target's own temporal context. Next, we clarify how our relational extrapolation relates to pair statistics together with Fig. 2.

Relation to Pair Statistics. For clarity, we characterize pair statistics using one-hop historical neighbors. Let $c_v(w; t)$ denote the number of occurrences of node w in the sampled one-hop history of v before time t, and define $T_\kappa(s, d; t) = \sum_w c_s(w; t)\kappa(w, d; t)$, where $\kappa(w, d; t)$ specifies a witness–target matching rule before aggregation.

Recurrence (direct support). A repeat-based signal, such as that used in CRAFT (Yi et al., 2025a), can be expressed by setting $\kappa_{\mathrm{rep}}(w, d; t) = \mathbb{1}\{w = d\}$, which yields $T_{\kappa_{\mathrm{rep}}}(s, d; t) = c_s(d; t)$. Thus, recurrence is directly exposed through witness identity. When d itself appears as a source-side witness, STATE evaluates the self-relation $\psi(\mathbf{m}_d, \mathbf{m}_d)$ without an explicit recurrence feature, providing explicit signals in equation 2.

Co-occurrence (indirect support). Neighbor co-occurrence, as used in methods such as DyGFormer (Yu et al., 2023), can be expressed by setting $\kappa_{\mathrm{co}}(w, d; t) = c_d(w; t)$, giving $T_{\kappa_{\mathrm{co}}}(s, d; t) = \sum_w c_s(w; t)c_d(w; t)$, the frequency-weighted overlap of the two endpoint histories. In STATE, this support is indirect: a previous interaction (w, d) has already shaped the pre-query states $\mathbf{m}_w$ and $\mathbf{m}_d$ before $\psi(\mathbf{m}_w, \mathbf{m}_d)$ is evaluated. The learned relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$ can therefore exploit shared-neighbor evidence without explicitly computing co-occurrence.

Neither (extrapolation). When $w \neq d$ and w has not previously interacted with d, neither recurrence nor co-occurrence support is available. However, $\psi(\mathbf{m}_w, \mathbf{m}_d)$ remains well-defined based on memory, allowing target-dependent relations to be inferred from temporal memory beyond predefined pattern support.

Insight. A simple learned witness–target relation is sufficient to unify the roles of multiple pair statistics: it exploits supported historical patterns when available and extrapolates beyond them when they are not.

Remark 2. *Proposition 1 of Appendix B.2 formalizes these fixed matching rules. Their link to* STATE *is structural and does not require $\psi(\mathbf{m}_w, \mathbf{m}_d)$ to reconstruct the discrete pair statistics. Under this view, recurrence is directly exposed through witness identity, co-occurrence is mediated through temporal memories shaped by prior witness–target interactions, and unsupported pairs rely*

on the same relation for memory-based extrapolation. Thus, STATE *replaces multiple predefined matching rules with a single learned relation that remains applicable beyond their support.*

Relational Token Construction. We combine the learned relation with the observed edge features $\mathbf{e}(t)$ and its time information. Each hop is assigned a learnable scalar weight ω_h, while padded positions receive zero weight. For each occurrence (w, t_w), we construct the token by

$$\mathbf{x}_w(t) = \mathbf{P}_x([\omega_{h_w}\mathbf{e}_w(t) \parallel \omega_{h_w}\boldsymbol{\phi}(\Delta t_w) \parallel \mathbf{P}_r(\omega_{h_w}\mathbf{r}_w(t))]), \tag{3}$$

where $\boldsymbol{\phi}(\cdot)$ denotes the time encoding (Xu et al., 2020), and $\mathbf{P}_r$ and $\mathbf{P}_x$ denote linear projections. Padding positions are masked before and after projection so that projection biases cannot create spurious tokens. Equation 3 therefore combines observed event information, temporal and structural position, and target-dependent relational encoding in a single token.

4.3 TEMPORAL NEIGHBOR AGGREGATION

Relational extrapolation has already injected target-dependent information into each historical token. The aggregation stage therefore focuses on contextualizing these tokens across the sampled history and compressing them into node representations. Concretely, we leverage a lightweight temporal neighbor aggregator inspired by gMLP (Liu et al., 2021) to mix information across historical positions, without relying on additional pair-statistic features or further query conditioning. Let $\mathbf{X}_v(t)$ denote the token sequence of endpoint $v \in \{s, d\}$ from equation 3. Because each token already incorporates target-dependent relational information, the same aggregation module is shared across endpoints. Each block first projects the normalized tokens into two branches, $[\mathbf{U}_v(t) \| \mathbf{V}_v(t)] = \mathrm{GELU}(\mathrm{LN}(\mathbf{X}_v(t))\mathbf{W}_{\mathrm{in}})$, where $\mathbf{W}_{\mathrm{in}}$ is the input projection. The second branch is then mixed across historical positions as $\widehat{\mathbf{V}}_v(t) = \mathbf{W}_{\mathrm{mix}}\,\mathrm{LN}(\mathbf{V}_v(t)) + \mathbf{b}_{\mathrm{mix}}\mathbf{1}^\top$, where $\mathbf{W}_{\mathrm{mix}}$ mixes information along the token dimension and $\mathbf{b}_{\mathrm{mix}}$ is broadcast across features. Feature-wise gating is followed by a residual update:

$$\mathbf{X}_v(t) = \mathbf{X}_v(t) + \Big(\mathbf{U}_v(t) \odot \widehat{\mathbf{V}}_v(t)\Big)\,\mathbf{W}_{\mathrm{out}}, \tag{4}$$

where $\mathbf{W}_{\mathrm{out}}$ projects the gated features back to the token dimension. Padding positions are masked throughout aggregation. We then obtain the node representations through attentive pooling:

$$\mathbf{z}_v(t) = \mathrm{AttnPool}\big(\mathbf{X}_v(t)\big), \tag{5}$$

where $\mathrm{AttnPool}(\cdot)$ denotes learnable attention-based pooling (Lin et al., 2017) over the valid historical tokens, excluding padded positions. The gMLP blocks contextualize each occurrence with the remaining token sequence, while attentive pooling retains the most informative evidence. For notational simplicity, we write $\mathbf{z}_s(t)$, although the source representation depends on target d through equation 2. Finally, $\mathbf{z}_s(t)$ and $\mathbf{z}_d(t)$ are inputted into a two-layer MLP to estimate the interaction likelihood, and the model is trained with binary cross-entropy against random negatives. All queries in a batch are scored using the memory states available at the beginning of the batch, and endpoint memories are updated only after all queries have been scored through equation 1.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING

Datasets. We collect 10 temporal graphs from (Poursafaei et al., 2022; Paranjape et al., 2017; Huang et al., 2023; Gastinger et al., 2024; Yi et al., 2025b): Wikipedia, Reddit, Flights, UCI, MOOC, THGL-Software, sx-askubuntu, sx-superuser, TGBL-Review and YouTube. They span two orders of magnitude in size (60K to 4.9M interactions), bipartite and unipartite graphs, and a repeat ratio from 0% to 88%; Table 4 lists their statistics and Appendix C.2 presents more details.

Baselines. We compare StatE against 11 baselines, covering several modeling paradigms: (1) query-agnostic modeling (i.e., JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGN (Rossi et al., 2020), TGAT (Xu et al., 2020), GraphMixer (Cong et al., 2023), L-STEP (Tieu et al., 2025), and CTDG-SSM (Raghuvanshi et al., 2026)), (2) query-aware modeling (i.e., CAWN (Wang et al., 2021c), DyGFormer (Yu et al., 2023), SALoM (Liu et al., 2025), and CRAFT (Yi et al., 2025a)), and (3) memory-based modeling (i.e., TGN (Rossi et al., 2020), SALoM (Liu et al., 2025)).

Table 1: MRR scores (%) of 12 compared methods on 10 temporal graphs. The **best** and second-best results are highlighted. "-" denotes TLE/OOM in our environment.

