REVISITING TEMPORAL LINK PREDICTION THROUGH STRUCTURAL PROXIMITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing temporal link prediction methods typically rely on learned temporal representations to capture evolving interaction patterns. In this paper, we show that a classical structural proximity measure can match or outperform these methods without training or learned temporal sequence modeling. We revisit Personalized PageRank (PPR) on historical interaction graphs, using simple sliding-window retention or exponential time decay. Surprisingly, across 14 datasets, Time-decay PPR ranks first on nine and second on four among the compared methods, including datasets with few repeated edges. To understand this result, we examine the historical paths connecting source nodes to previously unseen destinations. We find that such paths contain useful predictive information, and that PPR outperforms a strong neural baseline on queries where historical structure clearly distinguishes the true destination from negative candidates. These results show that strong performance on current benchmarks does not necessarily require complex temporal models. More fundamentally, they raise the question of how much predictive value learned temporal modeling adds beyond historical structural proximity. Our findings motivate reassessing progress in temporal link prediction with strong classical baselines.

1 INTRODUCTION

Temporal graphs represent relationships between entities through interactions over time and are widely used in social networks, recommender systems, and financial transactions. A central challenge is temporal link prediction, which aims to predict which node pairs are likely to interact in the future based on observed interactions. Accurate predictions support applications such as item recommendation and social recommendations as users' interests and behaviors evolve. Future interactions may either recur between previously connected node pairs or establish new relations, requiring models to exploit historical information while generalizing beyond direct interaction history.

Existing temporal graph learning methods learn representations through dynamic node states, temporal neighborhood aggregation, or historical sequence encoding (Rossi et al., 2020; Xu et al., 2020; Cong et al., 2023; Yu et al., 2023). In contrast, EdgeBank shows that simply memorizing historical edges can already achieve strong performance when interactions frequently recur, particularly under evaluation settings with easily distinguishable negative candidates (Poursafaei et al., 2022). This observation has motivated benchmarks that place greater emphasis on prediction beyond direct interaction memorization. TGB broadens evaluation across diverse domains and more realistic protocols (Huang et al., 2023), while TGB-Seq introduces datasets with fewer repeated edges to challenge models to capture sequential dynamics and generalize to unseen relations (Yi et al., 2025b). More recently, CRAFT matches candidate destinations against the source's recent interaction history using learnable node embeddings and target-aware cross-attention, achieving strong performance with a comparatively simple architecture (Yi et al., 2025a).

However, reducing repeated edges does not remove the structural information contained in the historical graph. A source and a previously unseen destination may never have interacted directly while still being connected through historical multi-hop paths. This structural view has a long history in classical link prediction. Common Neighbors and Adamic–Adar score node pairs through local neighborhood overlap, while Katz and rooted PageRank capture multi-step connections (Liben-Nowell & Kleinberg, 2003). Personalized PageRank (PPR) provides a natural probe of structural

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.

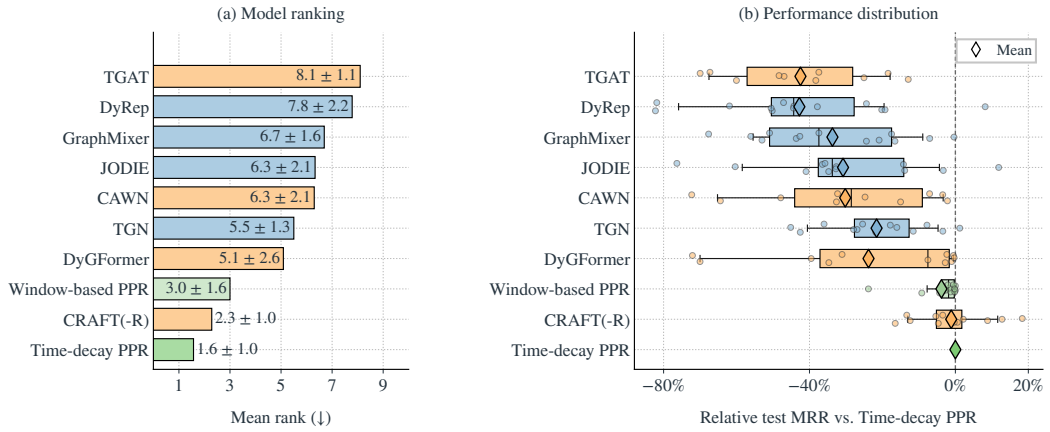

Figure 1: (a) Mean rank over each method's reported results across 14 datasets; missing results are excluded. Standard deviations are across datasets, and ties share the same rank. (b) Relative test MRR against Time-decay PPR across available results from 14 datasets. Dots represent dataset results, boxes indicate interquartile ranges, whiskers mark the 10th–90th percentiles, and diamonds denote means. Negative values favor Time-decay PPR.

proximity: starting from a query source, it aggregates random walks of different lengths and therefore assigns scores not only to previous neighbors but also to nodes reachable through multi-hop paths (Gleich, 2015). Consequently, a destination can receive a high PPR score even when it has never directly interacted with the source. This raises a more basic question: **how much predictive information does historical multi-hop structure provide for future link prediction?**

To investigate this question, we revisit PPR as a controlled, training-free baseline for measuring the predictive value of historical structural proximity. To accommodate evolving interaction streams, we construct historical graphs using simple temporal retention mechanisms. Window-based PPR keeps only recent interactions, while Time-decay PPR gradually downweights older events. As shown in Figure 1, Window-based PPR performs competitively with existing temporal graph models, showing that multi-hop structure in a recent historical graph provides a strong predictive signal. More surprisingly, Time-decay PPR ranks first on nine of the 14 datasets and second on another four among the compared methods, including datasets with few repeated edges.

Further analysis connects this performance to how historical structure evolves. New interactions often close existing multi-hop paths into cycles, suggesting that future links are frequently foreshadowed by structural connectivity already present in the historical graph. At the query level, historical path patterns can distinguish the true destination from negative candidates, and PPR outperforms a strong neural baseline on queries where historical structural signals are particularly discriminative. These findings show that historical structural proximity remains highly predictive even when direct edge recurrence is limited. This does not imply that temporal modeling is unnecessary. Instead, our results suggest that future progress should be evaluated relative to the structural signal already available in historical graphs. This motivates evaluation protocols that better control for structural proximity and targeted analysis of queries where historical paths are weak or ambiguous. They also suggest developing models that focus temporal learning on information not already explained by historical connectivity. Our main contributions are as follows:

- We systematically revisit PPR as a training-free structural baseline for future link prediction. Across 14 datasets, Time-decay PPR ranks first on nine and second on four, while PPR without temporal decay also remains competitive.
- We analyze the predictive value of historical structural proximity from the perspective of graph evolution. By examining cycle closure, historical path discriminability, and model predictions, we study how indirect historical connections support unseen-link prediction and how effectively models use this evidence.
- We disentangle the effects of historical retention, interaction recency, and repeated interactions, and use these analyses to reassess how much predictive value learned temporal modeling adds beyond historical graph structure under current evaluation protocols.

2 PRELIMINARIES AND RELATED WORK

2.1 PROBLEM FORMULATION

Temporal Graph. A temporal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a node set $\mathcal{V}$ and a sequence of timestamped interactions $\mathcal{E}$, ordered by non-decreasing time. Each interaction $e_i = (u_i, v_i, t_i)$ records an event between source node u_i and destination node v_i at timestamp t_i, such as a purchase or a message. Multiple interactions may occur between the same source and destination at different timestamps.

Future Link Prediction. Given a query source s, query time t, and the interaction history $\mathcal{H}_t = \{(u_i, v_i, t_i) \in \mathcal{E} \mid t_i < t\}$, future link prediction aims to identify the destination d that interacts with s at time t. Following the evaluation protocols of TGB and TGB-Seq, we formulate this task as candidate ranking (Huang et al., 2023; Yi et al., 2025b). A candidate destination set $\mathcal{C}$ contains the true destination d and a set of negative candidates. Using only $\mathcal{H}_t$, a predictor assigns a score $f(s, v, t)$ to each candidate $v \in \mathcal{C}$, with the objective of ranking the true destination ahead of the negative candidates.

2.2 RELATED WORK

Existing approaches to future link prediction exploit historical interactions through both structural proximity and learned temporal representations.

Structural Proximity and Personalized PageRank. Structural proximity methods score node pairs directly from neighborhood overlap and multi-step connections in historical networks. Common Neighbors and Adamic–Adar capture local associations through shared neighbors, whereas Katz and rooted PageRank account for connections or walks of different lengths. Liben-Nowell & Kleinberg (2003) systematically compared these measures, demonstrating the predictive value of historical topology for new links. PPR produces rankings personalized to a node or a set of nodes and has also been used for personalized search and local community detection (Jeh & Widom, 2003; Avron & Horesh, 2015).

Subsequent work has explored incremental ranking updates as graphs change and their use in tracking local communities (Zhang et al., 2016; Nathan et al., 2017), as well as incorporating diffusion into graph representation learning. PPNP/APPNP (Klicpera et al., 2019) combines node prediction with PPR propagation, while GPR-GNN (Chien et al., 2021) learns weights for different propagation orders. PPR thus serves both as a direct measure of structural proximity and as a propagation component in learned models.

Dynamic Graph Representation Learning. Dynamic graph representation learning extracts predictive information through node-state updates, temporal neighborhood aggregation, and historical sequence encoding. JODIE, DyRep, and TGN maintain node states that evolve with interaction events (Kumar et al., 2019; Trivedi et al., 2019; Rossi et al., 2020). TGAT, GraphMixer, and DyGFormer learn node associations from time-aware neighborhoods or interaction sequences (Xu et al., 2020; Cong et al., 2023; Yu et al., 2023). CRAFT instead predicts interactions by matching candidate destinations against the source's recent history (Yi et al., 2025a).

Beyond learned representations, EdgeBank (Poursafaei et al., 2022) shows that a training-free memory of historical edges can perform strongly when interactions frequently recur, exposing limitations of evaluation settings with easily distinguishable negative candidates. PopTrack tracks recent destination popularity and highlights the influence of negative sampling on temporal graph evaluation (Daniluk & Dabrowski, 2023). Recency- and frequency-based heuristics can also match or outperform neural models on several temporal graph benchmarks (Cornell et al., 2025). TGB (Huang et al., 2023) extends evaluation to large-scale datasets from diverse domains and more challenging candidate-ranking protocols, while TGB-Seq (Yi et al., 2025b) further emphasizes datasets with few repeated edges, highlighting the difficulty of predicting unseen relations and capturing sequential dynamics. These developments motivate revisiting a complementary question: how much predictive information remains in the historical graph structure itself under modern future link prediction protocols?

3 REVISITING PPR FOR TEMPORAL LINK PREDICTION

In this section, we describe how we use PPR (Brin & Page, 1998; Gleich, 2015) to measure the predictive value of historical graph structure for future link prediction. We first formulate finite-step PPR as a multi-hop structural proximity measure (Klicpera et al., 2019), then describe how timestamped interactions are converted into historical graphs through temporal retention and repeated-interaction aggregation. Finally, we use the resulting PPR scores to rank candidate destinations.

