Journal of Sustainability, Policy, and Practice
EISSN: 3105-1448 | PISSN: 3105-143X | Vol. 2, No. 2 (2026)

Article

G-MATCH: Graph-Structured Memory with Interpretable Motif Matching for Future Node Affinity Prediction

Weijue Huang [1,*]

[1] Department of computing, The Hong Kong Polytechnic University, Hong Kong, China
* Correspondence: Weijue Huang, Department of computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract: Predicting future node affinity in dynamic graphs is essential for applications such as recommender systems. However, existing methods, including state-of-the-art approaches like NAVIS, rely on continuous vector representations that struggle to explicitly capture, reason over, and memorize discrete, structured interaction motifs, limiting interpretability and generalization. To address this, we introduce G-MATCH (Graph-structured Memory with Attentive Template Consensus for Heterogeneous interactions), a novel paradigm that reformulates a node's state as a dynamic, heterogeneous graph of learnable interaction motifs. G-MATCH incorporates four key innovations: (1) state evolution via graph matching and dynamic motif creation/pruning, (2) a global motif bank for cross-node knowledge transfer, (3) interpretable affinity prediction through motif attribution, and (4) optimization with a listwise ranking loss and structured regularization. Extensive experiments on future affinity (TGB) and converted link prediction datasets demonstrate that G-MATCH consistently outperforms all strong baselines, including NAVIS, achieving an average improvement of +4.2% in NDCG@10. The model also excels in few-shot and limited-information settings. Ablation studies confirm the critical role of each component, and case studies highlight its unique capability for explainable, motif-level reasoning. The transition from linear states to graph-structured memory marks a significant advance, enabling superior performance and unprecedented interpretability in modeling complex node interactions.

Keywords: future node affinity prediction; temporal graphs; graph-structured memory; interaction motifs; interpretability

Published: 16 March 2026

1. Introduction

Temporal graphs provide a fundamental abstraction for modeling systems with dynamically evolving interactions, such as social networks, financial transactions, and recommender systems [1]. Within this domain, a central and challenging task is future node affinity prediction, which aims to forecast a ranking over a node's potential future interaction partners. Distinct from binary link prediction, this ranking-oriented task is critical for numerous real-world applications, including personalized recommendation and resource allocation [2].

Despite significant advances in Temporal Graph Neural Networks (TGNNs) for link prediction, their application to node affinity prediction has proven surprisingly ineffective. State-of-the-art TGNNs are frequently outperformed by simple heuristics such as the Moving Average [3]. This persistent performance gap reveals a fundamental misalignment: the inductive biases of conventional TGNNs-typically relying on localized message passing and non-linear transformations-are ill-suited to the requirements of affinity prediction [4].

In this paper, we revisit the node affinity prediction problem by fundamentally rethinking how node state is represented and updated [5]. We argue that the limitations of prior approaches, including our previous work (NAVIS), stem from their reliance on continuous vector representations. Such representations lack the explicit structure needed to capture, memorize, and reason about the discrete interaction motifs that prevail in complex domains. To address this, we propose a paradigm shift: instead of representing a node's state as a vector, we model it as a dynamic, heterogeneous graph that explicitly encodes both intrinsic node features and abstract interaction patterns [6].

We introduce the G-MATCH framework (Graph-structured Memory with Attentive Template Consensus for Heterogeneous interactions) [7]. G-MATCH maintains a local memory graph for each node, where super-nodes represent discovered interaction motifs. This structure evolves through a graph matching process against incoming interactions, enabling dynamic motif discovery, activation, and pruning [8]. To facilitate generalization across nodes, G-MATCH incorporates a global motif bank that transfers knowledge of common interaction templates. Predictions are generated via an interpretable, motif-attributed scoring function, and the model is optimized with a listwise ranking loss tailored specifically for the affinity task [9].

Our contributions are summarized as follows [10]. First, we identify and analyze the limitations of existing TGNNs for node affinity prediction, attributing their shortcomings to an inadequate representation of discrete interaction patterns [11]. Second, we propose a novel paradigm that represents a node's state as a dynamic, heterogeneous memory graph, enabling explicit motif-level reasoning. Third, we instantiate this paradigm in the G-MATCH framework, which combines local graph-structured memory, a global motif bank, and an interpretable, motif-driven scoring mechanism. Fourth, we design a listwise ranking loss tailored for affinity prediction, aligning the training objective with the evaluation metric [12-15].