Method	Wikipedia	Reddit	Flights	UCI	MOOC	Software	Askubuntu	Superuser	Review	YouTube
JODIE	76.48±1.72	77.16±1.27	20.37±4.18	51.43±1.74	19.91±3.06	-	68.85±1.20	66.08±0.97	41.43±0.15	41.67±2.86
DyRep	67.42±4.21	72.33±2.14	15.62±0.47	14.00±2.71	17.71±1.27	-	38.14±1.64	35.99±1.22	40.06±0.59	35.12±1.17
TGAT	72.72±1.50	78.03±0.25	53.90±1.07	31.50±0.48	31.55±4.14	48.33±2.01	51.72±0.62	45.62±1.77	19.64±0.23	43.56±2.53
TGN	82.22±0.39	79.25±0.24	64.24±2.11	45.29±6.43	39.03±4.53	-	54.47±1.77	50.36±0.98	37.48±0.23	55.16±5.89
GraphMixer	73.57±1.48	70.87±0.38	42.26±0.49	59.58±1.00	29.02±0.28	55.75±1.01	74.84±0.69	72.82±0.57	36.89±1.50	58.87±0.12
L-STEP	84.42±0.52	82.06±0.48	-	58.62±1.99	26.15±2.56	-	74.58±2.10	70.74±2.28	-	-
CTDG-SSM†	87.61±0.75	87.76±0.93	64.81±3.68	61.01±0.82	15.56±2.04	50.18±1.51	46.18±1.36	41.76±1.19	21.95±2.27	33.59±1.55
CAWN	86.07±0.08	87.66±0.04	80.06±2.10	66.96±0.48	27.02±0.57	58.87±1.71	61.30±1.06	58.71±2.34	19.30±0.10	47.55±1.08
DyGFormer	88.69±0.04	88.70±0.05	**84.13**±3.46	76.61±0.26	42.20±1.31	64.63±0.39	56.38±0.44	55.33±0.22	22.39±1.52	46.08±3.44
CRAFT	88.25±0.26	**89.33**±0.11	77.15±0.07	75.11±0.09	62.32±0.39	55.70±0.48	67.46±0.40	64.02±0.37	41.77±0.06	58.92±0.16
SALoM	88.51±0.04	88.56±0.10	-	74.71±0.13	37.76±0.21	-	37.08±2.48	46.28±1.71	-	-
StatE	**90.01**±0.07	88.63±0.06	83.16±0.04	**79.89**±0.03	**66.06**±1.37	**72.47**±0.22	**76.55**±0.96	**75.12**±0.28	**49.26**±0.54	**65.37**±0.68
Improve(%)	+1.49%	-0.78%	-1.15%	+4.28%	+6.00%	+12.13%	+2.28%	+3.16%	+17.93%	+10.95%

Evaluation protocol. Following baselines (Liu et al., 2025; Yi et al., 2025a), we evaluate temporal link prediction under ranking settings, where we use the Mean Reciprocal Rank (MRR) as the evaluation metric; we sample 100 negative instances per positive one. We report the mean and standard deviation over three runs. **Additional implementation details are provided in Appendix C.1, with further experiments and analyses in Appendix C.**

5.2 Overall Effectiveness

To evaluate the overall effectiveness of StatE, we compare it with 11 representative baselines on ten temporal graphs using MRR. Table 1 reports the MRR results.

Finding 1: Strong overall performance across diverse temporal graphs. StatE consistently outperforms the strongest query-agnostic baseline on all ten temporal graphs, with an average relative MRR improvement of 19.76%, demonstrating the importance of query-aware relation modeling for temporal link prediction. Among query-aware methods, StatE achieves the best MRR on eight of the ten datasets, with notable gains of 12.13%, 17.93%, and 10.95% on THGL-Software, TGBL-Review, and YouTube, respectively. Interestingly, these larger gains occur on datasets with relatively weak repeat support: their repeat ratios are 40.90%, 2.94%, and 0%, respectively. In contrast, the only two datasets where StatE slightly trails the strongest query-aware baseline are the highly repetitive Reddit (88.32% repeat, −0.78%) and Flights (79.50% repeat, −1.15%). This contrast is consistent with our motivating hypothesis: explicit pair statistics are particularly effective when historical support is abundant, whereas relational extrapolation becomes increasingly useful as such support weakens. We examine this relationship directly at the query level in Section 5.3.

5.3 Performance Beyond Pair-Statistic Support

We next examine whether the dataset-level pattern in Table 1 persists at the query level. We partition transductive test queries according to the pair-statistic support available before prediction: *direct*, where the queried pair has appeared previously; *co-only*, where direct recurrence is absent but co-occurrence support is available; and *none*, where neither recurrence nor co-occurrence support exists. This decomposition further separates first-seen queries that still benefit from shared-neighbor evidence from those without either form of predefined support, which may exclude some test queries. Fig. 3 reports the results of eight methods on two datasets in MRR.

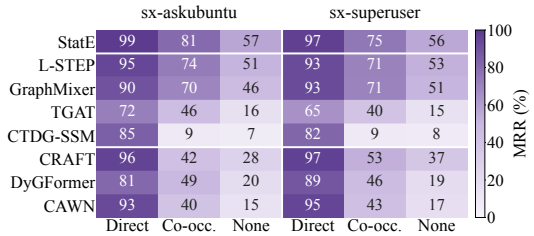


Figure 3: MRR across direct, co-occurrence-only, and neither-support regimes on two datasets.

Finding 2: StatE's advantage increases as pair-statistic support weakens. Across sx-askubuntu and sx-superuser, StatE achieves the best or tied-best performance in all three regimes.

Its average relative improvement over the strongest baseline increases from only 1.56% under direct support to 7.55% under co-occurrence-only support and 8.71% when neither statistic is available. This widening margin reveals where the benefit of relational extrapolation arises. When exact-pair recurrence is available, existing methods can already exploit a strong predefined matching signal, leaving relatively little room for improvement. Once recurrence is absent, and especially when both recurrence and co-occurrence are unavailable, the advantage of STATE becomes substantially larger. These results directly support our central claim that the learned witness–target relation remains useful beyond predefined pair-statistic support, rather than deriving its gains primarily from historically supported patterns.

5.4 ROBUSTNESS BEYOND RANDOM NEGATIVE SAMPLING

The default random-negative protocol may favor pair-statistic signals, since randomly sampled negative candidates often lack the historical support available to positive interactions. We therefore evaluate the same trained checkpoints, without retraining, under two harder protocols: *historical* negatives that have interacted with the source before the query, and *inductive* negatives corresponding to source–target pairs that are unseen during training but appear in the test. We report the representative result on sx-askubuntu in Fig. 4; results on three other datasets are provided in Appendix C.3.

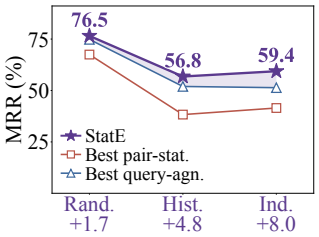


Figure 4: MRR (%) on sx-askubuntu under 3 samplers. Numbers denote the absolute improvement of STATE.

Finding 3: The gain of STATE is not specific to random-negative evaluation. On sx-askubuntu, the relative improvement of STATE over the strongest baseline increases from 2.28% under random negatives to 9.31% under historical negatives and 15.58% under inductive negatives. The substantially larger margins under both harder protocols show that the advantage of STATE does not rely on easily distinguishable random candidates. Instead, relational extrapolation remains effective when targets have stronger historical plausibility or when the queried source–target relation has not been observed during training. This provides complementary evidence that the gains observed in Section 5.3 reflect relational generalization rather than an artifact of the standard negative-sampling protocol.

5.5 ABLATION STUDY

To investigate the contribution of each component in STATE, we consider four ablations: replacing the unobserved relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$ with the observed relation $\psi(\mathbf{m}_w, \mathbf{m}_s)$ in equation 2 (*w. WS*); removing the relational token entirely (*w/o relation*); replacing the stable node memory with the last-step node representation (*w/o memory*); and replacing the gMLP spatial mixer with an MLP of equal depth (*w. MLP*). We report the results in Table 2.

Finding 4: Our relational extrapolation is the primary source of improvement. Removing the relational token (w/o relation) causes the largest performance degradation, reducing MRR by 25.12% on average across 4 datasets. This substantial drop confirms that our target-dependent relational encoding is the primary source of STATE's performance gains. Together, these results show that our relational extrapolation encoding provides useful evidence across directly supported, memory-mediated, and unsupported relations, and drives the main performance gain of STATE. Importantly, because target dependence is injected before pooling through this relation, removing it collapses the historical evidence toward target-agnostic encoding that downstream aggregation cannot recover.