3.1 PPR AS MULTI-HOP STRUCTURAL PROXIMITY

Personalized PageRank measures proximity to a source node through a random walk with restart. For a transition matrix $\mathbf{P}$ and source one-hot row vector $\mathbf{e}_s$, the stationary PPR vector $\boldsymbol{\pi}_s$ satisfies

$$\boldsymbol{\pi}_s = \alpha \mathbf{e}_s + (1-\alpha)\boldsymbol{\pi}_s \mathbf{P}, \tag{1}$$

where $\alpha \in (0,1)$ is the restart probability. At each step, the walk returns to the source with probability α and follows an outgoing graph transition with probability $1-\alpha$. We use a finite-step approximation initialized by $\mathbf{p}_s^{(0)} = \mathbf{e}_s$:

$$\mathbf{p}_s^{(\ell+1)} = \alpha \mathbf{e}_s + (1-\alpha)\mathbf{p}_s^{(\ell)}\mathbf{P}, \qquad \ell = 0, \ldots, L-1. \tag{2}$$

After L iterations,

$$\mathbf{p}_s^{(L)} = \alpha \sum_{\ell=0}^{L-1} (1-\alpha)^{\ell} \mathbf{e}_s \mathbf{P}^{\ell} + (1-\alpha)^{L} \mathbf{e}_s \mathbf{P}^{L}. \tag{3}$$

Each term $\mathbf{e}_s\mathbf{P}^{\ell}$ captures connectivity through walks of length ℓ. Therefore, finite-step PPR combines direct and multi-hop structural connections up to L propagation steps, while α controls their relative contributions. Importantly, a candidate that has never directly interacted with the source can still receive a nonzero score when it is connected to the source through historical multi-hop paths. We also include $\alpha = 0$, for which the finite-step iteration reduces to an L-step random walk without restart, with scores given by $\mathbf{e}_s\mathbf{P}^{L}$.

3.2 HISTORICAL GRAPH CONSTRUCTION

Applying PPR to temporal link prediction requires representing interactions of different ages and repeated occurrences in a historical graph. We construct an undirected weighted graph from interactions observed before the query time t, without adding self-loops. Temporal retention determines how each event is weighted, and repeated-interaction aggregation combines events between the same node pair. Let $\mathcal{T}_{uv}(t)$ denote the multiset of interaction timestamps between u and v before t.

Temporal retention. We assign each interaction a weight based on its age Δt, using either exponential decay with half-life $h > 0$ (Zuo et al., 2018; Qiu et al., 2020; Lu et al., 2024) or a sliding window of length $W > 0$ (Poursafaei et al., 2022):

$$\kappa_h^{\text{decay}}(\Delta t) = 2^{-\Delta t/h}, \qquad \kappa_W^{\text{window}}(\Delta t) = \mathbb{I}[\Delta t \le W]. \tag{4}$$

Exponential decay emphasizes recent interactions while preserving older connections. Sliding windows impose a hard cutoff that can remove connections supporting multi-step propagation. We refer to the corresponding PPR variants as Time-decay PPR and Window-based PPR, respectively. Both include full-history retention as h or W tends to infinity.

Repeated-interaction aggregation. For either retention function κ, we combine events from the same node pair as

$$w_{uv}(t) = \mathcal{A}\big(\{\kappa(t-\tau) : \tau \in \mathcal{T}_{uv}(t)\}\big), \qquad \mathcal{A} \in \{\max, \text{sum}\}, \tag{5}$$

where the braces denote a multiset. Max aggregation retains the latest interaction's weight, yielding last-decay or binary-window weighting; sum aggregation accumulates all event weights, yielding sum-decay or count-window weighting. These choices distinguish the recency or presence of a relation from the accumulated evidence of its recurrence. The weights define a symmetric adjacency matrix $\mathbf{A}_t$, which we row-normalize to obtain $\mathbf{P}_t$.

3.3 Finite-Step Diffusion and Candidate Scoring

Given the historical transition matrix $\mathbf{P}_t$, we run finite-step PPR from the query source s. For each candidate destination $v \in \mathcal{C}$, we use $f(s, v, t) = [\mathbf{p}_t^{(L)}(s)]_v$ as its prediction score and rank candidates in descending order, following source-rooted PageRank for link prediction (Liben-Nowell & Kleinberg, 2003).

4 Experiments

4.1 Experimental Setup

Datasets and baselines. We evaluate 14 datasets from TGB, TGB-Seq, and established temporal graph benchmarks, grouped into seen-dominant and unseen-dominant regimes following Yi et al. (2025a). Their domains, sizes, and repeated-edge ratios are summarized in Appendix A. We compare with the reported results of JODIE, DyRep, TGAT, TGN, CAWN, GraphMixer, DyGFormer, and CRAFT, using CRAFT-R for the seen-dominant group and CRAFT for the unseen-dominant group (Yi et al., 2025a). Model references and configuration details are in Appendix B. We additionally evaluate EdgeBank (Poursafaei et al., 2022) and heuristics based on pairwise interaction history and destination recency, frequency, and activity; definitions and results are provided in Appendix G.1.

Evaluation and selection. We use chronological training, validation, and test splits and evaluate candidate ranking by Mean Reciprocal Rank (MRR). Released split assignments and benchmark-provided negative candidates are retained where available (Huang et al., 2023; Yi et al., 2025b); other candidate sets follow their corresponding ranking protocols. For both Time-decay PPR and Window-based PPR, we select configurations by validation MRR before test evaluation; the search space is detailed in Appendix B. PPR scores each timestamp group or timestamp-aligned batch before updating the historical graph with its positive interactions; refresh settings are detailed in Appendix B.

4.2 Main Results

Time-decay PPR achieves the best test MRR on 9 of the 14 datasets and ranks second on another 4, including ties (Tables 1 and 2). Without learning node representations or prediction parameters, it remains competitive on both seen-dominant and unseen-dominant datasets. Window-based PPR is also competitive, but Time-decay PPR matches or exceeds it on every dataset. We examine the role of temporal retention in Section 5.3.

Table 1: Test MRR $\times 100$ (↑) on unseen-dominant datasets. Baseline results are from Yi et al. (2025a).

Method	GoogleLocal	Yelp	Taobao	YouTube	Flickr	WikiLink	tgbl-review
JODIE	41.86±1.49	69.88±0.31	48.36±2.18	41.67±2.86	46.21±0.83	57.94±1.33	41.43±0.15
DyRep	37.73±1.34	57.69±1.05	40.03±2.40	35.12±4.13	38.04±4.19	42.63±1.33	40.06±0.59
TGAT	19.78±0.24	—	—	43.56±2.53	23.53±3.35	—	19.64±0.23
TGN	54.13±1.97	69.79±0.24	60.28±0.54	55.16±5.89	46.03±6.78	62.94±2.16	37.48±0.23
CAWN	18.26±0.02	25.71±0.09	—	47.55±1.08	48.69±6.08	—	19.30±0.10
GraphMixer	21.31±0.14	33.96±0.19	31.54±0.02	58.87±0.12	45.01±0.08	48.57±0.02	36.89±1.50
DyGFormer	18.39±0.02	21.68±0.20	—	46.08±3.44	49.58±2.87	—	22.39±1.52
CRAFT	62.35±0.46	**72.69**±0.61	70.68±0.42	58.92±0.16	62.34±0.84	75.48±0.84	**41.77**±0.06
Window-based PPR	**65.89**±0.00	71.31±0.00	70.60±0.00	67.51±0.00	70.46±0.00	**86.16**±0.00	28.20±0.00
Time-decay PPR	**65.89**±0.00	72.28±0.00	**71.81**±0.00	**70.56**±0.00	**71.93**±0.00	**86.16**±0.00	37.03±0.00

Mean ± standard deviation (deterministic PPR: ±0.00). **1st**, 2nd, and 3rd use displayed means; ties share ranks. A dash denotes an unreported result.

On seen-dominant datasets where Time-decay PPR outperforms the strongest reported baselines, its relative MRR gains are generally modest, reaching approximately 2.9% on UCI and 2.4% on

Table 2: Test MRR $\times 100$ ($\uparrow$) on seen-dominant datasets. Baseline results are from Yi et al. (2025a).

Method	Wikipedia	Reddit	LastFM	UCI	Flights	tgbl-coin	tgbl-flight
JODIE	76.48±1.72	77.16±1.27	18.49±2.87	51.43±1.74	20.37±4.18	—	—
DyRep	67.42±4.21	72.33±2.14	17.76±6.56	14.00±2.71	15.62±0.47	45.20±4.60	55.60±1.40
TGAT	72.72±1.50	78.03±0.25	24.10±1.26	31.50±0.48	53.90±1.07	60.92±0.57	—
TGN	82.22±0.39	79.25±0.24	25.59±4.43	45.29±6.43	64.24±2.11	58.60±3.70	70.50±2.00
CAWN	86.07±0.08	87.66±0.04	35.10±0.34	66.96±0.48	80.06±2.10	—	—
GraphMixer	73.57±1.48	70.87±0.38	26.76±1.42	59.58±1.00	42.26±0.49	75.57±0.27	—
DyGFormer	88.69±0.04	88.70±0.05	46.50±0.51	76.61±0.26	84.13±3.46	75.17±0.38	—
CRAFT-R	88.25±0.26	89.33±0.11	**55.21**±0.23	75.11±0.09	83.16±0.16	**88.47**±0.25	**91.39**±0.03
Window-based PPR	86.01±0.00	89.44±0.00	44.85±0.00	71.54±0.00	85.69±0.00	79.03±0.00	89.25±0.00
Time-decay PPR	**89.17**±0.00	**89.61**±0.00	46.66±0.00	**78.80**±0.00	**86.15**±0.00	81.29±0.00	89.45±0.00

Mean ± standard deviation (deterministic PPR: ±0.00). **1st**, 2nd, and 3rd use displayed means; ties share ranks. A dash denotes an unreported result.

Flights. More strikingly, it remains highly competitive on unseen-dominant datasets, where direct edge recurrence is less prevalent. Its relative MRR gains over the strongest reported baselines reach approximately 19.8% on YouTube, 15.4% on Flickr, and 14.2% on WikiLink, with smaller gains of 5.7% on GoogleLocal and 1.6% on Taobao. These results motivate our analysis in Section 5 of how historical paths and structural proximity support the prediction of unseen relations.

Additional comparisons in Appendix G.1 show that pair-recency and pair-frequency perform strongly on several seen-dominant datasets. By contrast, destination activity outperforms these pair-based heuristics on GoogleLocal, YouTube, and Flickr and approaches the strongest reported neural baseline on tgbl-review. These results highlight the dataset-dependent effectiveness of simple historical statistics. Time-decay PPR, however, remains competitive across both seen- and unseen-dominant regimes, suggesting that historical structural proximity provides useful predictive information across diverse interaction patterns. We further evaluate activity-weighted negative sampling on five datasets. Time-decay PPR remains competitive under activity-weighted candidate sets; complete results and analysis are provided in Appendix G.2.

5 Analysis

In this section, we analyze how historical structural proximity supports temporal link prediction. We first examine cycle formation in temporal graphs relative to a random reference, then study how often new links close historical paths and whether these paths distinguish positive from negative candidates. We further investigate how effectively a strong neural baseline uses this structural evidence. Finally, we examine the effects of temporal retention, repeated-interaction aggregation, and propagation.

5.1 Structural Evolution and Its Predictive Value

PPR's strong performance on unseen-dominant datasets motivates examining how historical structural proximity supports the prediction of unseen relations. New interactions may connect nodes already linked by historical paths, closing these paths into cycles (Figure 2). We first ask whether such cycles accumulate earlier in real interaction sequences than under random edge ordering.

Motivated by randomized-reference models for temporal networks (Gauvin et al., 2022), we use the following reference.