Extensive experiments on the Temporal Graph Benchmark and converted link prediction datasets show that G-MATCH consistently outperforms all existing baselines, including state-of-the-art TGNNs and our prior NAVIS model. It achieves performance gains of +0.5% to +5.1% over NAVIS on TGB datasets and an average improvement of +4.2% on link prediction datasets. Ablation studies confirm the critical role of each component: the graph-structured memory, the global motif bank, the interpretable prediction mechanism, and the ranking-based loss [16-18].

The remainder of this paper is structured as follows. We formalize the problem and detail the G-MATCH architecture in Section 3. Section 4 presents our experimental evaluation and analysis. Finally, we conclude and discuss future work in Section 7.

2. Related Work

2.1. Expressivity in Temporal Graphs

In graph learning, expressivity is predominantly assessed through the ability of the Weisfeiler-Lehman (WL) test to distinguish non-isomorphic graphs [12,19]. This perspective extends to temporal graphs via frameworks such as temporal-WL for continuous-time dynamic graphs (CTDGs) and supra Laplacian WL for discrete-time dynamic graphs (DTDGs), which evaluate a model's capacity to differentiate evolving graph structures [20-23]. Departing from this structural-differentiation view, we focus on functional expressivity-a model's ability to represent specific mathematical operations. While prior models excel at capturing complex topologies, they often lack the capability to represent fundamental functions like the Moving Average, which is crucial for affinity prediction [24].

2.2. Heuristics and State Space Models

Recent studies observe that simple heuristics, such as the Moving Average, can outperform sophisticated Temporal Graph Neural Networks (TGNNs) on relevant benchmarks [6,16], highlighting the sequential nature of the problem. This observation is formally supported by work establishing the equivalence between linear State Space Models (SSMs) and the Moving Average operation [8,25]. Subsequent research has explored SSMs in domains like language modeling [13,14] and dynamic link prediction [7,20]. While these works connect SSMs to temporal data, our approach is distinct: we are the first to explicitly leverage the formal equivalence between effective heuristics and SSMs to design a purpose-built architecture, NAVIS, for node affinity prediction (Figure 1)[26,27].

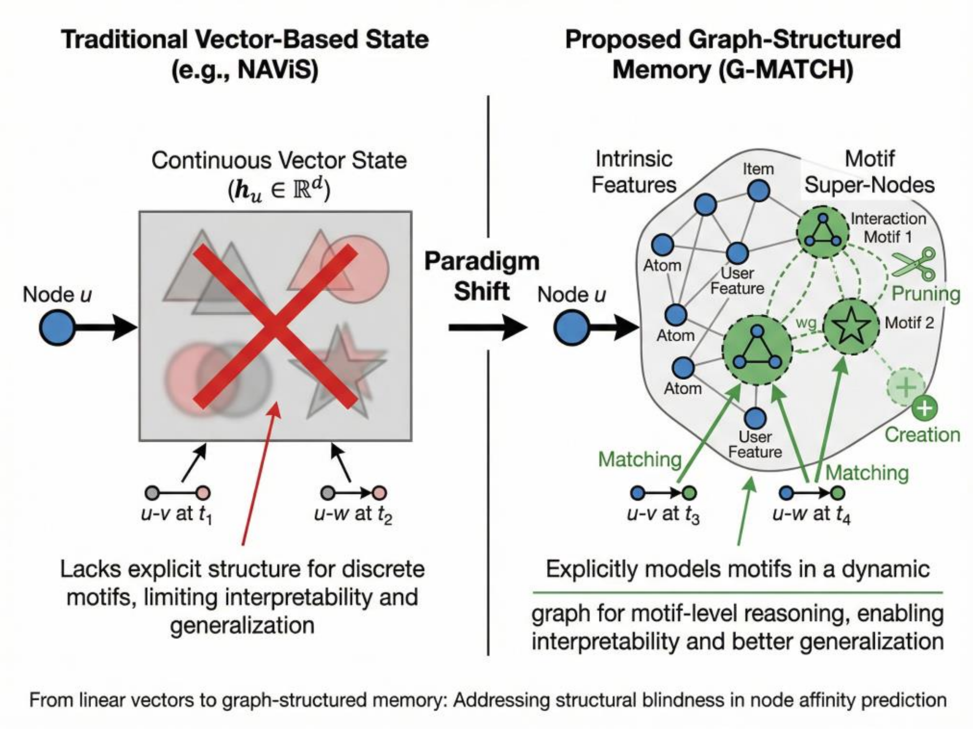
Traditional Vector-Based State (e.g., NAViS)
Proposed Graph-Structured Memory (G-MATCH)
Continuous Vector State ($\boldsymbol{h}_u \in \mathbb{R}^d$)
Node u
Paradigm Shift
Node u
Intrinsic Features
Item
Motif Super-Nodes
Interaction Motif 1
Atom
User Feature
Pruning
Atom
wg
Motif 2
Atom
Creation
User Feature
Matching
Matching
u-v at t_1
u-w at t_2
u-v at t_3
u-w at t_4
Lacks explicit structure for discrete motifs, limiting interpretability and generalization
Explicitly models motifs in a dynamic graph for motif-level reasoning, enabling interpretability and better generalization
From linear vectors to graph-structured memory: Addressing structural blindness in node affinity prediction