Table 2: Ablation study (MRR) of STATE on 4 graphs.

Variant	Wikipedia	UCI	MOOC	YouTube
StatE	**90.01**$\pm$**0.07**	**79.89**$\pm$**0.03**	**66.06**$\pm$**1.37**	**65.37**$\pm$**0.68**
w/o relation	77.08$\pm$0.23	57.68$\pm$0.59	35.00$\pm$0.00	57.99$\pm$0.33
w. WS	86.75$\pm$0.01	77.31$\pm$0.29	52.37$\pm$1.23	61.29$\pm$0.78
w/o memory	89.01$\pm$0.07	78.42$\pm$0.14	60.97$\pm$0.11	62.80$\pm$0.24
w. MLP	88.35$\pm$0.01	77.55$\pm$0.17	55.54$\pm$0.15	62.59$\pm$0.20

Finding 5: The gain of STATE does not come from simply adding a learnable vector. Replacing the witness–target relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$ with the witness–source relation $\psi(\mathbf{m}_w, \mathbf{m}_s)$ (w. WS)

reduces MRR by 8.45% on average, despite retaining the same learnable relation module. This is consistent with Fig. 7 (Appendix), where $\psi(\mathbf{m}_w, \mathbf{m}_s)$ remains near chance because it is invariant to the targets, whereas $\psi(\mathbf{m}_w, \mathbf{m}_d)$ provides clear target-dependent discrimination. Thus, target conditioning provides additional gains beyond merely introducing an additional learnable relational representation.

Finding 6: Stable memory and temporal mixing both contribute to relational encoding. Replacing the stable memory with the latest node representation (w/o memory) reduces MRR by 3.65% on average, showing that smoothed temporal states provide more reliable inputs for witness–target comparison. Replacing the gMLP mixer with an MLP (w. MLP) reduces MRR by 6.24% on average, indicating that contextualizing relational evidence across historical tokens further improves the final representation.

5.6 Learned Relations versus Explicit Pair Statistics

We further examine whether our relation encoding merely recovers explicit pair statistics. We conduct four variants: *Stats only* replaces the learned relation with explicit recurrence (Yi et al., 2025a) and co-occurrence statistics (Yu et al., 2023); *StatE + Stats* adds both statistics to StatE; and *StatE + Rep* and *StatE + Co-occ* add them individually. Results on three datasets are shown in Table 3.

Finding 7: A simple learned relation captures substantial information beyond explicit pair statistics. Explicit statistics are strong query-aware signals: relative to *w/o relation*, *Stats only* improves MRR by 39.4% on average across the three datasets. Nevertheless, StatE without explicit statistics achieves a further 4.7% average improvement over *Stats only*. This benefit may stem from relational extrapolation when neither recurrence nor co-occurrence support is available.

Table 3: Learned relation versus pair statistics on 3 datasets (MRR, %), omitting standard deviations.

Variant	Learned	Explicit	UCI	MOOC	YouTube
w/o relation	✗	✗	57.68	35.00	57.99
Stats only	✗	✓	80.42	59.71	62.80
StatE	✓	✗	79.89	66.06	65.37
StatE + Stats	✓	✓	80.47	69.03	68.35
StatE + Rep	✓	rep	80.44	64.67	65.31
StatE + Co-occ	✓	co	79.92	68.44	68.38

Finding 8: Recurrence is largely subsumed, while co-occurrence retains complementarity. Adding explicit recurrence to StatE changes MRR by only -0.5% on average, suggesting that its predictive value is already largely absorbed by our learned witness–target relation. In contrast, explicit co-occurrence yields a 2.7% average improvement, indicating that co-neighbor structure retains complementary information beyond what is captured implicitly by temporal memories. This asymmetry further supports the unified view in Proposition 1: the learned relation need not reconstruct each pair statistic explicitly, but can subsume their structural roles to different degrees through a common witness–target matching mechanism. Despite its simplicity, the same relation accommodates historically supported evidence while remaining applicable beyond predefined pattern support.

6 Conclusion

In this paper, we revisit query-aware link prediction on temporal graphs beyond pair statistics. We identify *Pair-Statistic Support Bias*, highlighting that strong overall performance can partly reflect the exploitation of predefined patterns rather than relational generalization beyond their support. To address this limitation, we propose StatE, a simple yet effective approach based on relational extrapolation for temporal graph learning. By learning witness–target relations from stable temporal node states before aggregation, StatE infers potential yet unobserved relations while unifying directly supported recurrence, memory-mediated co-occurrence, and extrapolative evidence beyond predefined pair-pattern support. Extensive experiments demonstrate the overall effectiveness of StatE. Further analyses show improved first-seen prediction while maintaining strong repeated-link performance, and validate relational extrapolation beyond predefined support. Overall, our theoretical and empirical analyses show that a simple witness–target relation can unify the roles of predefined matching rules while remaining effective both within and beyond pair-statistic support.

AI USE STATEMENT

We use generative AI tools solely to refine the writing of this manuscript, including improving grammar and presentation. We review AI-assisted edits to ensure that they accurately reflect our technical content and claims. We take full responsibility for the final content of this work, including any text produced or revised with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide implementation details in the Appendix and plan to release the implementation code upon acceptance.

REFERENCES

Qijie Bai, Changli Nie, Haiwei Zhang, Dongming Zhao, and Xiaojie Yuan. Hgwavenet: A hyperbolic graph neural network for temporal link prediction. In *The ACM Web Conference*, pp. 523–532. ACM, 2023.

Fabrizio De Castelli, Alessio Gravina, Moshe Eliasof, Carola-Bibiane Schönlieb, and Davide Bacciu. Adaptive memory retention in dynamic graphs. In *International Conference on Machine Learning*, 2026.

Ke Cheng, Linzhi Peng, Pengyang Wang, Heng Chang, Junchen Ye, and Bowen Du. On the scalability of temporal relative positional encoding for dynamic link prediction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 298–309, 2025.

Jacob Chmura, Shenyang Andy Huang, Tran Gia Bao Ngo, Ali Parviz, Farimah Poursafaei, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, Matthias Fey, and Reihaneh Rabbany. Tgm: a modular and efficient library for machine learning on temporal graphs. In *International Conference on Learning Representations*, 2026.

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? In *The International Conference on Learning Representations*, 2023.

Or Feldman, Krishna Sri Ipsit Mantri, Moshe Eliasof, and Chaim Baskin. Revisting node affinity prediction in temporal graphs. In *International Conference on Learning Representations*, volume 2026, pp. 104203–104222, 2026.

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

Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, et al. Tgb 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. *Advances in neural information processing systems*, 37:140199–140229, 2024.

Franziska Heeg and Ingo Scholtes. Using time-aware graph neural networks to predict temporal centralities in dynamic graphs. *Advances in Neural Information Processing Systems*, 37:30149–30178, 2024.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. *Advances in Neural Information Processing Systems*, 36:2056–2073, 2023.

Ming Jin, Yuan-Fang Li, and Shirui Pan. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs. *Advances in Neural Information Processing Systems*, 35:19874–19886, 2022.

Yannis Karmim, Marc Lafon, Raphaël F S'niehotta, and Nicolas Thome. Supra-laplacian encoding for transformer on dynamic graphs. *Advances in Neural Information Processing Systems*, 37: 17215–17246, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1269–1278. ACM, 2019.

Veronica Lachi, Francesco Ferrini, Antonio Longa, Bruno Lepri, Andrea Passerini, and Manfred Jaeger. Bridging theory and practice in link representation with graph neural networks. In *Annual Conference on Neural Information Processing Systems*, 2025.

Zhao-Rong Lai, Zheng-Sen Zhou, Liangda Fang, Yongsen Zheng, and Ziliang Chen. ℓ_1 latent distance based continuous-time graph representation. In *International Conference on Learning Representations*, 2026.

Hao Li, Hao Wan, Yuzhou Chen, Dongsheng Ye, Yulia Gel, and Hao Jiang. Tmetanet: Topological meta-learning framework for dynamic link prediction. In *International Conference on Machine Learning*, pp. 35977–35999, 2025.