Definition (Random-order reference). Let $G_M = (V, E_M)$ be the final aggregated simple graph with M distinct edges. Draw a uniformly random permutation of E_M, and let $\widetilde{G}_m$ contain its first m edges. The expected number of simple k-cycles is

$$\mathbb{E}[N_k(\widetilde{G}_m) \mid G_M] = N_k(G_M)\frac{\binom{m}{k}}{\binom{M}{k}}. \tag{6}$$

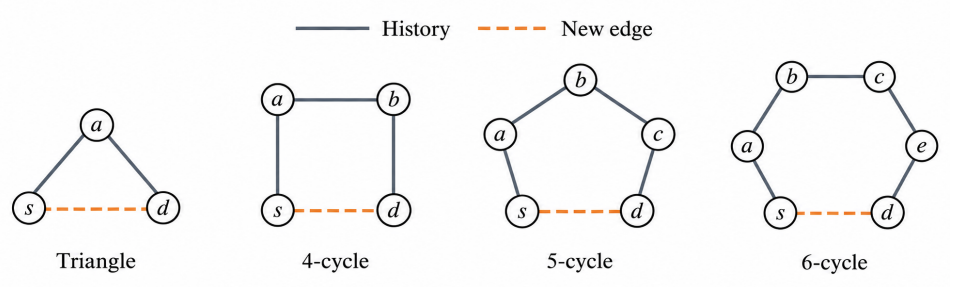

Figure 2: A new edge closes a historical path into a cycle.

Across seven datasets, observed four- and six-cycle counts exceed this expectation at every measured intermediate snapshot, indicating earlier cycle accumulation under the real ordering. Appendix D provides the derivation and comparison curves.

To connect this evolution pattern to incoming interactions, each new edge closes a k-cycle for every existing $(k-1)$-step simple path between its endpoints. Inspired by EdgeBank's temporal edge appearance plots (Poursafaei et al., 2022), Figure 3 extends the repeated/new distinction by grouping new pairs according to cycle closure. Each node pair is counted once per display interval. Wikipedia exhibits substantial recurrence, while new relations on UCI, GoogleLocal, and YouTube often close historical paths. Paths are defined on the aggregated historical graph without requiring chronological order among their interactions.

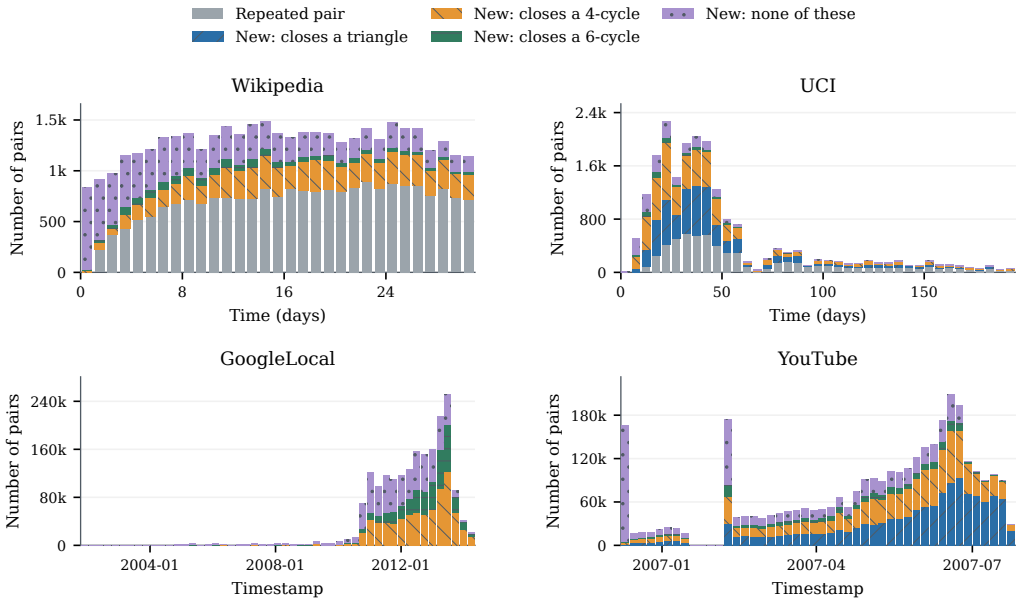

Figure 3: Unique node pairs per time interval, grouped by recurrence and closure of historical triangles, four-cycles, or six-cycles. Earlier categories take precedence; definitions are in Appendix C.

Historical paths provide predictive evidence when they distinguish positive from negative candidates. Figure 7 in Appendix C compares path coverage on complete test sets, using prediction-time graphs and original candidates. Wikipedia and UCI show strong one-step coverage gaps; GoogleLocal and YouTube have near-zero one-step coverage but clear three-step and two-step gaps, respectively. Table 6 further evaluates ranking by unweighted simple-path counts. For previously unconnected nodes in a graph without self-loops, every three-step walk is a simple path, linking this diagnostic to PPR's transition-weighted three-step component. We next examine how effectively PPR and a strong neural baseline use this evidence.

5.2 Why Do Neural Models Fall Short?

We compare Time-decay PPR with CRAFT (Yi et al., 2025a), a strong neural baseline using target-aware matching of recent interaction histories, on GoogleLocal and YouTube. Test queries are grouped by how well historical paths distinguish positive from negative candidates. In the strong-structure group, the true destination has strictly more three-step simple paths from the source than every negative candidate. Path count alone therefore ranks it uniquely first. The remaining queries form medium-structure, weak-structure, and no-three-step-support groups, defined in Appendix E. Grouping uses historical path counts rather than model scores.

Within the strong-structure group, PPR achieves relative MRR improvements of approximately **9.3% and 13.5%** over the evaluated CRAFT system on GoogleLocal and YouTube, respectively (Figure 4). These queries account for **38.85% and 49.13%** of the respective test sets. When weighted by their query proportions, the within-group performance gaps account for nearly all of PPR's overall net MRR gain on GoogleLocal and approximately half on YouTube (Table 8). These results suggest that modeling historical interaction sequences does not necessarily ensure effective use of higher-order structural information for link prediction.

5.3 Temporal Retention, Aggregation, and Propagation

We next examine how historical graph construction and propagation parameters shape PPR's use of the structural evidence identified above.

Temporal retention and repeated interactions. Table 3 compares exponential decay, sliding-window retention, and full-history retention without decay. For the window ablation, we retain the aggregation and propagation parameters of Time-decay PPR and select only the window length using validation MRR. Exponential decay achieves the best or tied-best performance across all eight datasets among the evaluated configurations. This suggests that temporal link prediction benefits from emphasizing recent interactions while retaining temporally weighted long-term history. Figure 5 further shows that an appropriate half-life outperforms both an overly short half-life and no decay. The preferred time scale varies across datasets, motivating adaptive time scales and multi-scale history modeling.

Figure 4: Test MRR by structural group; percentages denote query shares.

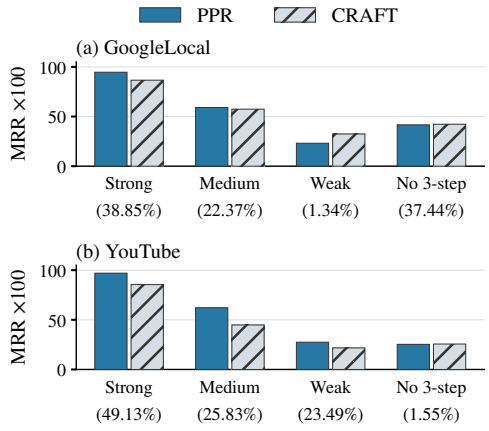

Table 3: Ablations of Time-decay PPR (validation MRR ×100).

Dataset	Time-decay PPR	w. window, w/o decay	w/o decay
Wikipedia	**90.74**	86.69	85.42
Reddit†	**89.46**	89.28	89.28
LastFM	**43.05**	40.68	31.46
UCI	**72.60**	66.26	51.42
Flights	**85.71**	85.16	85.06
YouTube	**75.64**	71.21	72.11
Flickr	**73.75**	70.78	70.46
tgbl-coin	**78.69**	75.26	71.08

† The selected window covers all history. Paired settings are given in Appendix F.

Figure 5 also illustrates the joint effects of half-life and repeated-interaction aggregation. On unseen-dominant datasets such as YouTube and Flickr, the low prevalence of repeated interactions results in little difference between last-decay and sum-decay. On seen-dominant datasets, however, accumulating repeated interactions offers no consistent advantage: across the evaluated finite half-lives, last-decay consistently performs better on Wikipedia, whereas sum-decay consistently performs better on Reddit and Flights. These findings suggest that the choice of aggregation should account for the interaction patterns and temporal distribution of repeated events. Complete curves for seven datasets, including Flickr, are provided in Appendix F.

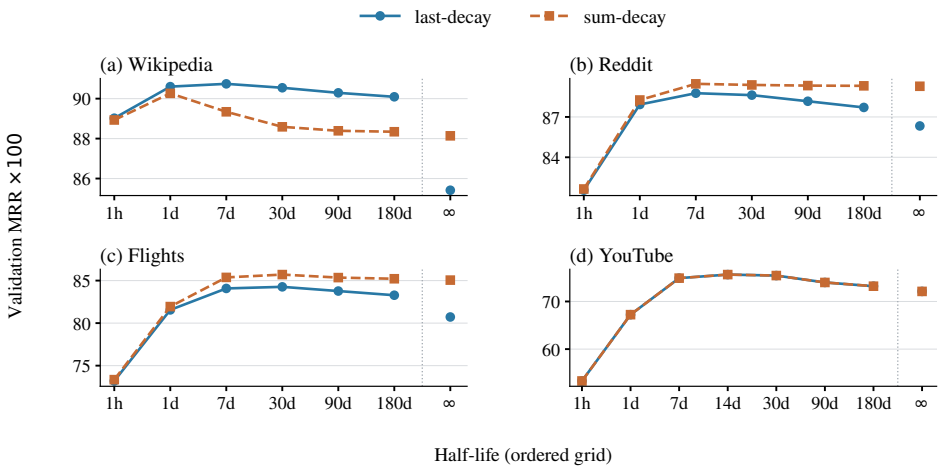

Figure 5: Validation sensitivity to half-life and aggregation, with α and propagation depth fixed. Half-lives follow an ordered grid; ∞ denotes no decay. Each panel has its own vertical scale.

Propagation parameters. Figure 6 shows how propagation depth K (the iteration count L in Section 3.1) and restart probability α affect prediction performance. On most datasets, shallow propagation achieves or approaches the best performance. One-step propagation already performs well on seen-dominant datasets, whereas unseen-dominant datasets typically obtain their main gains at $K = 3$ to 5; increasing the depth further does not consistently improve performance. Preferred values of α also vary across datasets, motivating further exploration of learnable weights over propagation orders. All sensitivity results use validation data; additional datasets and fixed settings are reported in Appendix F.

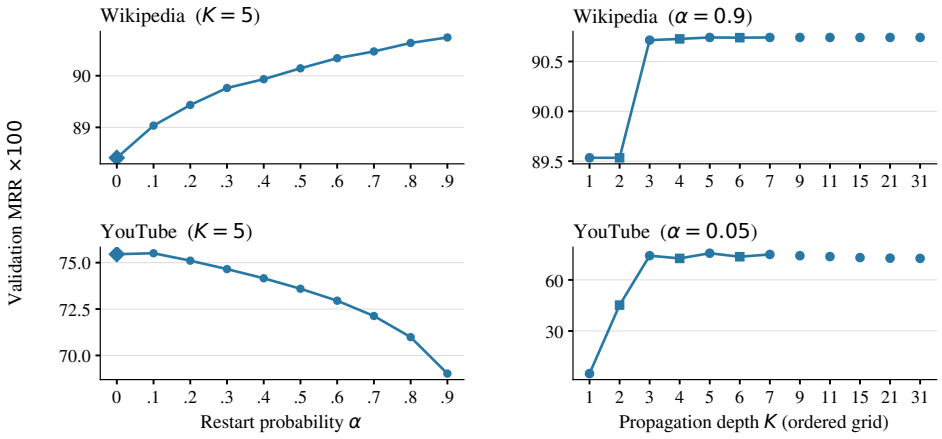

Figure 6: Validation sensitivity to α and K on Wikipedia and YouTube. Other parameters are fixed. K uses an ordered grid; markers show measured values. Each panel has its own vertical scale.