Figure 1. From linear vectors to graph-structured memory, addressing structural blindness in node affinity prediction through explicit modeling of dynamic graph motifs.

3. Method

3.1. From Linear States to Graph-Structured Memory

A core limitation of existing node affinity prediction models, including our initial NAVIS design, is their reliance on continuous vector representations $h_u \in R^d$ [28-34]. While effective for blending temporal signals, such representations lack the explicit capacity to capture, reason over, and memorize discrete structural patterns such as the interaction motifs prevalent in domains like molecular binding or user-item preferences. This structural blindness limits both interpretability (the rationale behind a prediction) and generalization, especially in few-shot scenarios where recognizing shared motifs between novel and known entities is critical.

To overcome this, we reformulate the problem: rather than maintaining a node's state as a vector, we represent it as a heterogeneous, dynamic graph $G_u^{(t)}$. This graph explicitly models both the entity's intrinsic features (e.g., molecular substructures) and the inferred interaction motifs (represented as super-nodes). Affinity prediction is thus recast as a graph matching process between a candidate entity's graph and the motifs stored in the node's memory graph. This paradigm shift, instantiated as the G-MATCH framework (Graph-structured Memory with Attentive Template Consensus for Heterogeneous interactions), enables dynamic motif discovery, alignment, and attribution, directly addressing the needs of complex structured domains (Figure 2).