Jintang Li, Zhouxin Yu, Zulun Zhu, Liang Chen, Qi Yu, Zibin Zheng, Sheng Tian, Ruofan Wu, and Changhua Meng. Scaling up dynamic graph representation learning via spiking neural networks. In *The AAAI Conference on Artificial Intelligence*, pp. 8588–8596. AAAI Press, 2023.

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

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING. In *International Conference on Learning Representations*, 2017.

Hanwen Liu, Longjiao Zhang, Rui Wang, Tongya Zheng, Sai Wu, Chang Yao, and Mingli Song. Salom: Structure aware temporal graph networks with long-short memory updater. In *Annual Conference on Neural Information Processing Systems*, 2025.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215, 2021.

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

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

Youda Mo, Chaobo He, Junwei Cheng, Peng Mei, and Quanlong Guan. Entangled no more: Multi-domain decoupling for robust dynamic graph neural networks. In *International Conference on Machine Learning*, 2026.

Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *ACM international conference on web search and data mining*, pp. 601–610, 2017.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *The AAAI Conference on Artificial Intelligence*, pp. 5363–5370. AAAI Press, 2020.

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

Ayushman Raghuvanshi, Thummaluru Siddartha Reddy, Sundeep Prabhakar Chepuri, and Mahesh Chandran. Learning long range spatio-temporal representations over continuous time dynamic graphs with state space models. In *International Conference on Machine Learning*, 2026.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML Workshop on Graph Representation Learning*, 2020.

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

Kiarash Shamsi, Tran Gia Bao Ngo, Razieh Shirzadkhani, Shenyang Huang, Farimah Poursafaei, Poupak Azad, Reihaneh Rabbany, Baris Coskunuzer, Guillaume Rabusseau, and Cuneyt Gurcan Akcora. MiNT: Multi-network transfer benchmark for temporal graph learning. In *Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

Weining Shi, Zhisen Wen, Qinggang Zhang, Chentao Zhang, and Zhihong Zhang. Temporal graph thumbnail: Robust representation learning with global evolutionary skeleton. In *International Conference on Learning Representations*, 2026.

Aimin Sun and Zhiguo Gong. Structure-aware model for representation learning on temporal graphs. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Yadong Sun, Xiaofeng Cao, Ivor Tsang, and Heng Tao Shen. Analytical construction on geometric architectures: Transitioning from static to temporal link prediction. In *International Conference on Machine Learning*, 2025.

Yuxing Tian, Yiyan Qi, and Fan Guo. Freedyg: Frequency enhanced continuous-time dynamic graph model for link prediction. In *The International Conference on Learning Representations*, 2024.

Yuxing Tian, Yiyan Qi, Fengran Mo, Weixu Zhang, Jian Guo, and Jian-Yun Nie. Learning discriminative and generalizable anomaly detector for dynamic graph with limited supervision. In *International Conference on Machine Learning*, 2026.

Katherine Tieu, Dongqi Fu, Zihao Li, Ross Maciejewski, and Jingrui He. Learnable spatial-temporal positional encoding for link prediction. In *International Conference on Machine Learning*, pp. 59570–59597. PMLR, 2025.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *The International Conference on Learning Representations*. OpenReview.net, 2019.

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

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

Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, and Zhenyu Guo. APAN: asynchronous propagation attention network for real-time temporal graph embedding. In *The International Conference on Management of Data*, pp. 2628–2638. ACM, 2021b.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *The International Conference on Learning Representations*. OpenReview.net, 2021c.

Yili Wang, Tairan Huang, Changlong He, Qiutong Li, and Jianliang Gao. Simple and efficient heterogeneous temporal graph neural network. In *Annual Conference on Neural Information Processing Systems*, 2025a.

Yunan Wang, Jianxin Li, and Ziwei Zhang. Global-recent semantic reasoning on dynamic text-attributed graphs with large language models. In *International Conference on Learning Representations*, 2026.

Zhe Wang, Sheng Zhou, Jiawei Chen, Zhen Zhang, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. Dynamic graph transformer with correlated spatial-temporal positional encoding. In *ACM International Conference on Web Search and Data Mining*, pp. 60–69, 2025b.

Yuxia Wu, Yuan Fang, and Lizi Liao. On the feasibility of simple transformer for dynamic graph modeling. In *ACM web conference*, pp. 870–880, 2024.

Yuxia Wu, Lizi Liao, and Yuan Fang. Retrieval augmented generation for dynamic graph modeling. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1434–1443, 2025.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *The International Conference on Learning Representations*, 2020.

Yuanyuan Xu, Wenjie Zhang, Xiwei Xu, Binghao Li, and Ying Zhang. Scalable and effective temporal graph representation learning with hyperbolic geometry. *IEEE Transactions on Neural Networks and Learning Systems*, 2024a.

Yuanyuan Xu, Wenjie Zhang, Ying Zhang, Maria Orlowska, and Xuemin Lin. Timesgn: Scalable and effective temporal graph neural network. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 3297–3310. IEEE, 2024b.

Jiayi Yang, Xing Wei, Chunchun Chen, Yi Feng, Wengang Guo, Rui Fan, Xiaofeng Cao, Xin Sun, and Wei Ye. Fast mixture of curvature-aware experts for diverse and dynamic graph topologies. In *International Conference on Machine Learning*, 2026.

Lu Yi, Runlin Lei, Fengran Mo, Yanping Zheng, Zhewei Wei, and Yuhang Ye. Future link prediction without memory or aggregation. In *Annual Conference on Neural Information Processing Systems*, 2025a.

Lu Yi, Jie Peng, Yanping Zheng, Fengran Mo, Zhewei Wei, Yuhang Ye, Yue Zixuan, and Zengfeng Huang. Tgb-seq benchmark: Challenging temporal gnns with complex sequential dynamics. In *The International Conference on Learning Representations*, 2025b.

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.

Zhongyi Yu, Jianqiu Wu, Zhenghao Wu, Shuhan Zhong, Weifeng Su, Chul-Ho Lee, and Weipeng Zhuo. Tami: Taming heterogeneity in temporal interactions for temporal graph link prediction. In *Annual Conference on Neural Information Processing Systems*, 2025.

Siwei Zhang, Yun Xiong, Yateng Tang, Jiarong Xu, Xi Chen, Zehao Gu, Xuehao Zheng, Zi'an Jia, and Jiawei Zhang. Unifying text semantics and graph structures for temporal text-attributed graphs with large language models. In *Annual Conference on Neural Information Processing Systems*, 2025a.

Xin Zhang, Fei Cai, Jianming Zheng, Zhiqiang Pan, Wanyu Chen, Honghui Chen, and Chonghao Chen. Triangle matters! topdyg: Topology-aware transformer for link prediction on dynamic graphs. In *ACM on Web Conference*, pp. 3607–3617, 2025b.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. *Advances in Neural Information Processing Systems*, 35:6074–6089, 2022.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Zhou Qin, Weigao Wen, Hui Xue, Haoyang Li, and Wenwu Zhu. Spectral invariant learning for dynamic graphs under distribution shifts. *Advances in Neural Information Processing Systems*, 36:6619–6633, 2023.

Kesen Zhao and Liang Zhang. Causality-inspired spatial-temporal explanations for dynamic graph neural networks. In *The International Conference on Learning Representations*, 2024.

Yumeng Zhou, Mingzhe Liu, Leilei Sun, Yifei Huang, Liangzhe Han, Chuanren Liu, and Tongyu Zhu. Position-aware neighbor aggregation for dynamic link prediction. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 4156–4167, 2025.

Tao Zou, Yuhao Mao, Junchen Ye, and Bowen Du. Repeat-aware neighbor sampling for dynamic graph learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4722–4733, 2024.

Tao Zou, Chengfeng Wu, Tianxi Liao, Junchen Ye, and Bowen Du. Global-lens transformers: Adaptive token mixing for dynamic link prediction. In *AAAI Conference on Artificial Intelligence*, volume 40, pp. 16575–16583, 2026.

Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2857–2866, 2018.