6 Conclusion

In this paper, we revisit temporal link prediction through structural proximity and reassess classical Personalized PageRank as a training-free baseline. With appropriate temporal retention and repeated-interaction aggregation, the resulting Time-decay PPR achieves strong performance across widely used benchmarks. Our analyses connect this performance to historical structural proximity, showing that multi-step historical paths provide useful predictive evidence even for previously unseen relations. Comparisons with a strong neural baseline further reveal a gap between the structural evidence available and its use in prediction. These findings motivate temporal graph models that better exploit historical structural proximity and highlight the importance of strong classical baselines when evaluating progress in temporal link prediction.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and language polishing, as well as the implementation of part of the code. We did not use generative AI tools to design experiments, assist with proofs, interpret results, or propose research hypotheses. The LLM-generated code was verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the PPR variants, and Section 4.1 specifies the evaluation protocol. Appendix B provides hyperparameter settings and graph-update procedures. Additional baselines and candidate construction are detailed in Appendix G.

REFERENCES

Haim Avron and Lior Horesh. Community detection using time-dependent personalized pagerank. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1795–1803. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/avron15.html.

Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Ubicrawler: a scalable fully distributed web crawler. *Softw. Pract. Exp.*, 34(8):711–726, 2004. doi: 10.1002/SPE.587. URL https://doi.org/10.1002/spe.587.

Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: a multiresolution coordinate-free ordering for compressing social networks. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar (eds.), *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pp. 587–596. ACM, 2011. doi: 10.1145/1963405.1963488. URL https://doi.org/10.1145/1963405.1963488.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Networks*, 30(1-7):107–117, 1998. doi: 10.1016/S0169-7552(98)00110-X. URL https://doi.org/10.1016/S0169-7552(98)00110-X.

Meeyoung Cha, Alan Mislove, and P. Krishna Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl (eds.), *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pp. 721–730. ACM, 2009. doi: 10.1145/1526709.1526806. URL https://doi.org/10.1145/1526709.1526806.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=n6jl7fLxrP.

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 Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ayPPc0SyLv1.

Filip Cornell, Oleg Smirnov, Gabriela Zarzar Gandler, and Lele Cao. On the power of heuristics in temporal graphs. In Arno Blaas, Priya D'Costa, Fan Feng, Andreas Kriegler, Ian Mason, Zhaoying Pan, Tobias Uelwer, Jennifer Williams, Yubin Xie, and Rui Yang (eds.), *Proceedings on "I Can't Believe It's Not Better: Challenges in Applied Deep Learning" at ICLR 2025 Workshops, 28 April 2025, Singapore*, volume 296 of *Proceedings of Machine Learning Research*, pp. 37–46. PMLR, 2025. URL https://proceedings.mlr.press/v296/cornell25a.html.

Michal Daniluk and Jacek Dabrowski. Temporal graph models fail to capture global temporal dynamics. *CoRR*, abs/2309.15730, 2023. doi: 10.48550/ARXIV.2309.15730. URL https://doi.org/10.48550/arXiv.2309.15730.

Laetitia Gauvin, Mathieu Génois, Márton Karsai, Mikko Kivelä, Taro Takaguchi, Eugenio Valdano, and Christian L. Vestergaard. Randomized reference models for temporal networks. *SIAM Rev.*, 64(4):763–830, 2022. doi: 10.1137/19M1242252. URL https://doi.org/10.1137/19m1242252.

David F. Gleich. Pagerank beyond the web. *SIAM Rev.*, 57(3):321–363, 2015. doi: 10.1137/140976649. URL https://doi.org/10.1137/140976649.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael M. Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/066b98e63313162f6562b35962671288-Abstract-Datasets_and_Benchmarks.html.

Glen Jeh and Jennifer Widom. Scaling personalized web search. In Gusztáv Hencsey, Bebo White, Yih-Farn Robin Chen, László Kovács, and Steve Lawrence (eds.), *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pp. 271–279. ACM, 2003. doi: 10.1145/775152.775191. URL https://doi.org/10.1145/775152.775191.

Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. A re-visit of the popularity baseline in recommender systems. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pp. 1749–1752. ACM, 2020. doi: 10.1145/3397271.3401233. URL https://doi.org/10.1145/3397271.3401233.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 1269–1278. ACM, 2019. doi: 10.1145/3292500.3330895. URL https://doi.org/10.1145/3292500.3330895.

Jiacheng Li, Jingbo Shang, and Julian J. McAuley. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 6159–6169. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.426. URL https://doi.org/10.18653/v1/2022.acl-long.426.

David Liben-Nowell and Jon M. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*, pp. 556–559. ACM, 2003. doi: 10.1145/956863.956972. URL https://doi.org/10.1145/956863.956972.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. In Amir Globersons, Lester Mackey, Danielle

Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ff7bf6014f7826da531aa50f4538ee19-Abstract-Conference.html.

Alan Mislove, Massimiliano Marcon, P. Krishna Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In Constantine Dovrolis and Matthew Roughan (eds.), *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference, IMC 2007, San Diego, California, USA, October 24-26, 2007*, pp. 29–42. ACM, 2007. doi: 10.1145/1298306.1298311. URL https://doi.org/10.1145/1298306.1298311.

Eisha Nathan, Anita Zakrzewska, E. Jason Riedy, and David A. Bader. Local community detection in dynamic graphs using personalized centrality. *Algorithms*, 10(3):102, 2017. doi: 10.3390/A10030102. URL https://doi.org/10.3390/a10030102.

Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *J. Assoc. Inf. Sci. Technol.*, 60(5):911–932, 2009. doi: 10.1002/ASI.21015. URL https://doi.org/10.1002/asi.21015.

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d49042a5d49818711c401d34172f9900-Abstract-Datasets_and_Benchmarks.html.

Zhenyu Qiu, Wenbin Hu, Jia Wu, Weiwei Liu, Bo Du, and Xiaohua Jia. Temporal network embedding with high-order nonlinear information. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5436–5443. AAAI Press, 2020. doi: 10.1609/AAAI.V34I04.5993. URL https://doi.org/10.1609/aaai.v34i04.5993.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *CoRR*, abs/2006.10637, 2020. URL https://arxiv.org/abs/2006.10637.

Rakshit S. Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HyePrhR5KX.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=KYPz4YsCPj.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rJeW1yHYwH.

An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian John McAuley. Personalized showcases: Generating multi-modal explanations for recommendations. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (eds.), *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pp. 2251–2255. ACM, 2023. doi: 10.1145/3539618.3592036. URL https://doi.org/10.1145/3539618.3592036.

Lu Yi, Runlin Lei, Fengran Mo, Yanping Zheng, Zhewei Wei, and Yuhang Ye. Future link prediction without memory or aggregation. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025a. URL http://papers.nips.cc/paper_files/paper/2025/hash/036912a83bdbb1fd792baf6532f102d8-Abstract-Conference.html.

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 Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=8e2LirwiJT.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d611019afba70d547bd595e8a4158f55-Abstract-Conference.html.

Hongyang Zhang, Peter Lofgren, and Ashish Goel. Approximate personalized pagerank on dynamic graphs. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1315–1324. ACM, 2016. doi: 10.1145/2939672.2939804. URL https://doi.org/10.1145/2939672.2939804.

Jingwei Zhuo, Ziru Xu, Wei Dai, Han Zhu, Han Li, Jian Xu, and Kun Gai. Learning optimal tree models under beam search. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11650–11659. PMLR, 2020. URL http://proceedings.mlr.press/v119/zhuo20a.html.

Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In Yike Guo and Faisal Farooq (eds.), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 2857–2866. ACM, 2018. doi: 10.1145/3219819.3220054. URL https://doi.org/10.1145/3219819.3220054.

A Dataset Statistics

Table 4: Dataset statistics reported by Yi et al. (2025a).

Dataset	Interaction	Nodes	Events	Timestamps	Repeat (%)
Unseen-dominant					
GoogleLocal	Business review	206,244/267,336	1,913,967	1,771,060	0.00
Yelp	Business review	1,338,688/405,081	19,760,293	14,646,734	25.18
Taobao	Product interaction	760,617/863,016	18,853,792	139,171	16.58
YouTube	Follow	402,422	3,288,028	203	0.00
Flickr	Follow	233,836	7,223,559	134	0.00
WikiLink	Hyperlink	1,361,972	34,163,774	2,198	0.00
tgbl-review	Product review	352,636/298,590	4,873,540	6,865	0.19
Seen-dominant					
Wikipedia	Page edit	8,227/1,000	157,474	152,757	88.41
Reddit	Subreddit interaction	10,000/984	672,447	669,065	88.32
LastFM	Song listening	980/1,000	1,293,103	1,283,614	88.01
UCI	Message	1,899	59,835	58,911	66.06
Flights	Flight	13,169	1,927,145	122	79.50
tgbl-coin	Transaction	638,486	22,809,486	1,295,720	82.93
tgbl-flight	Flight	18,143	67,169,570	1,385	96.48

Bipartite node counts are source/destination counts; timestamps are distinct. Repeat is the percentage of events whose ordered pair appeared strictly earlier (Yi et al., 2025b).

Wikipedia, Reddit, and LastFM record page edits, subreddit interactions, and music listening, respectively (Kumar et al., 2019). UCI contains user messages (Panzarasa et al., 2009), and Flights records flights between airports (Poursafaei et al., 2022). TGB-Seq provides the benchmark versions of GoogleLocal, Yelp, and Taobao (business and product interactions), YouTube and Flickr (social links), and WikiLink (hyperlinks) (Yi et al., 2025b). Their underlying sources include Google Local Reviews (Li et al., 2022; Yan et al., 2023), the Yelp Open Dataset, and the Taobao User Behavior dataset (Zhuo et al., 2020); Flickr (Cha et al., 2009), YouTube (Mislove et al., 2007), and WikiLink (Boldi et al., 2004; 2011) are traced to the cited network collections. The tgbl-review, tgbl-coin, and tgbl-flight datasets are from TGB and cover product reviews, financial transactions, and flights, respectively (Huang et al., 2023). Table 4 reports dataset sizes and recurrence rates as reported by Yi et al. (2025a).

B Experimental Details

The main tables use baseline results reported by Yi et al. (2025a). The compared methods are JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020), TGN (Rossi et al., 2020), CAWN (Wang et al., 2021), GraphMixer (Cong et al., 2023), DyGFormer (Yu et al., 2023), and CRAFT (Yi et al., 2025a). These cover recurrent node states, temporal neighborhood aggregation, temporal walks, and sequence modeling.

For TGB-Seq, we retain the released split assignments. Final event proportions can differ from 70%/15%/15% after preprocessing. All graph and propagation parameters are selected on validation data before test evaluation.

B.1 Hyperparameter Selection

For the neural baselines evaluated in our experiments, we use the best hyperparameter configurations provided by TGB-Seq and CRAFT (Yi et al., 2025b;a).