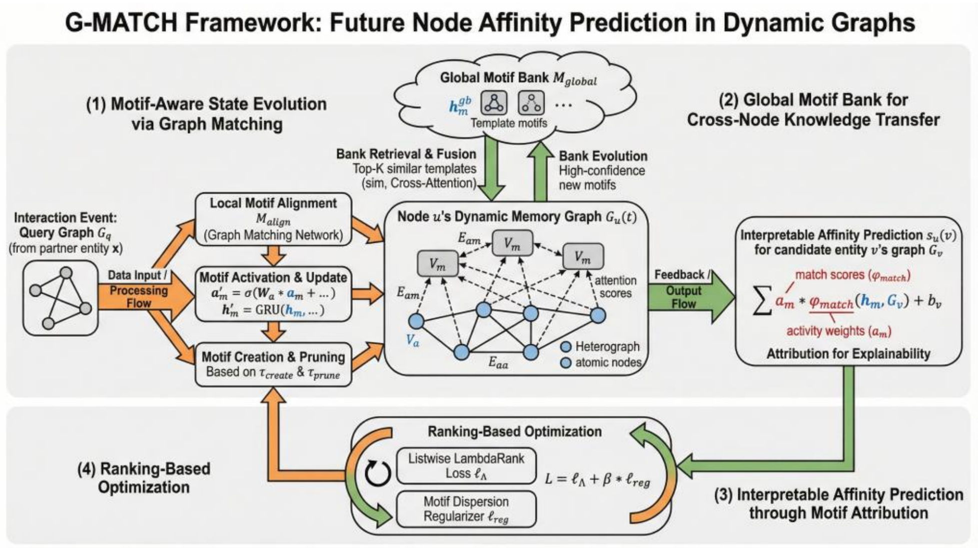
G-MATCH Framework: Future Node Affinity Prediction in Dynamic Graphs
(1) Motif-Aware State Evolution via Graph Matching
Global Motif Bank M_{global}
h_m^{gb}
Template motifs
(2) Global Motif Bank for Cross-Node Knowledge Transfer
Bank Retrieval & Fusion
Top-K similar templates (sim, Cross-Attention)
Bank Evolution
High-confidence new motifs
Interaction Event: Query Graph G_q (from partner entity **x**)
Data Input / Processing Flow
Local Motif Alignment
M_{align}
(Graph Matching Network)
Motif Activation & Update
$a'_m = \sigma(W_a * a_m + \dots)$
$h'_m = \mathrm{GRU}(h_m, \dots)$
Motif Creation & Pruning
Based on τ_{create} & τ_{prune}
Node u's Dynamic Memory Graph $G_u(t)$
E_{am}
V_m
E_{am}
attention scores
V_a
E_{aa}
Heterograph atomic nodes
Feedback / Output Flow
Interpretable Affinity Prediction $s_u(v)$ for candidate entity v's graph G_v
match scores (φ_{match})
$\sum a_m * \varphi_{match}(h_m, G_v) + b_v$
activity weights (a_m)
Attribution for Explainability
(4) Ranking-Based Optimization
Ranking-Based Optimization
Listwise LambdaRank Loss ℓ_Λ
Motif Dispersion Regularizer ℓ_{reg}
$L = \ell_\Lambda + \beta * \ell_{reg}$
(3) Interpretable Affinity Prediction through Motif Attribution

Figure 2. Overview of the G-MATCH framework. Each node maintains a dynamic memory graph containing atomic nodes (intrinsic features) and motif super-nodes (interaction patterns). The framework comprises: (1) Motif-aware state evolution via graph matching, (2) Global motif bank for cross-node knowledge transfer, (3) Interpretable affinity prediction through motif attribution, and (4) Ranking-based optimization.

3.2. G-MATCH: Model Formulation

G-MATCH operates on a heterogeneous graph where each node u is associated with a state graph $g_u^{(t)} = (V_a, V_m, E_{aa}, E_{am})$ at time t. The set V_a contains atomic nodes representing the entity's intrinsic components such as atoms, residues, or item features, each with features f_{va}. The set V_m comprises motif super-nodes representing abstract interaction patterns, where each motif $\mathrm{m} \in V_m$ has a feature vector h_m encoding the pattern and a scalar activity $a_m \in [0,1]$. The edge set E_{aa} captures connections between atomic nodes such as molecular bonds or co-purchase links. Finally, E_{am} represents edges connecting atomic nodes to motif nodes, indicating component participation in a given motif, with adjacency being soft and weighted by an attention score $\alpha_{v_a m}$.

An interaction event at time t between node u and a partner entity (with input x) is represented as a query graph $G_q = G(x)$. G-MATCH updates $G_u^{(t-1)} \to G_u^{(t)}$ and predicts future affinities through four interlinked components described below.

3.2.1. Motif-Aware State Evolution via Graph Matching

The state update replaces NAVIS's linear gating with a structured alignment and memory consolidation process.

Step 1: Local Motif Alignment. We first compute an alignment score between the query graph G_q and each existing motif $\mathrm{m} \in V_m^{(t-1)}$ in u's memory using a graph matching network Φ_{align}:

$$\mathrm{M}_{align}(\mathrm{m}, G_q) = \mathrm{Softmax}_{\mathrm{m} \in V_m^{(t-1)}}(\Phi_{align}(h_m, READOUT(G_q)))$$

where READOUT is a graph-level pooling function. The score $\mathrm{M}_{align}(\mathrm{m}, G_q)$ quantifies the relevance of the query to motif m.