A Additional Motivating Analysis

UCI and sx-askubuntu exhibit substantially different interaction patterns: UCI is repeat-dominant with a repeat ratio of 66.08%, whereas sx-askubuntu is much less repetitive with a repeat ratio of 38.11%. This contrast allows us to examine whether the effect of pair statistics persists under weaker historical pair support. As shown in Fig. 5, we observe the same qualitative pattern on sx-askubuntu. Removing the corresponding pair statistic causes much larger relative degradation on repeated queries: MRR decreases by 11.08% for DyGFormer, 51.75% for CRAFT, and 72.31% for SALoM. In contrast, first-seen MRR increases by 3.85% and 6.20% for DyGFormer and CRAFT, respectively, while SALoM decreases by 51.35%. This indicates that the benefit of explicit pair statistics remains concentrated on queries where the corresponding historical support is present, even on a substantially less repetitive graph.

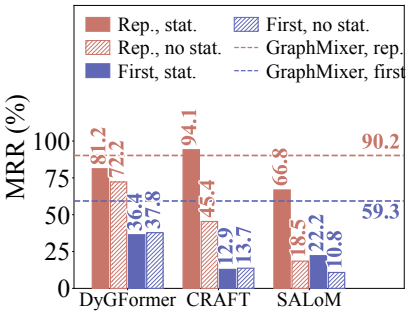


Figure 5: Effect of removing pair statistics on repeated and first-seen queries on sx-askubuntu. Dashed lines denote the statistic-free GraphMixer.

The statistic-free GraphMixer further highlights this imbalance, achieving strong first-seen performance while remaining competitive on repeated queries. Together with the UCI results in Fig. 1, these results provide further evidence of *Pair-Statistic Support Bias*: aggregate performance can be driven by gains on historically supported queries while masking weaker generalization beyond pair-statistic support.

B Theoretical Analysis

B.1 Theoretical Analysis for Stable Memory

Theorem 1 (Tracking-error decomposition)**.** *Fix a node v, and let $t_0, \dots, t_k < t$ be consecutive update events of v strictly before the query time t, ordered by timestamp with deterministic tie-breaking for identical timestamps. For each $i = 1, \dots, k$, define $\Delta t_i := t_i - t_{i-1}$, $\rho_i := \exp(-\lambda \Delta t_i/3600)$, and $\kappa_i := \beta + (1-\beta)\rho_i$, where timestamps are measured in seconds. Suppose the raw node embedding admits the decomposition $\mathbf{z}_v(t_i) = \mathbf{h}_v(t_i) + \boldsymbol{\xi}_v(t_i)$, $i = 0, 1, \dots, k$, where $\mathbf{h}_v(t_i)$ is a latent structural state and $\boldsymbol{\xi}_v(t_i)$ is an event-level residual. Then, for every $i = 1, \dots, k$,*

$$\|\mathbf{m}_v(t_i) - \mathbf{h}_v(t_i)\| \le \kappa_i \|\mathbf{m}_v(t_{i-1}) - \mathbf{h}_v(t_{i-1})\| + \kappa_i \|\mathbf{h}_v(t_i) - \mathbf{h}_v(t_{i-1})\| + (1-\kappa_i)\|\xi_v(t_i)\|. \quad (6)$$

Proof. Fix $i \in \{1, \dots, k\}$. For notational brevity, define

$$\mathbf{m}_i := \mathbf{m}_v(t_i),\ \mathbf{h}_i := \mathbf{h}_v(t_i),\ \boldsymbol{\xi}_i := \boldsymbol{\xi}_v(t_i),\ \mathbf{z}_i := \mathbf{z}_v(t_i),\ \tilde{\mathbf{z}}_i := \tilde{\mathbf{z}}_v(t_i). \quad (7)$$

Since the i-th update instance immediately follows the $(i-1)$-th update instance under the timestamp order with deterministic tie-breaking, the pre-update memory at the i-th update equals the post-update memory after the previous update:

$$\mathbf{m}_v(t_i^-) = \mathbf{m}_v(t_{i-1}) = \mathbf{m}_{i-1}. \quad (8)$$

Applying equation 1 at time t_i, we obtain

$$\mathbf{m}_i = \rho_i \mathbf{m}_{i-1} + (1-\rho_i)\tilde{\mathbf{z}}_i,\ \tilde{\mathbf{z}}_i = (1-\beta)\mathbf{z}_i + \beta \mathbf{m}_{i-1}. \quad (9)$$

Substituting the second identity into the first yields

$$\begin{aligned} \mathbf{m}_i &= \rho_i \mathbf{m}_{i-1} + (1-\rho_i)\big((1-\beta)\mathbf{z}_i + \beta \mathbf{m}_{i-1}\big) \\ &= \big(\rho_i + \beta(1-\rho_i)\big)\mathbf{m}_{i-1} + (1-\beta)(1-\rho_i)\mathbf{z}_i \\ &= \kappa_i \mathbf{m}_{i-1} + (1-\kappa_i)\mathbf{z}_i, \end{aligned} \quad (10)$$

where the last equality follows from the definition of κ_i, since $1 - \kappa_i = 1 - \beta - (1-\beta)\rho_i = (1-\beta)(1-\rho_i)$.

Using the decomposition in Theorem 1, $\mathbf{z}_i = \mathbf{h}_i + \boldsymbol{\xi}_i$, we obtain

$$\begin{aligned}\mathbf{m}_i - \mathbf{h}_i &= \kappa_i \mathbf{m}_{i-1} + (1-\kappa_i)\mathbf{z}_i - \mathbf{h}_i \\ &= \kappa_i \mathbf{m}_{i-1} + (1-\kappa_i)(\mathbf{h}_i + \boldsymbol{\xi}_i) - \mathbf{h}_i \\ &= \kappa_i \mathbf{m}_{i-1} - \kappa_i \mathbf{h}_i + (1-\kappa_i)\boldsymbol{\xi}_i.\end{aligned} \tag{11}$$

Adding and subtracting $\kappa_i \mathbf{h}_{i-1}$ gives

$$\begin{aligned}\mathbf{m}_i - \mathbf{h}_i &= \kappa_i \mathbf{m}_{i-1} - \kappa_i \mathbf{h}_{i-1} + \kappa_i \mathbf{h}_{i-1} - \kappa_i \mathbf{h}_i + (1-\kappa_i)\boldsymbol{\xi}_i \\ &= \kappa_i(\mathbf{m}_{i-1} - \mathbf{h}_{i-1}) + \kappa_i(\mathbf{h}_{i-1} - \mathbf{h}_i) + (1-\kappa_i)\boldsymbol{\xi}_i.\end{aligned} \tag{12}$$

Because $\lambda > 0$ and $\Delta t_i \geq 0$, we have $\rho_i \in (0, 1]$. Together with $\beta \in [0, 1)$, this implies $\kappa_i \in [\beta, 1]$, so both κ_i and $1 - \kappa_i$ are nonnegative. Therefore, taking norms and applying the triangle inequality yields

$$\begin{aligned}\|\mathbf{m}_i - \mathbf{h}_i\| &\leq \kappa_i \|\mathbf{m}_{i-1} - \mathbf{h}_{i-1}\| + \kappa_i \|\mathbf{h}_{i-1} - \mathbf{h}_i\| + (1-\kappa_i)\|\boldsymbol{\xi}_i\| \\ &= \kappa_i \|\mathbf{m}_{i-1} - \mathbf{h}_{i-1}\| + \kappa_i \|\mathbf{h}_i - \mathbf{h}_{i-1}\| + (1-\kappa_i)\|\boldsymbol{\xi}_i\|.\end{aligned} \tag{13}$$

Rewriting this inequality in the original notation, we obtain

$$\|\mathbf{m}_v(t_i) - \mathbf{h}_v(t_i)\| \leq \kappa_i \|\mathbf{m}_v(t_{i-1}) - \mathbf{h}_v(t_{i-1})\| + \kappa_i \|\mathbf{h}_v(t_i) - \mathbf{h}_v(t_{i-1})\| + (1-\kappa_i)\|\boldsymbol{\xi}_v(t_i)\|, \tag{14}$$

which is exactly equation 6. Since $i \in \{1, \ldots, k\}$ was arbitrary, the claim holds for every $i = 1, \ldots, k$. This completes the proof. □

Takeaway. Theorem 1 shows that the memory has controlled tracking error under slowly varying structural states and bounded event-level perturbations. Thus, stable memory suppresses event-level noise while preserving evolving structural information.