We select the configuration of each PPR variant independently using validation MRR. Time-decay PPR uses last or sum aggregation, while Window-based PPR uses binary or count aggregation. Both

share the following propagation and time-scale grids:

$$\alpha \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\},$$
$$K \in \{1, 2, 3, 4, 5, 6, 7, 9, 11, 15, 21, 31\},$$
$$h, W \in \{1\,\mathrm{h}, 1\,\mathrm{d}, 7\,\mathrm{d}, 14\,\mathrm{d}, 30\,\mathrm{d}, 90\,\mathrm{d}, 180\,\mathrm{d}, \infty\}.$$

Here, K is the iteration count L in Section 3.1, and ∞ denotes full-history retention. Without decay, last and sum reduce to binary and count weights, respectively. Table 5 lists the configurations used in the main tables. Unlike the independently selected main-table configurations, the window ablation in Section 5.3 varies the window length while retaining the aggregation and propagation settings of Time-decay PPR.

Table 5: Configurations used for the main-table PPR results.

	Window-based PPR				Time-decay PPR			
Dataset	Aggr.	W	α	K	Aggr.	h	α	K
Wikipedia	count	7 d	0.6	7	last	7 d	0.9	5
Reddit	count	∞	0.6	3	sum	7 d	0.6	3
LastFM	count	30 d	0.8	4	sum	7 d	0.4	4
UCI	count	7 d	0	3	last	1 d	0	3
Flights	count	90 d	0.9	3	sum	30 d	0.8	3
GoogleLocal	binary	∞	0.1	9	sum	∞	0.1	9
YouTube	count	∞	0.05	3	last	14 d	0.05	5
Flickr	binary	90 d	0.1	3	last	7 d	0.1	3
Yelp	binary	∞	0.1	5	last	180 d	0.05	7
Taobao	binary	∞	0.1	5	last	1 d	0.1	5
WikiLink	count	∞	0.2	5	sum	∞	0.2	5
tgbl-review	binary	∞	0	31	sum	90 d	0	31
tgbl-coin	count	30 d	0.7	7	sum	1 d	0.9	9
tgbl-flight	count	30 d	0.9	3	sum	30 d	0.2	1

Historical-state refresh. We use event batches of 2,048 interactions on Taobao, Yelp, and tgbl-coin, and 256 on GoogleLocal; the remaining datasets refresh after each timestamp group. Batches retain complete timestamp groups, and all queries in a batch are scored from its starting history before its positive interactions are added.

Incremental graph maintenance. Our incremental implementation maintains each node pair's event count, latest timestamp, and accumulated exponential weight. For half-life h, let $\lambda = \ln 2/h$ and choose a fixed reference time t_0. The time-decayed weights can be written as

$$w_{uv}(t) = e^{-\lambda(t-t_0)}\widetilde{w}_{uv}, \qquad \widetilde{w}^{\mathrm{last}}_{uv} = e^{\lambda(\tau^{\mathrm{last}}_{uv}-t_0)}, \quad \widetilde{w}^{\mathrm{sum}}_{uv} = \sum_{\tau\in\mathcal{T}_{uv}(t)} e^{\lambda(\tau-t_0)}.$$

The common factor $e^{-\lambda(t-t_0)}$ cancels during row normalization. Algorithm 1 summarizes the update for a selected configuration. It stores log-weights a_{uv} and b_{uv} for last and sum aggregation, respectively; an unobserved pair has both values $-\infty$. Setting $\lambda = 0$ covers the no-decay case. Row normalization uses shifted log-weights for numerical stability.

B.2 Computing Resources

Our experiments were conducted on servers equipped with NVIDIA RTX A6000 and A100 GPUs. Both PPR variants use CPU-based historical graph maintenance and GPU-accelerated sparse matrix operations for finite-step propagation.

Algorithm 1 Incremental historical graph maintenance for Time-decay PPR

Require: Initial history $\mathcal{H}_0$; chronological query batches $\{B_j\}$ retaining complete timestamp groups; aggregation $g \in \{\text{last}, \text{sum}\}$; h, α, K

1: Set $\lambda = \ln 2/h$ (zero if $h = \infty$) and fix reference time t_0
2: Initialize pair log-weights from $\mathcal{H}_0$ using lines 12–13
3: Build the symmetric sparse graph and row-normalized transition matrix P
4: **for** each batch B_j **do**
5: **for** each query $(s, t, \mathcal{C})$ in B_j **do**
6: Score $\mathcal{C}$ by K-step PPR from s using the fixed matrix P
7: **end for**
8: $R \leftarrow \varnothing$; newPair $\leftarrow$ false
9: **for** each observed positive interaction (u, v, τ) in B_j with $u \neq v$ **do**
10: Treat $\{u, v\}$ as an unordered pair
11: newPair $\leftarrow$ newPair $\vee$ [$\{u, v\}$ has not appeared]
12: $a_{uv} \leftarrow \max\{a_{uv}, \lambda(\tau - t_0)\}$
13: $b_{uv} \leftarrow \text{logaddexp}(b_{uv}, \lambda(\tau - t_0))$
14: $R \leftarrow R \cup \{u, v\}$
15: **end for**
16: Use a_{uv} if $g = \text{last}$, otherwise b_{uv}, as each pair's log-weight
17: **if** newPair **then**
18: Rebuild the sparse topology and row-normalize all rows of P
19: **else**
20: Renormalize all incident edge weights in each row $u \in R$ of P
21: **end if**
22: **end for**

C Additional Structural Analysis

Path Discriminability. Figure 7 compares positive and negative candidate coverage by one-, two-, and three-step simple paths across seven datasets. For each path length, positive coverage is the fraction of test queries whose source is connected to the true destination by at least one such path. Negative coverage is the corresponding fraction of negative candidates connected to the query source.

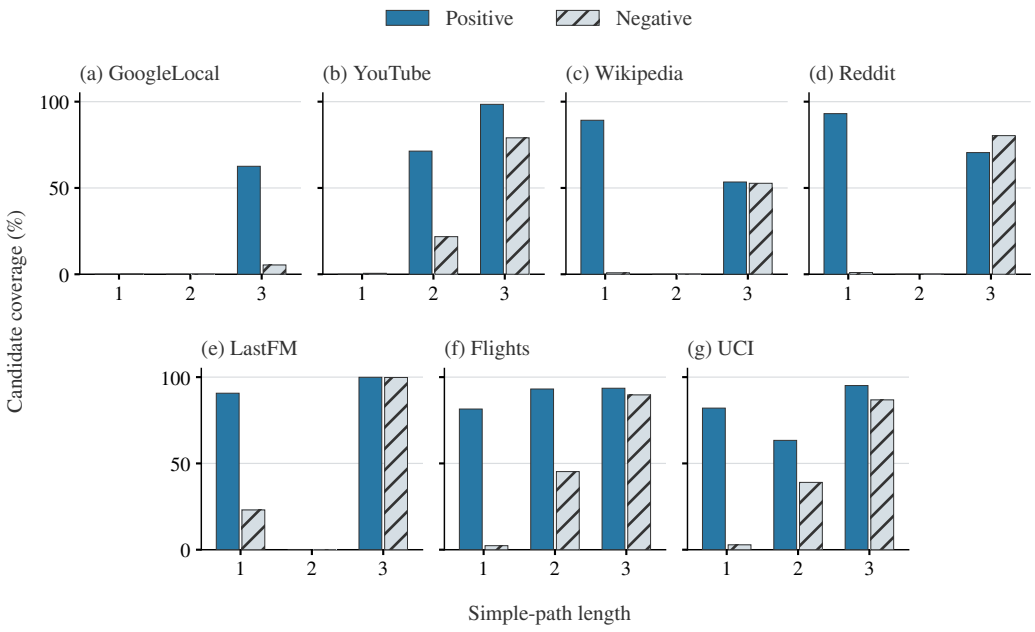

Figure 7: Positive and negative candidate coverage by one-, two-, and three-step simple paths on seven complete test sets.

Wikipedia, Reddit, LastFM, Flights, and UCI exhibit large positive–negative gaps in one-step coverage, indicating that direct historical interactions provide a strong recurrence signal. Longer paths are not uniformly more discriminative: three-step coverage is similar for positives and negatives on Wikipedia, higher for negatives on Reddit, and nearly complete for both on LastFM. In contrast, GoogleLocal has almost no direct historical links but a clear three-step coverage gap. YouTube shows a large two-step gap, while three-step paths reach many negative candidates as well. Flights and UCI also show two-step gaps. These patterns illustrate that useful path lengths depend on the dataset; broader reach alone does not ensure better candidate discrimination. Two-step coverage is zero between opposite partitions of a bipartite graph.

Table 6 further evaluates candidate ranking by two- and three-step simple-path counts on the unweighted historical graphs available at prediction time. Repeated interactions are merged into a single edge, and no temporal weighting is applied. All reported values use complete test sets and the original candidates. Three-step path counts achieve AUCs of 80.14, 90.56, 95.77, and 94.60 on GoogleLocal, YouTube, Flickr, and WikiLink, respectively, showing that historical paths alone can distinguish positive from negative candidates. The useful path length varies across datasets: on Flights, two-step counts outperform three-step counts in both AUC and MRR.

Table 6: Query-averaged ROC-AUC and MRR ($\times 100$) of two- and three-step simple-path counts on the test set.

	2-hop		3-hop	
Dataset	AUC	MRR	AUC	MRR
Bipartite graphs				
Wikipedia	—	—	58.75	18.05
Reddit	—	—	70.59	28.04
LastFM	—	—	53.54	8.66
GoogleLocal	—	—	80.14	48.61
Yelp	—	—	92.38	63.16
Taobao	—	—	86.92	59.54
tgbl-review	—	—	70.47	20.93
Homogeneous graphs				
UCI	66.54	12.58	81.15	21.39
Flights	93.71	67.50	90.69	56.83
YouTube	79.04	36.21	90.56	58.16
Flickr	91.83	60.41	95.77	58.18
WikiLink	88.38	62.45	94.60	66.91
tgbl-flight	79.99	59.73	76.05	51.50

—: omitted because source and destination nodes in opposite partitions of a bipartite graph have no two-step paths. AUC gives half credit to ties; MRR uses average ranks for tied scores.

Structural Evolution Patterns. Figures 3 and 8 show how recurrence and cycle closure vary over time. Each undirected pair is counted once per display interval, at its first event. Using the unweighted full-history graph strictly before that timestamp, pairs are classified in order as repeated, new closing a triangle, new closing a four-cycle, new closing a six-cycle, or none of these. The three closure categories require simple paths of length two, three, and five, respectively. Equal-timestamp events are classified together before graph updates.

Repeated pairs dominate Reddit, LastFM, Flights, tgbl-coin, and tgbl-flight. Among new pairs, four-cycle closure is prominent on Reddit and LastFM, whereas triangle closure is more common on Flights and tgbl-flight; tgbl-coin shows both. Yelp and Taobao instead contain many new pairs closing four-cycles, together with a substantial six-cycle category. On tgbl-review, six-cycle closure is especially prominent, revealing connections through longer historical paths even when shorter closure conditions are absent. WikiLink contains many triangle-closing new pairs, alongside four-cycle closures. Flickr combines frequent triangle closure with a sharp activity burst in which many new pairs close none of the examined cycle types.