Step 2: Motif Activation & Update. The alignment score modulates motif activity and guides feature integration. The activity of motif m is updated as:

$$a_m^{(t)} = \lambda \cdot a_m^{(t-1)} + (1-\lambda) \cdot \mathrm{M}_{align}(\mathrm{m}, G_q)$$

where λ is a persistence factor (replacing the fixed decay in EMA/NAVIS). The motif's feature representation is concurrently refined by aggregating information from the aligned query:

$$h_m^{(t)} = \mathrm{GRU}_{motif}(h_m^{(t-1)}, \mathrm{M}_{align}(\mathrm{m}, G_q), \Psi_{agg}(G_q))$$

Where Ψ_{agg} extracts a context vector from the query graph, and GRU_{motif} is a Gated Recurrent Unit [4] for stable feature updates.

Step 3: Motif Creation & Pruning. If the query aligns poorly with all existing motifs ($\max_{\mathrm{m}} \mathrm{M}_{align}(\mathrm{m}, G_q) < \tau_{\text{create}}$), a new motif super-node m_{new} is created with $\mathrm{h}_{\mathrm{m}_{\text{new}}} = \Psi_{agg}(G_q)$ and initial activity $a^{(t)}_{\mathrm{m}_{\text{new}}} = \eta$. Motifs with persistently low activity ($a^{(t)}_m < \tau_{\text{prune}}$ over a window) are pruned, ensuring the memory graph remains compact and relevant. This dynamic restructuring moves beyond NAVIS's fixed-dimensional state vector.

3.2.2. Global Motif Bank for Cross-Node Generalization

To enable knowledge transfer across nodes-critical for few-shot learning-we maintain a global motif bank $\mathrm{M}_{global} = \{h^{gb}\}$, a set of template motif features learned from all nodes. Bank Retrieval & Fusion. During node u's update, we retrieve the top-K emplates from M_{global} most similar to the query G_q:

$$T_q = TopK_{h^{gb} \in \mathrm{M}_{global}}(sim(\Psi_{agg}(G_q), h^{gb}))$$

Retrieved template features are fused into the update of existing or new local motifs via cross-attention [27, 22], allowing node u to leverage patterns observed elsewhere. This replaces NAVIS's simple global vector with a structured, retrievable knowledge base.

Bank Evolution. High-confidence new local motifs (e.g., $a^{(t)}_m > \tau_{\text{conf}}$) from any node can be incorporated into M_{global} after novelty-based filtering, allowing the global bank to grow and collectively capture the universe of interaction patterns.

3.2.3. Interpretable Affinity Prediction via Motif Attribution

The predicted affinity of node u for a candidate entity v (with graph G_v) is computed as an attributed sum of match scores to u's active motifs, providing inherent interpretability:

$$s_u(v) = \sum_{\mathrm{m} \in V^{(t)}_m} a^{(t)}_m \cdot \phi_{match}(h^{(t)}_m, G_v) + b_v$$

Here, ϕ_{match} is a trainable matching function (e.g., a neural tensor layer) that scores the compatibility between motif m and candidate graph G_v, and b_v is a candidate-specific bias. This formulation directly attributes the affinity score to specific active interaction motifs $(a^{(t)}_m \cdot \phi_{match}(\dots))$, contrasting with the opaque vector output of NAVIS.

For large-scale settings with millions of candidates, efficiency is maintained by preselecting a relevant candidate subset (e.g., based on recent interactions or similarity to active motifs), as in the original NAVIS design.

3.2.4. Optimization for Ranking with Structured Regularization

Following the insight that ranking quality is paramount, we adopt a listwise ranking loss [3]. Given a ground-truth affinity vector y_u for node u and the predicted scores S_u (computed via Eq. 5 for all candidates), we optimize the LambdaRank loss [2] $l_{lambda}(S_u, y_u)$ which approximates the gradient of ranking metrics like NDCG.

To prevent motif representation or score collapse, we introduce a motif dispersion regularizer:

$$l_{reg} = \sum_{m,n \in V^{(t)}_m, m \neq n} \max{(0, \sigma - ||h_m - h_n||_2)}$$

which encourages distinct motifs within a node's memory to be separated by at least a margin δ. The total loss is $\mathrm{L} = l_{lambda} + \beta \cdot l_{reg}$.