Remark 3 (Bounded tracking under slowly varying structure)**.** *The recursive bound in Theorem 1 can be unrolled to make the tracking interpretation explicit. Let* $\alpha_i := \|\mathbf{m}_v(t_i) - \mathbf{h}_v(t_i)\|, \mu_i := \|\mathbf{h}_v(t_i) - \mathbf{h}_v(t_{i-1})\|, \nu_i := \|\boldsymbol{\xi}_v(t_i)\|$. *Then equation 6 gives*

$$\alpha_i \leq \kappa_i \alpha_{i-1} + \kappa_i \mu_i + (1-\kappa_i)\nu_i. \tag{15}$$

Suppose that the latent structural drift and the event-level residual are bounded as $\mu_i \leq \delta, \nu_i \leq \epsilon$, *and that the updates are uniformly contractive, i.e.,* $\kappa_i \leq \bar{\kappa} < 1, \forall i$, *where* δ *measures the maximum one-step change of the latent structural state and* ϵ *upper-bounds event-level residual perturbations. By recursively applying the above inequality, we obtain*

$$\alpha_i \leq \Big(\prod_{j=1}^{i} \kappa_j\Big)\alpha_0 + \sum_{\ell=1}^{i} \Big(\prod_{j=\ell+1}^{i} \kappa_j\Big)\kappa_\ell \mu_\ell + \sum_{\ell=1}^{i} \Big(\prod_{j=\ell+1}^{i} \kappa_j\Big)(1-\kappa_\ell)\nu_\ell, \tag{16}$$

where an empty product is defined as 1. *Using* $\mu_\ell \leq \delta$ *and* $0 \leq \kappa_j \leq \bar{\kappa}$ *for all* j, *the structural-drift term satisfies*

$$\sum_{\ell=1}^{i} \Big(\prod_{j=\ell+1}^{i} \kappa_j\Big)\kappa_\ell \mu_\ell \leq \delta \sum_{\ell=1}^{i} \bar{\kappa}^{i-\ell+1} = \frac{\bar{\kappa}(1-\bar{\kappa}^i)}{1-\bar{\kappa}}\delta. \tag{17}$$

For the residual term, with the convention $\prod_{j=i+1}^{i} \kappa_j = 1$, *we have*

$$\Big(\prod_{j=\ell+1}^{i} \kappa_j\Big)(1-\kappa_\ell) = \prod_{j=\ell+1}^{i} \kappa_j - \prod_{j=\ell}^{i} \kappa_j. \tag{18}$$

Summing over $\ell = 1, \ldots, i$ *gives* $\sum_{\ell=1}^{i} \Big(\prod_{j=\ell+1}^{i} \kappa_j\Big)(1-\kappa_\ell) = 1 - \prod_{j=1}^{i} \kappa_j \leq 1$. *Therefore, since* $\nu_\ell \leq \epsilon$,

$$\sum_{\ell=1}^{i} \Big(\prod_{j=\ell+1}^{i} \kappa_j\Big)(1-\kappa_\ell)\nu_\ell \leq \epsilon. \tag{19}$$

Substituting the two bounds into equation 16 yields

$$\alpha_i \le \bar{\kappa}^i \alpha_0 + \frac{\bar{\kappa}(1-\bar{\kappa}^i)}{1-\bar{\kappa}}\delta + \epsilon. \tag{20}$$

Equivalently,

$$\|\mathbf{m}_v(t_i) - \mathbf{h}_v(t_i)\| \le \bar{\kappa}^i \|\mathbf{m}_v(t_0) - \mathbf{h}_v(t_0)\| + \frac{\bar{\kappa}(1-\bar{\kappa}^i)}{1-\bar{\kappa}}\delta + \epsilon. \tag{21}$$

This bound separates the tracking error into three components: a geometrically decaying initial error, an accumulated structural-drift term controlled by δ, and a bounded residual floor ϵ. Hence, when the latent structure evolves slowly and event-level residuals are bounded, the memory remains close to the latent structural state. The uniform contraction condition $\kappa_i \le \bar{\kappa} < 1$ is an additional sufficient condition for the unrolled bound; the per-step recursive bound in Theorem 1 itself holds without this extra assumption. This unrolled contraction bound further assumes either strictly positive inter-update gaps after tie-breaking or merged same-timestamp updates; otherwise, $\Delta t_i = 0$ gives $\rho_i = 1$ and hence $\kappa_i = 1$, violating uniform contraction.

Takeaway. When the latent structure evolves slowly and event-level residuals are bounded, the memory remains close to the latent structural state.

B.2 Pair Statistics as Fixed Relational Matching Rules

Let $\mathcal{H}_v(t)$ denote the one-hop historical multiset of node v before time t. Define the occurrence count $c_v(w;t) = \sum_{(u,t_u)\in\mathcal{H}_v(t)} \mathbf{1}\{u = w\}$ and the support indicator $\bar{c}_v(w;t) = \mathbf{1}\{c_v(w;t) > 0\}$. When $\mathcal{H}_v(t)$ is sampled, these quantities are defined with respect to the sampled history. For a source-side witness weight $a_s(w;t)$ and a scalar witness–target matching rule $\kappa(w,d;t)$, we can define

$$T_{a,\kappa}(s,d;t) = \sum_{w\in\mathcal{V}} a_s(w;t)\kappa(w,d;t). \tag{22}$$

Proposition 1 (Pair statistics as fixed relational matching rules)**.** *For any query $(s,d;t)$, the following identities hold.*

(i) Recurrence. *Let $a_s(w;t) = c_s(w;t)$ and $\kappa_{\mathrm{rep}}(w,d;t) = \mathbf{1}\{w = d\}$. Then*

$$T_{a,\kappa_{\mathrm{rep}}}(s,d;t) = c_s(d;t), \tag{23}$$

which is the number of occurrences of target d in the source history.

(ii) Frequency-weighted co-occurrence. *Let $a_s(w;t) = c_s(w;t)$ and $\kappa_{\mathrm{co}}(w,d;t) = c_d(w;t)$. Then*

$$T_{a,\kappa_{\mathrm{co}}}(s,d;t) = \sum_{w\in\mathcal{V}} c_s(w;t)c_d(w;t), \tag{24}$$

which is the frequency-weighted overlap between the two historical neighbor multisets.

(iii) Binary common-neighbor count. *Let $a_s(w;t) = \bar{c}_s(w;t)$ and $\kappa_{\mathrm{cn}}(w,d;t) = \bar{c}_d(w;t)$. Then*

$$T_{a,\kappa_{\mathrm{cn}}}(s,d;t) = \left|\operatorname{supp} c_s(\cdot;t) \cap \operatorname{supp} c_d(\cdot;t)\right|. \tag{25}$$

Proof. We prove the three cases separately.

(i) Recurrence. Substituting $a_s(w;t) = c_s(w;t)$ and $\kappa_{\mathrm{rep}}(w,d;t) = \mathbf{1}\{w = d\}$ into equation 22 gives

$$T_{a,\kappa_{\mathrm{rep}}}(s,d;t) = \sum_{w\in\mathcal{V}} c_s(w;t)\mathbf{1}\{w = d\} = c_s(d;t). \tag{26}$$

The indicator removes all witnesses except $w = d$, so the resulting value is exactly the number of occurrences of target d in the source history.

(ii) Frequency-weighted co-occurrence. Substituting $a_s(w;t) = c_s(w;t)$ and $\kappa_{\mathrm{co}}(w,d;t) = c_d(w;t)$ gives

$$T_{a,\kappa_{\mathrm{co}}}(s,d;t) = \sum_{w\in\mathcal{V}} c_s(w;t)c_d(w;t). \tag{27}$$

Using the definitions of the occurrence counts, this can be expanded as

$$\begin{aligned}\sum_{w\in\mathcal{V}} c_s(w;t)c_d(w;t) &= \sum_{w\in\mathcal{V}} \left(\sum_{(u,t_u)\in\mathcal{H}_s(t)} \mathbf{1}\{u=w\}\right)\left(\sum_{(v,t_v)\in\mathcal{H}_d(t)} \mathbf{1}\{v=w\}\right) \\ &= \sum_{(u,t_u)\in\mathcal{H}_s(t)}\ \sum_{(v,t_v)\in\mathcal{H}_d(t)} \mathbf{1}\{u=v\}.\end{aligned} \tag{28}$$

Thus, each pair of historical occurrences contributes one exactly when the two endpoints share the same witness. If a node w occurs $c_s(w;t)$ times in the source history and $c_d(w;t)$ times in the destination history, it contributes $c_s(w;t)c_d(w;t)$ in total. Therefore, equation 27 is precisely the frequency-weighted historical neighborhood overlap.