D Randomized Reference and Cycle Growth

This appendix expands the analysis of structural evolution in Section 5.1 to help explain why PPR, as a multi-hop structural proximity measure, is effective for temporal link prediction. We compare observed cycle growth with a random ordering of the same edges to identify patterns in how histori-

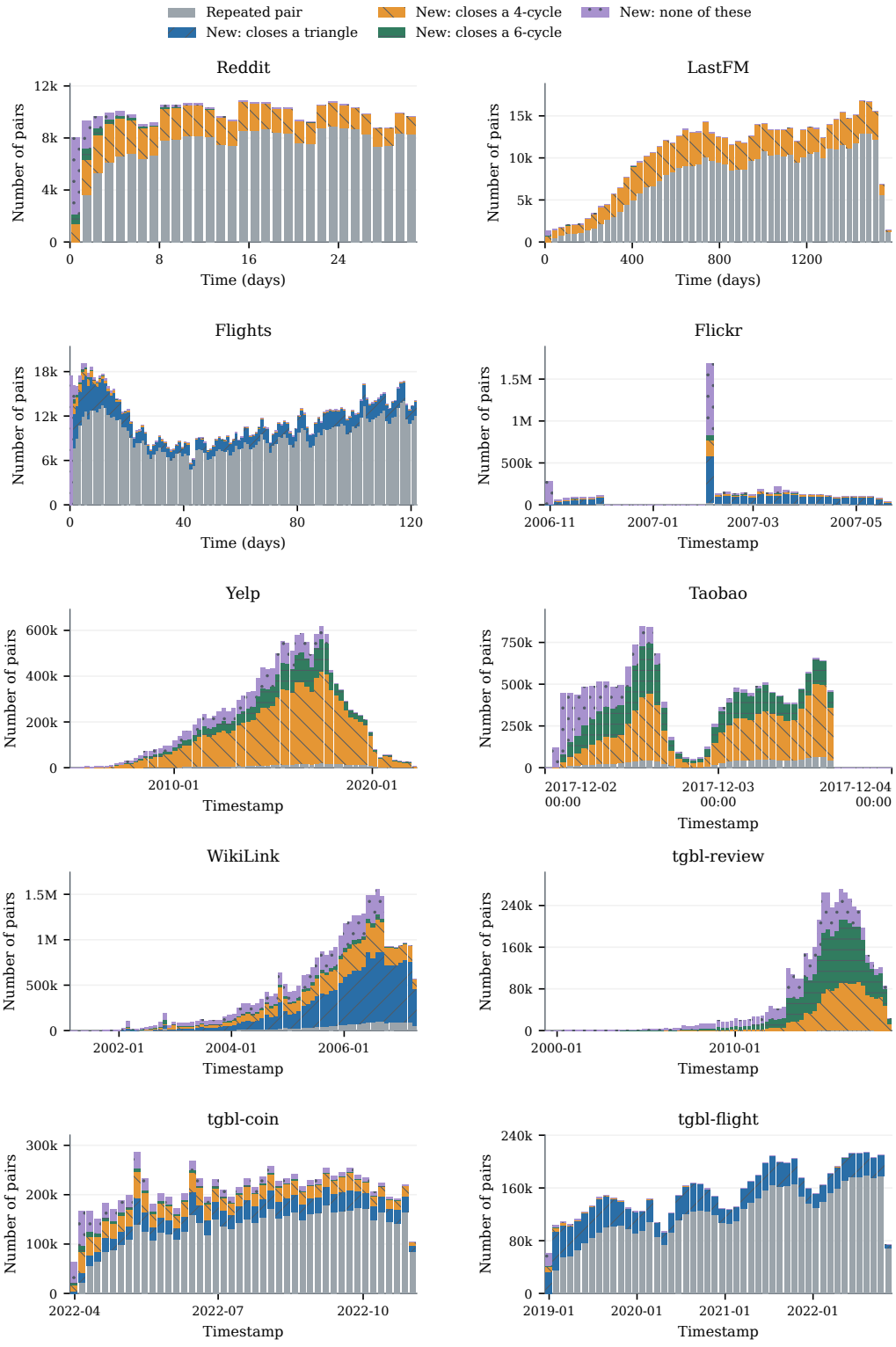

Figure 8: Recurrence and historical cycle closure on the remaining ten datasets, using the categories and counting protocol of Figure 3.

cal connections develop. We present the random-order reference and its derivation, followed by the cycle-growth comparisons and final structural counts.

Paths and cycle closure. A new interaction between previously unconnected nodes closes a k-cycle whenever its endpoints are connected by a $(k-1)$-step simple path. Specifically, for an absent edge $e = \{s, d\}$ in a simple undirected graph G and $k \geq 3$,

$$N_k(G + e) - N_k(G) = S_{k-1}(s, d; G),$$

where $N_k(G)$ counts simple k-cycles and $S_{k-1}(s, d; G)$ counts such paths. Each newly formed cycle corresponds to exactly one existing path between the endpoints. This relation connects cycle formation to the historical multi-hop paths examined in our prediction analysis. However, cycles can also accumulate under random edge arrival, motivating a reference against which to compare the observed growth.

D.1 Random-Order Reference

To examine how edge arrival order shapes cycle formation, we compare the observed sequence with a reference that preserves the final graph while randomizing the order of its edges. Following the constrained-randomization framework of Gauvin et al. (2022), let $G_M = (V, E_M)$ denote the final undirected simple graph constructed from the entire dataset, with repeated interactions merged and self-loops removed. Here, $M = |E_M|$ is the number of distinct edges. A uniformly random permutation of E_M defines the reference sequence, and its first m edges form $\widetilde{G}_m$. We compare the observed and randomized graphs at the same number of distinct edges, so repeated interactions do not advance this measure of graph growth.

Let $N_k(G)$ denote the number of simple k-cycles in G, counting each cycle once regardless of its starting vertex or traversal direction. A cycle in the final graph appears in $\widetilde{G}_m$ exactly when all of its k edges occur among the first m positions. For $m \geq k$, this probability is

$$\Pr(c \subseteq \widetilde{G}_m \mid G_M) = \frac{\binom{M-k}{m-k}}{\binom{M}{m}} = \frac{\binom{m}{k}}{\binom{M}{k}}.$$

Summing the indicator of this event over all final cycles gives

$$\mathbb{E}[N_k(\widetilde{G}_m) \mid G_M] = N_k(G_M)\frac{\binom{m}{k}}{\binom{M}{k}}.$$

This follows from linearity of expectation and does not require independence between cycles. The expectation is zero when $m < k$.

This reference provides a baseline for when cycles emerge within a fixed final topology. Observed counts above the expectation indicate that the real edge ordering forms these cycles earlier in graph growth. The comparison is retrospective: the final graph defines the reference, while prediction uses the available historical graph.

D.2 Cycle Growth Across Datasets

Using this reference, Figure 9 compares observed cycle accumulation with its expectation under random edge ordering, while Figure 10 shows the corresponding differences. Both figures track graph growth by the fraction of distinct edges observed, and normalize cycle counts by their totals in the final graph. Curves above the random-order expectation therefore indicate that a larger fraction of the final cycles has already formed at the same stage of graph growth. Across the examined datasets, four- and six-cycle counts exceed the expectation at all measured intermediate snapshots. Thus, the observed interaction order tends to bring together the edges forming these cycles earlier than a uniformly random ordering would.

The magnitude of this effect varies across datasets and cycle lengths. GoogleLocal exhibits particularly large deviations for both four- and six-cycles, indicating that these structures emerge well before the final graph is fully observed. UCI and Flights also show pronounced excesses across the examined cycle lengths. On YouTube, the deviation is more pronounced for triangles than for

five-cycles, showing that different local structures need not develop in the same way. These observations connect graph evolution to the historical paths studied in our prediction analysis: when a new edge completes a cycle, its endpoints were already connected by an indirect path. The candidate-discriminability analysis in Section 5.1 and Appendix C then examines whether such connections help distinguish future interaction targets from negative candidates.

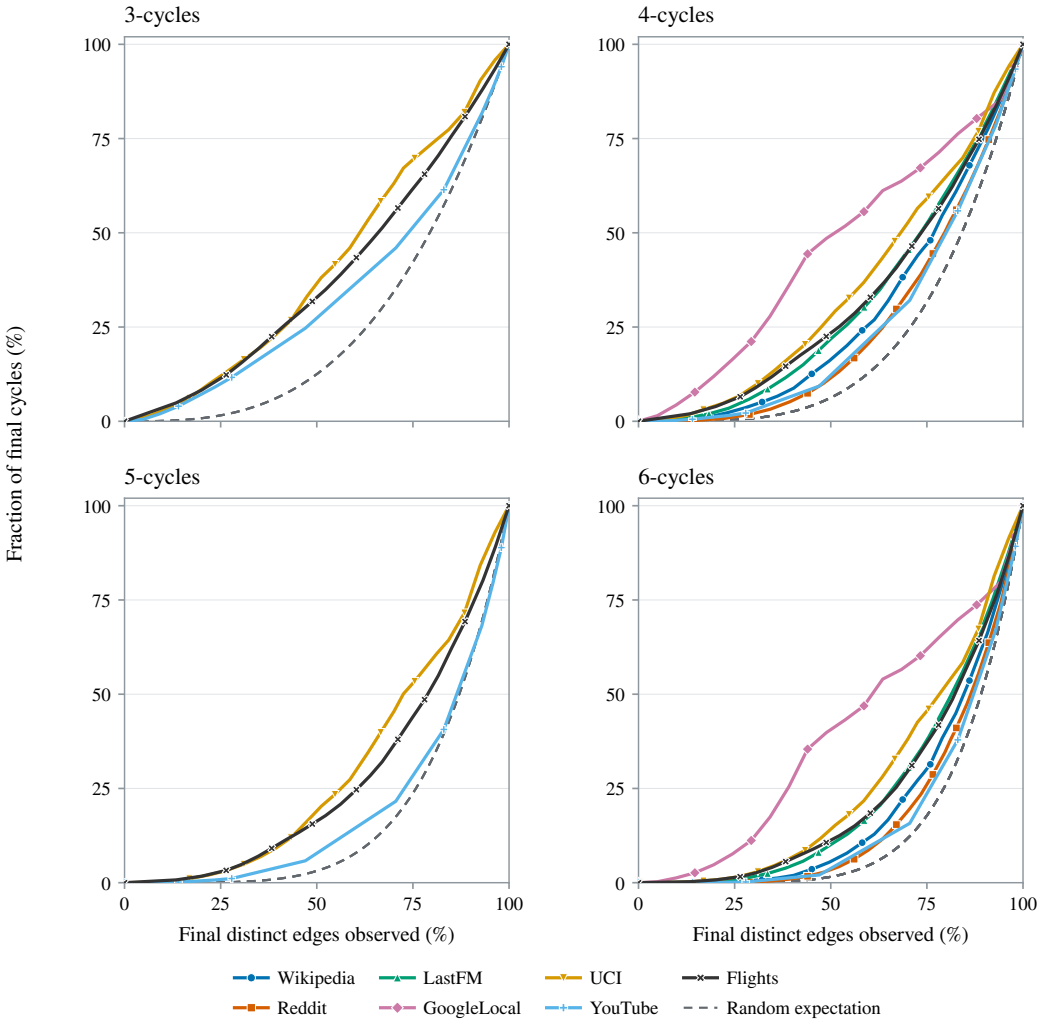

Figure 9: Three- through six-cycle accumulation versus the exact random-order expectation, indexed by distinct-edge coverage.

D.3 Final Structural Counts

Table 7 reports the total number of interactions and the structural counts of the final graph G_M, aggregated over the entire dataset. We merge repeated interactions into undirected edges and remove self-loops. A simple k-cycle contains k distinct vertices and is counted once, regardless of its starting vertex or traversal direction. Counts depend on the aggregated topology, without requiring chronological order among interactions. The reported M and $N_k(G_M)$ provide the final edge and cycle totals used in the preceding growth curves: intermediate counts are normalized by $N_k(G_M)$, and the random-order expectation is evaluated at the same distinct-edge progress m/M.

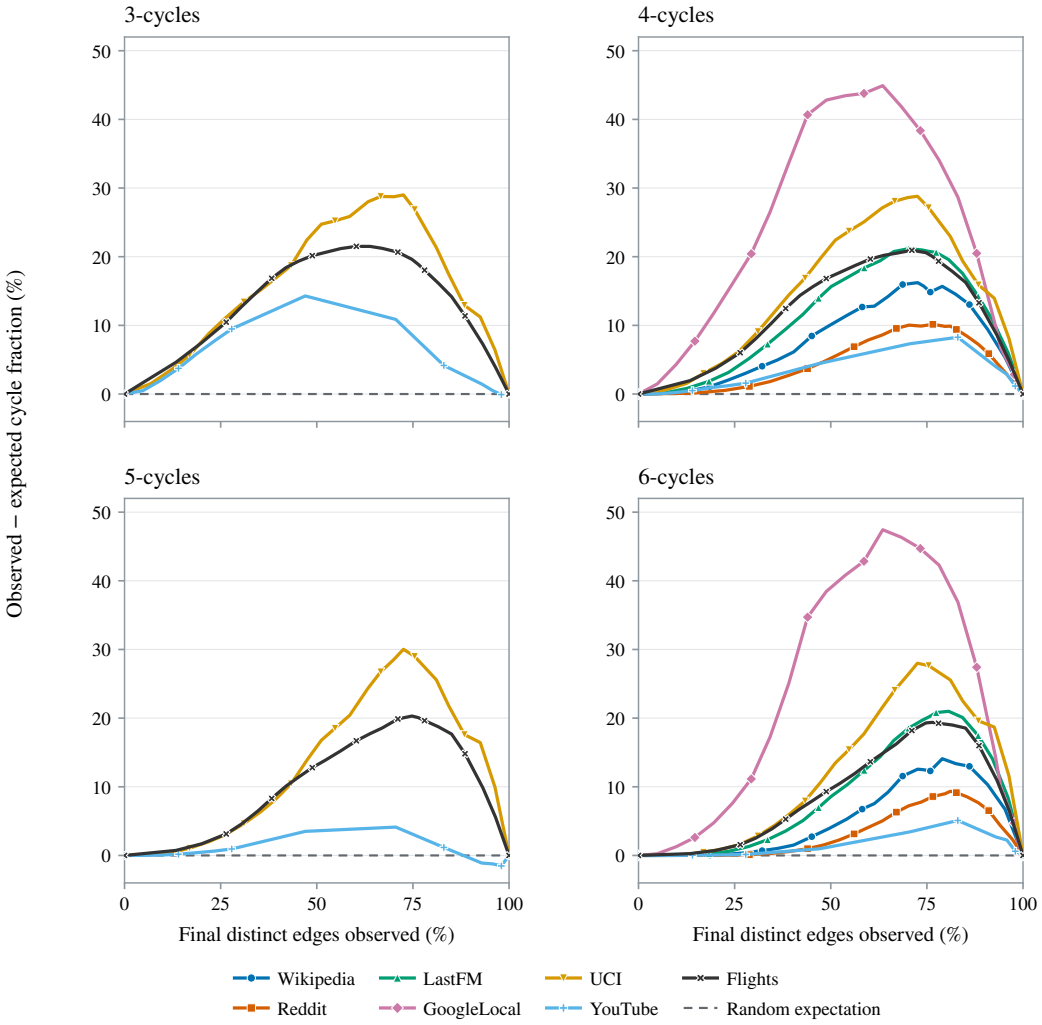

Figure 10: Observed minus expected cycle fractions across datasets. Positive values indicate earlier cycle accumulation than under random edge ordering.

Table 7: Interactions, distinct edges, and simple cycles in the final aggregated graph.

Dataset	Interactions	Distinct edges M	3-cycles	4-cycles	5-cycles	6-cycles
GoogleLocal	1,913,967	1,913,967	0	2.03×10^7	0	3.26×10^{10}
YouTube	3,288,028	3,288,028	6.57×10^6	5.96×10^9	3.90×10^{11}	3.83×10^{14}
Wikipedia	157,474	18,257	0	1.22×10^5	0	3.38×10^7
Reddit	672,447	78,516	0	1.25×10^7	0	4.15×10^{10}
LastFM	1,293,103	154,993	0	5.98×10^8	0	1.85×10^{13}
Flights	1,927,145	295,795	3.94×10^6	5.08×10^8	8.15×10^{10}	1.47×10^{13}
UCI	59,835	13,838	1.43×10^4	7.29×10^5	1.85×10^7	8.76×10^8

E Structural Group Performance Decomposition

We group test queries by how well three-step simple-path counts distinguish the true destination from negative candidates. Each candidate is scored by the number of such paths connecting it to the query source. In the strong-structure group, the true destination has strictly more paths than every negative candidate. Among the remaining queries with at least one path to the true destination, the medium-structure group contains those where it ranks within the top ten, and the weak-structure

group contains those where it ranks below the top ten. Tied candidates receive their average rank. Queries with no three-step simple path to the true destination form the no-three-step-support group.

Using the main-table PPR configurations, we compare PPR and CRAFT on the same test queries and candidate sets. Table 8 reports each group's query share, MRR, and contribution to the overall performance gap. We calculate this contribution by multiplying the within-group MRR difference by the group's share of test queries; these contributions sum to the overall gap. Strong-structure queries account for nearly all of PPR's net gain on GoogleLocal and about half on YouTube.

Table 8: Test MRR $\times 100$ by structural group. Δ is PPR minus CRAFT; contribution is query share times Δ.

Dataset	Group	Share (%)	PPR	CRAFT	Δ	Contribution
GoogleLocal	Strong	38.85	94.63	86.61	+8.02	+3.12
GoogleLocal	Medium	22.37	59.09	57.43	+1.65	+0.37
GoogleLocal	Weak	1.34	23.11	32.42	-9.31	-0.12
GoogleLocal	No 3-step	37.44	41.65	42.24	-0.60	-0.22
GoogleLocal	All	100.00	65.89	62.75	+3.14	+3.14
YouTube	Strong	49.13	96.99	85.48	+11.51	+5.66
YouTube	Medium	25.83	62.18	44.83	+17.35	+4.48
YouTube	Weak	23.49	27.47	21.71	+5.76	+1.35
YouTube	No 3-step	1.55	25.29	25.63	-0.34	-0.01
YouTube	All	100.00	70.56	59.07	+11.49	+11.49

F Sensitivity Settings and Additional Results

Figures 11 and 12 report validation MRR across half-life, aggregation, restart probability, and propagation-depth sweeps. Each sweep varies the indicated parameters while keeping the others at the Time-decay PPR settings in Table 5. Half-life and K use ordered grids, with the no-decay endpoint separated. Circles and squares denote odd and even K, and a diamond denotes $\alpha = 0$.

For the window ablation in Table 3, we select the window length using validation MRR while retaining the aggregation and propagation settings of Time-decay PPR. Table 9 reports the selected windows and the mean fraction of historical events they exclude.

Table 9: Selected windows and mean fractions of excluded historical events.

Dataset	Window	Excluded events
Wikipedia	7d	68.47%
Reddit	90d	0.00%
LastFM	30d	96.93%
UCI	7d	90.92%
Flights	90d	13.53%
YouTube	90d	31.95%
Flickr	90d	18.92%
tgbl-coin	30d	79.49%

Temporal retention. Table 9 shows that the selected windows retain very different amounts of history across datasets. On Reddit, the selected window covers the entire available history, whereas on LastFM and UCI it excludes most historical events. A window length therefore does not correspond to a consistent information budget across datasets. Figure 11 further shows different responses to the decay time scale: UCI favors a short half-life and deteriorates as older interactions receive more weight, while Reddit and Flights maintain relatively stable performance over a broader range of longer half-lives. YouTube and Flickr perform poorly with a one-hour half-life but improve substantially when the time scale extends to days or weeks. These differences suggest that useful historical connections persist over different time scales across datasets.

Interaction between retention and aggregation. The effect of repeated-interaction aggregation also depends on temporal retention. On Wikipedia, last-decay outperforms sum-decay at every eval-

uated finite half-life, but this ordering reverses without decay. On LastFM, the two aggregation rules perform similarly at shorter half-lives, whereas sum aggregation performs substantially better in the full-history setting. These results show that the value of accumulating repeated interactions cannot be assessed independently of how their ages are weighted. By contrast, the nearly overlapping curves on YouTube and Flickr indicate that temporal weighting has a much larger effect than the choice between last and sum aggregation on these datasets.

Propagation depth and restart. Figure 12 shows that the benefit of additional propagation depends on the predictive information already available locally. Wikipedia and Reddit obtain further gains when moving from one to three steps, despite their strong one-step performance. YouTube and Flickr rely much more heavily on indirect connections, with substantial improvements over one-step propagation and their strongest results at shallow multi-step depths. Increasing depth further yields little benefit or reduces performance. Restart preferences also differ: larger restart probabilities perform well on Wikipedia and Flights, whereas YouTube and Flickr favor smaller values. Together, these patterns suggest that both the range of propagation and the balance between shorter and longer walks should reflect the dataset's interaction patterns.

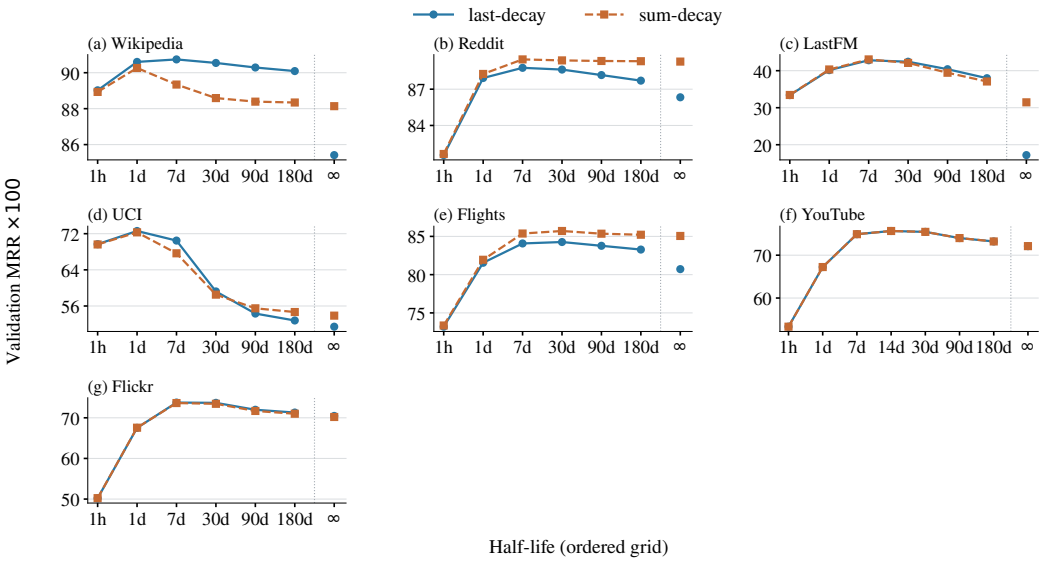

Figure 11: Half-life and aggregation sensitivity on all seven datasets. Validation MRR $\times 100$; panel-specific vertical scales.