(iii) Binary common-neighbor count. By definition, $\bar{c}_v(w;t) = \mathbf{1}\{c_v(w;t) > 0\}$. Hence $\bar{c}_s(w;t)\bar{c}_d(w;t) = 1$ if and only if w appears at least once in both histories. Therefore, we obtain

$$\begin{aligned}T_{a,\kappa_{\mathrm{cn}}}(s,d;t) &= \sum_{w\in\mathcal{V}} \bar{c}_s(w;t)\bar{c}_d(w;t) \\ &= \sum_{w\in\mathcal{V}} \mathbf{1}\{c_s(w;t)>0,\, c_d(w;t)>0\} \\ &= \left|\operatorname{supp} c_s(\cdot;t) \cap \operatorname{supp} c_d(\cdot;t)\right|.\end{aligned} \tag{29}$$

This is exactly the binary common-neighbor count. □

Relation to STATE. Proposition 1 shows that recurrence and co-occurrence can be expressed as fixed witness–target matching rules before pooling. STATE instead uses the learned relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$. For recurrence, the connection is direct: when $w = d$, the relational operator evaluates $\psi(\mathbf{m}_d, \mathbf{m}_d)$. For co-occurrence, the connection is memory-mediated: prior interactions between w and d can shape their pre-query temporal states before $\psi(\mathbf{m}_w, \mathbf{m}_d)$ is evaluated. When neither pattern is present, the same relation remains well-defined for memory-based extrapolation. This link is structural and does not require $\psi(\mathbf{m}_w, \mathbf{m}_d)$ to reconstruct the discrete pair statistics.

C Additional Experiments

C.1 Additional Experimental Setup

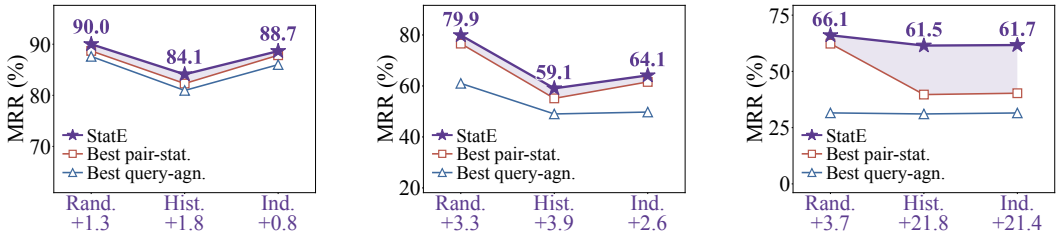
For the main experiments, STATE and all baselines are implemented in DyGLib (Yu et al., 2023) and evaluated under the same data splits, batch size, and evaluation protocol. For STATE, we set the number of sampled neighbors to 20, batch size to 256, number of layers to 2, and embedding dimension to 172. We apply collision checking in negative sampling to avoid information leakage. The learning rate is 10^{-4} and optimization uses Adam (Kingma & Ba, 2014). We set $\beta = 0.3$ in Eq. (1). All timestamps are in seconds, and we fix $\lambda = \mathrm{softplus}(-1) \approx 0.31$ for all datasets. Following standard practice (Huang et al., 2023), we chronologically split interactions into training, validation, and test sets. We use $H = 1$ for Wikipedia, Reddit, UCI, sx-superuser, and TGBL-Review, and $H = 2$ for sx-askubuntu, THGL-Software, MOOC, YouTube, and Flights. All experiments are conducted on Linux GPU servers with NVIDIA V100-SXM2-32GB GPUs.

C.2 Datasets

Table 4 summarizes the 10 temporal graphs used in our experiments, including their numbers of nodes, interactions, unique edges, timestamps, density, and graph type. The *repeat ratio* denotes the fraction of interactions whose $(\text{source}, \text{destination})$ pair has appeared earlier in the stream. We order the datasets by this ratio, with the upper block being repeat-dominant ($\geq 40\%$) and the lower block first-seen-dominant ($\leq 40\%$). The datasets span a broad range of interaction patterns, from highly repetitive graphs such as Wikipedia and Reddit to nearly entirely first-seen graphs such as TGBL-Review and YouTube.

Table 4: Statistics of the 10 datasets. Repeat ratio is the fraction of interactions whose (source, destination) pair already appeared earlier in the stream; Density is the number of unique edges over $N(N-1)/2$.

Dataset	Nodes (users/items)	Edges	Unique Edges	Timestamps	Repeat ratio (%)	Density	Bipartite
Wikipedia	9,227	157,474	18,257	152,757	88.41	4.29×10^{-4}	✓
Reddit	10,984	672,447	78,516	669,065	88.32	1.30×10^{-3}	✓
Flights	13,169	1,927,145	395,072	122	79.50	4.56×10^{-3}	×
UCI	1,899	59,835	20,296	58,911	66.08	1.13×10^{-2}	×
MOOC	7,144	411,749	178,443	345,600	56.66	6.99×10^{-3}	✓
THGL-Software	681,927	1,489,806	880,418	689,549	40.90	3.79×10^{-6}	×
sx-askubuntu	159,316	964,437	596,933	960,866	38.11	4.70×10^{-5}	×
sx-superuser	194,085	1,443,339	924,886	1,437,199	35.92	4.91×10^{-5}	×
TGBL-Review	352,637	4,873,540	4,730,223	6,865	2.94	7.61×10^{-5}	✓
YouTube	402,422	3,288,028	3,288,028	203	0.00	4.06×10^{-5}	×



(a) Wikipedia (b) UCI (c) MOOC

Figure 6: MRR under random, historical and inductive negatives on three datasets. Each panel compares StatE with the strongest pair-statistic method and the strongest query-agnostic method under each protocol; the shaded band and the number below each protocol give the margin of StatE over the best baseline. Numbers below each protocol denote the absolute MRR-point margin of StatE over the strongest baseline.

C.3 Additional Robustness under Different Negative Sampling

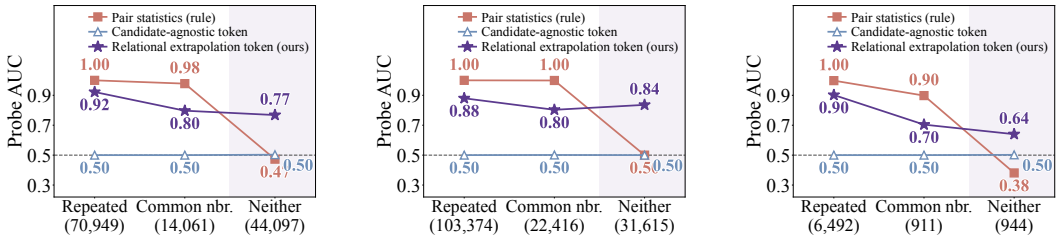
We provide additional evaluation under two harder negative-sampling protocols beyond the random negatives used in the main experiments. *Historical negatives* are destinations the source interacted with before the query time, whereas *inductive negatives* are destinations the source interacts with only in the test period, so the queried source–target pair is unseen during training. As in the main-text analysis, we evaluate the same trained checkpoints under all protocols without retraining; each hard protocol draws 100 negatives per query and the results are averaged over two independent draws. Fig. 6 reports the additional results on Wikipedia, UCI, and MOOC.

Finding 9: StatE remains robust under harder negative sampling across datasets. StatE achieves the highest MRR across all three protocols on Wikipedia, UCI, and MOOC. Although all methods degrade as negatives become harder, StatE consistently retains its advantage, showing that its effectiveness is not specific to uniformly sampled negatives.