G Additional Baseline and Candidate Analyses

G.1 Additional Baselines

We compare Time-decay PPR with seven training-free baselines to examine how much predictive information can be obtained from direct interaction history and destination statistics.

Pair-based baselines. EdgeBank assigns a binary score indicating whether a source–destination pair has previously interacted (Poursafaei et al., 2022). We evaluate full-history memory, EdgeBank$_\infty$, and windowed memory, EdgeBank$_{\text{tw}}$. Following the paper's window setting, the latter retains interactions within a window whose length equals the validation time span. Pair-recency scores each candidate by its most recent interaction time with the query source, while pair-frequency uses their cumulative interaction count (Cornell et al., 2025). These baselines preserve the source and destination roles of the original events.

Destination-based baselines. Destination-recency scores a candidate by its latest occurrence as a destination, irrespective of the query source. Destination-frequency counts all such occurrences

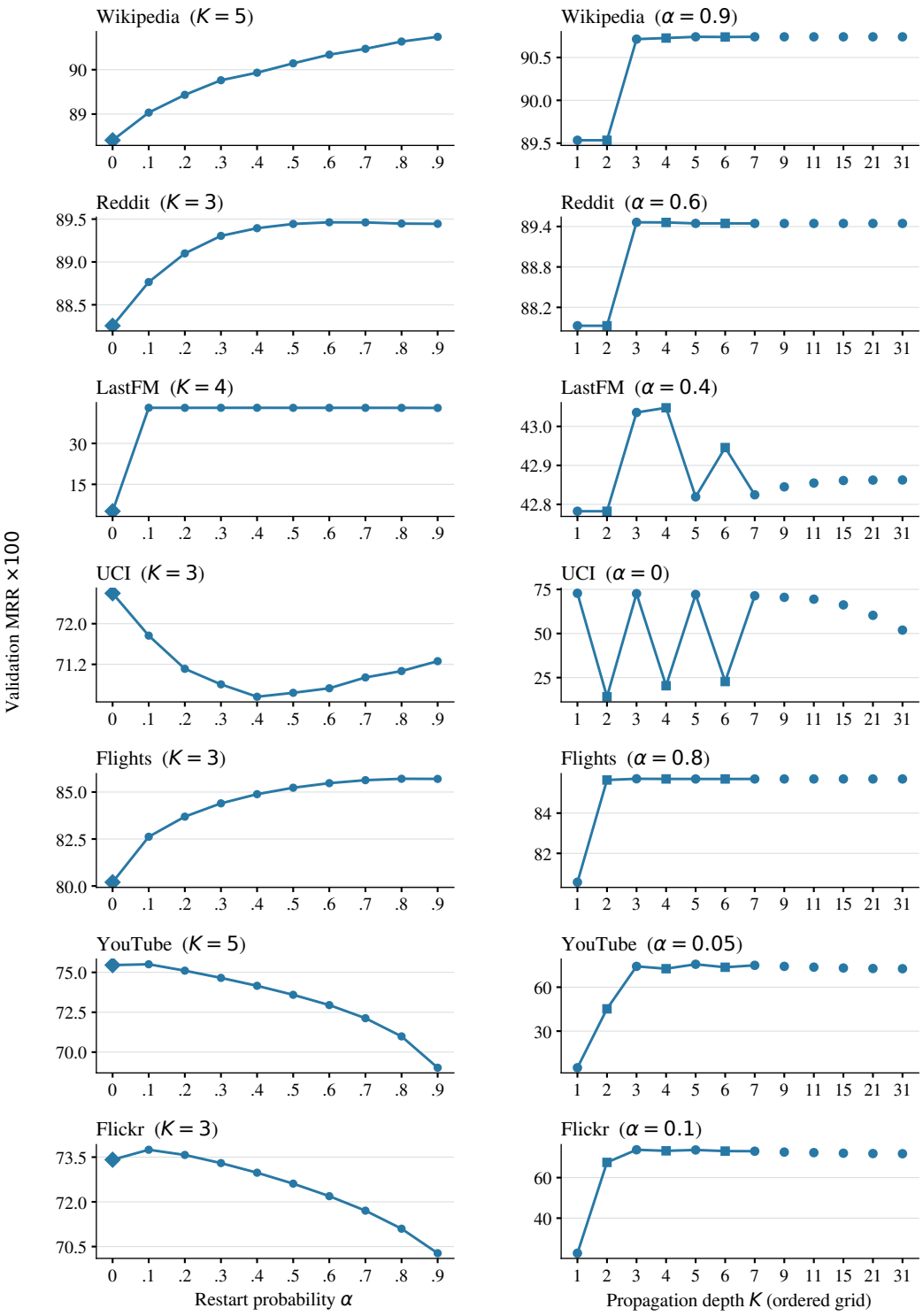

Figure 12: Sensitivity to α and K on seven datasets, with other parameters fixed. Validation MRR $\times 100$.

(Cornell et al., 2025). Time-aware popularity baselines have also been studied in recommendation and temporal graphs (Ji et al., 2020; Daniluk & Dabrowski, 2023). Destination-activity instead weights them by recency:

$$a_h(v,t) = \sum_{(u,v,\tau)\in\mathcal{E}:\,\tau<t} 2^{-(t-\tau)/h}, \tag{7}$$

where h is the half-life. Thus, destination-frequency measures cumulative activity, whereas destination-activity emphasizes recent interactions. Candidates without relevant history receive zero for count-based scores and negative infinity for recency scores.

Evaluation. We evaluate all heuristics on the 14 main-table datasets using the original test candidate sets. Scores use only interactions strictly before the query timestamp, with history updated from observed positive events. The activity half-life is selected by validation MRR from $\{1\text{h}, 1\text{d}, 7\text{d}, 14\text{d}, 30\text{d}, 90\text{d}, 180\text{d}, \infty\}$. The other baselines require no parameter search. Time-decay PPR results use the configurations and refresh settings of the main tables.

Results. Table 10 reports test MRR. Pair-recency and pair-frequency substantially improve upon binary edge memory on several seen-dominant datasets. On GoogleLocal, YouTube, and Flickr, destination activity provides stronger predictions than pair-based statistics, reaching MRRs of 20.01, 58.74, and 44.00, respectively, compared with 65.89, 70.56, and 71.93 for Time-decay PPR. On tgbl-review, destination activity reaches 41.58, exceeding Time-decay PPR and approaching the strongest reported neural baseline at 41.77. These comparisons illustrate the different predictive roles of relation recurrence and destination activity, alongside the broader competitiveness of historical structural proximity.

Table 10: Test MRR $\times 100$ of heuristic baselines on the original candidate sets. Time-decay PPR results follow the main tables.

	Unseen-dominant						
Method	GoogleLocal	Yelp	Taobao	YouTube	Flickr	WikiLink	tgbl-review
EdgeBank$_\infty$	1.96	9.77	20.28	1.95	1.96	1.96	1.89
EdgeBank$_{\text{tw}}$	1.96	5.89	14.62	1.96	1.96	1.96	2.05
Pair-recency	1.96	9.79	20.29	1.95	1.96	1.96	1.97
Pair-frequency	1.96	9.78	20.29	1.95	1.96	1.96	1.89
Destination-recency	13.41	24.78	20.72	34.57	20.78	30.55	33.01
Destination-frequency	18.39	24.79	30.03	51.31	32.37	47.50	28.06
Destination-activity	20.01	34.06	30.22	58.74	44.00	48.54	**41.58**
Time-decay PPR	**65.89**	**72.28**	**71.81**	**70.56**	**71.93**	**86.16**	37.03

	Seen-dominant						
Method	Wikipedia	Reddit	LastFM	UCI	Flights	tgbl-coin	tgbl-flight
EdgeBank$_\infty$	79.56	77.89	14.79	42.57	54.46	35.99	16.69
EdgeBank$_{\text{tw}}$	81.03	78.46	18.59	53.77	58.98	57.36	37.57
Pair-recency	87.49	87.22	43.62	66.87	75.91	78.23	83.24
Pair-frequency	84.84	87.79	36.72	56.51	77.97	70.07	87.23
Destination-recency	48.97	35.51	18.92	50.26	5.19	72.63	18.31
Destination-frequency	12.22	47.28	7.70	17.92	40.59	61.30	60.27
Destination-activity	42.15	48.10	21.91	50.60	41.57	72.95	63.46
Time-decay PPR	**89.17**	**89.61**	**46.66**	**78.80**	**86.15**	**81.29**	**89.45**

EdgeBank$_{\text{tw}}$ uses a fixed validation-duration window; activity uses validation-selected decay. **First**, second, and third are marked per dataset. PPR results are those in the main comparison.

G.2 Activity-weighted Negative Sampling

Sampling strategy. We retain the original test queries and positive destinations. For each query, we sample 100 distinct negative destinations without replacement, with each draw weighted by the

remaining candidates' historical destination-activity scores defined in Appendix G.1. We exclude all positive destinations associated with the same source and query timestamp. Sampling uses only historical interactions, and the resulting candidate sets are shared across methods and training seeds.

Models and datasets. We compare Time-decay PPR with JODIE, DyRep, TGN, GraphMixer, DyGFormer, and CRAFT. Table 11 reports test MRR under activity-weighted negative sampling on Wikipedia, Flights, GoogleLocal, YouTube, and Flickr.

Table 11: Test MRR $\times 100$ ($\uparrow$) under activity-weighted negative sampling (mean $\pm$ standard deviation over three seeds).

Method	Wikipedia	Flights	GoogleLocal	YouTube	Flickr
JODIE	66.39±0.51	11.06±2.00	25.63±0.12	5.88±0.78	17.60±1.47
DyRep	62.39±1.66	5.54±0.24	22.29±0.28	5.03±0.21	14.34±0.34
TGN	77.47±0.92	30.57±1.46	37.35±0.18	11.04±1.34	19.20±1.03
GraphMixer	60.93±0.17	11.95±0.28	7.01±0.05	4.84±0.15	4.82±0.18
DyGFormer	85.16±0.08	60.89±0.21	4.77±0.20	5.73±0.50	22.46±3.11
CRAFT / CRAFT-R	83.40±0.12	56.65±0.58	**49.54**±0.50	10.83±0.56	29.12±0.32
Time-decay PPR	**85.38**±0.00	**60.92**±0.00	47.96±0.00	**14.04**±0.00	**33.43**±0.00

CRAFT-R is used on Wikipedia and Flights; CRAFT is used on GoogleLocal, YouTube, and Flickr.

Results and analysis. Time-decay PPR remains competitive with the evaluated neural baselines under activity-weighted negative sampling. These results extend the comparison beyond the original benchmark candidates and show that historical structural proximity remains useful when negative candidates are drawn more frequently from recently active destinations. Together with the main results, this supports PPR as a competitive reference under different candidate distributions. The sensitivity of results to negative sampling further motivates evaluation protocols that assess model performance across multiple candidate distributions, providing a more comprehensive view of predictive capabilities.

H Implications for Future Research

Our goal is not to suggest that structural proximity alone is sufficient for temporal link prediction or that PPR should replace learned models. Instead, PPR provides a simple, strong reference for understanding how far historical structural proximity can go and identifying where learned models offer additional benefits. Our findings motivate combining learned temporal representations with historical structural proximity, while adapting temporal retention and repeated-interaction aggregation to different interaction patterns. They also encourage evaluation under multiple negative sampling strategies to better characterize model strengths and limitations and guide future progress.