Finding 10: StatE's advantage becomes more pronounced under harder negatives. Across the three datasets, the average relative improvement over the strongest baseline increases from 3.92% under random negatives to 21.39% under historical negatives and 19.41% under inductive negatives. The effect is particularly pronounced on MOOC, where the improvement grows from 6.00% to 54.86% and 53.13%, respectively. Historical negatives reduce the discriminative value of pair-specific history, while inductive negatives require ranking pairs unseen during training. These results further support the robustness of relational extrapolation beyond the standard random-negative protocol.

C.4 Additional Analysis of Relational Extrapolation

We further analyze whether the learned witness–target relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$ captures target-dependent evidence beyond predefined pair statistics. On sx-askubuntu, THGL-Software, and UCI,



(a) sx-askubuntu (b) THGL-Software (c) UCI

Figure 7: Probe AUC across repeated, common-neighbor, and neither-support regimes on three datasets.

test probe queries are grouped into *repeated*, *common neighbor*, and *neither* according to the positive target's relation to the sampled source witnesses. We compare three signals: (i) recurrence and co-occurrence statistics, (ii) the target-agnostic relation $\psi(\mathbf{m}_w, \mathbf{m}_s)$, and (iii) our target-dependent relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$. A logistic probe is fitted to the learned relations, and candidate targets are evaluated by AUC against random negatives, with 0.5 denoting chance.

Finding 11: Pair-statistic and target-agnostic signals are limited beyond their support. As shown in Fig. 7, on sx-askubuntu, pair statistics are highly predictive for repeated and common-neighbor queries, with AUCs of 1.00 and 0.98, but drop to 0.47 in the *neither* regime as their predefined structural support disappears. In contrast, the target-agnostic relation $\psi(\mathbf{m}_w, \mathbf{m}_s)$ remains at chance (0.50) across all three regimes. For fixed (s, t), the witness–source relation is invariant across candidate targets and therefore provides no target-dependent ranking signal. Moreover, the observed edge (s, w) already anchors w to the source through its edge, temporal, and hop features. This motivates modeling the unresolved witness–target relation in equation 2.

Finding 12: Relational extrapolation remains informative beyond predefined support. In contrast, the target-dependent relation $\psi(\mathbf{m}_w, \mathbf{m}_d)$ remains discriminative across all three regimes. On sx-askubuntu, it achieves AUCs of 0.92, 0.80, and 0.77 for repeated, common-neighbor, and *neither* queries, respectively. The same pattern holds on THGL-Software and UCI, where the *neither*-regime AUC remains 0.84 and 0.64. The same learned relation therefore remains informative even when neither recurrence nor co-occurrence support is available, providing direct representation-level evidence for relational extrapolation beyond predefined pair-statistic support.

Pair-statistic and target-agnostic signals weaken beyond predefined support, whereas the learned target-dependent relation remains discriminative. This provides representation-level evidence that relational extrapolation captures target-dependent information beyond recurrence and co-occurrence.

C.5 Additional Analysis of Learned Relations and Explicit Pair Statistics

We further analyze how the learned witness–target relation relates to the explicit pair statistics formalized in Proposition 1. All variants share the same memory, sampled witnesses, token dimension, and aggregation module, and differ only in the relational slot of each witness token. Specifically, the explicit-statistics variant uses $\mathbf{q}_w = [\mathbf{1}\{w = d\}, \log(1 + c_d(w; t))]$, corresponding to witness-level recurrence and co-occurrence, respectively. These statistics are representative of the recurrence- and co-occurrence-based signals used in existing query-aware temporal link predictors.

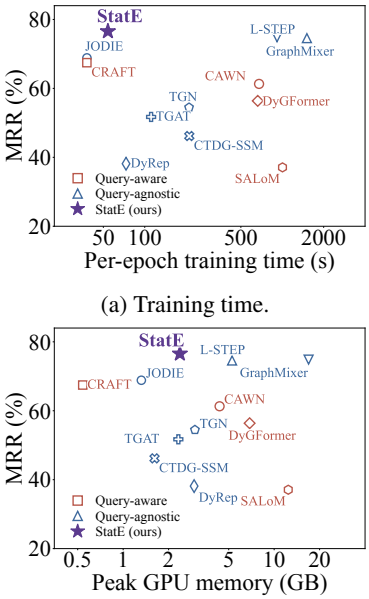
Finding 13: Explicit statistics are strong but do not fully replace learned relations. As shown in Table 3, replacing the empty target-dependent relations with explicit recurrence and co-occurrence substantially improves performance. Relative to *w/o relation*, *Stats only* improves MRR by 39.4% on UCI, 70.6% on MOOC, and 8.3% on YouTube, confirming that predefined pair statistics can provide strong target-dependent evidence. However, the learned relation is at least as effective as explicit statistics and can be substantially stronger. StatE differs from *Stats only* by -0.7% on UCI, while outperforming it by 10.6% on MOOC and 4.1% on YouTube. Therefore, the benefit of the learned witness–target relation cannot be explained solely by recovering the two explicit statistics used in *Stats only*.

C.6 Efficiency Evaluation

We evaluate training efficiency on sx-askubuntu with 964,437 temporal edges. Using a fixed batch size of 600, we compare the per-epoch training time, GPU memory usage, and MRR of StatE against 11 baselines. The same batch size is used for all methods to ensure a consistent efficiency comparison. We conduct all experiments using their official codes on our Linux GPU servers equipped with NVIDIA V100-SXM2-32GB GPUs.

Finding 14: StatE achieves an accuracy–efficiency trade-off. Fig. 8a reports per-epoch training time versus MRR. StatE achieves higher MRR while retaining training efficiency comparable to the lightweight JODIE and CRAFT. It is 12.3×–18.6× faster than the remaining query-aware methods and 1.4×–28.0× faster than query-agnostic baselines except JODIE, while delivering the best performance. These results demonstrate that relational extrapolation encoding introduces query awareness at modest computational cost.

Finding 15: StatE incurs lower memory cost than existing memory-based methods. Fig. 8b reports peak GPU memory versus MRR. On sx-askubuntu, StatE uses 2.38 GB of peak GPU memory, lower than the memory-based baselines TGN (2.99 GB) and SALoM (12.40 GB), while achieving the highest MRR. Overall, its memory footprint remains moderate among the compared methods, showing that the proposed temporal memory supports relational extrapolation without substantial memory overhead.



(a) Training time.

(b) Peak GPU memory.

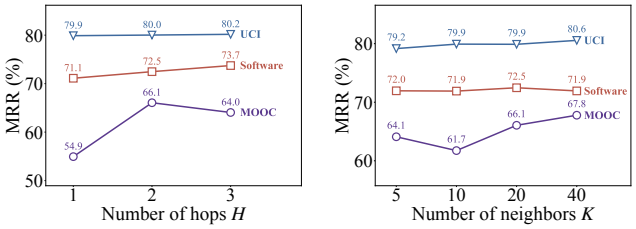
Figure 8: Efficiency evaluation on sx-askubuntu.

C.7 Parameter Sensitivity Analysis

We study the sensitivity of StatE to the number of hops H and sampled neighbors per hop K on UCI, MOOC, and Software. Fig. 9 reports the results.

Finding 16: Neighborhood depth depends on local interaction structure. On UCI and Software, increasing H from 1 to 3 changes MRR by only 0.38% and 3.66%, respectively. MOOC exhibits a different pattern: moving from one hop to two improves MRR by 20.40%, while a third hop provides no further gain. This suggests that the benefit of deeper neighborhoods depends on local interaction structure, such as neighborhood sparsity and temporal interaction patterns. Overall, one or two hops are sufficient across the evaluated datasets.

Finding 17: StatE is robust to the number of sampled witnesses. Performance of StatE remains stable across $K = 5$–40 on UCI and Software, with relative MRR variations of only 1.77% and 0.83%, respectively. MOOC is more sensitive to the witness-set size, with MRR improving by 5.77% from $K = 5$ to $K = 40$, suggesting that richer sampling can help when informative relational evidence is relatively sparse. Since increasing K also increases the relational computation cost, we use $K = 20$ throughout the paper.



(a) Hop

(b) Neighbor count

Figure 9: Effect of the number of hops and neighbors.