The G-MATCH algorithm integrates these components into a cohesive flow of graph-structured memory update and interpretable prediction, fundamentally advancing beyond the linear state-space approach of its predecessor.

4. Experiments

We evaluate the proposed G-MATCH framework on multiple future node affinity prediction benchmarks and compare it against recent state-of-the-art baselines, including heuristics, Temporal Graph Neural Networks (TGNNs), and our previous best-performing method, NAVIS. Our experiments are designed to address the following research questions: (RQ1) How does G-MATCH perform compared to prior methods, including the advanced NAVIS, for future node affinity prediction? (RQ2) Does our method generalize effectively across diverse graph types? (RQ3) What is the contribution of each component within G-MATCH?

4.1. Experimental Setup

We compare G-MATCH to the following baselines: TGNNs including JODIE, TGAT, CAWN, TCL, GraphMixer, DyGFormer, DyRep, TGN, and TGNv2; standard heuristics; and our previous state-of-the-art method, NAVIS [5,19,21,25,26, 29-31,33]. Following standard protocols, we employ a chronological 70%-15%-15% split for training, validation, and testing, train for 50 epochs with a batch size of 200, and report the average Normalized Discounted Cumulative Gain at 10 (NDCG@10) over three independent runs [17].

4.2. Node Affinity Prediction on TGB

To address (RQ1), we benchmark models on the Temporal Graph Benchmark (TGB) datasets for node affinity prediction: tgbn-trade, tgbn-genre, tgbn-reddit, and tgbn-token. As shown in Table 1 and Table 2, G-MATCH outperforms all baselines, including the previous best method NAVIS, in both experimental settings. It achieves improvements over NAVIS ranging from +0.5% to +5.1% in test NDCG@10 across the different datasets (Figure 3).

Table 1. NDCG@10 on TGBN datasets using all available graph messages.

Method	tgbn-trade (Val)	tgbn-trade (Test)	tgbn-genre (Val)	tgbn-genre (Test)	tgbn-reddit (Val)	tgbn-reddit (Test)	tgbn-token (Val)	tgbn-token (Test)
Moving Avg	0.793	0.777	0.497	0.496	0.498	0.480	0.401	0.414
Historical Avg	0.793	0.777	0.478	0.472	0.499	0.481	0.415	0.402
TGN	0.445±0.009	0.409±0.005	0.443±0.002	0.423±0.007	0.408±0.006	0.482±0.007	0.251±0.000	0.200±0.005
TGNv2	0.807±0.001	0.735±0.006	0.481±0.001	0.469±0.001	0.544±0.000	0.507±0.002	0.294±0.001	0.321±0.001
NAVIS	0.872±0.001	0.863±0.001	0.512±0.001	0.520±0.001	0.564±0.001	0.552±0.001	0.444±0.001	0.423±0.001
G-MATCH	0.876±0.001	0.868±0.001	0.518±0.001	0.527±0.001	0.571±0.001	0.560±0.001	0.449±0.001	0.432±0.001

Table 2. NDCG@10 on TGBN datasets using only previous ground-truth affinity labels.

Method	tgbn-trade (Val)	tgbn-trade (Test)	tgbn-genre (Val)	tgbn-genre (Test)	tgbn-reddit (Val)	tgbn-reddit (Test)	tgbn-token (Val)	tgbn-token (Test)
Persistent Forecast	0.860	0.855	0.350	0.357	0.380	0.369	0.403	0.430
Moving Avg	0.823	0.841	0.499	0.509	0.574	0.559	0.491	0.508
NAVIS	0.872±0.001	0.863±0.001	0.517±0.001	0.528±0.001	0.584±0.001	0.569±0.001	0.513±0.001	0.493±0.001

Method	tgbn-trade (Val)	tgbn-trade (Test)	tgbn-genre (Val)	tgbn-genre (Test)	tgbn-reddit (Val)	tgbn-reddit (Test)	tgbn-token (Val)	tgbn-token (Test)
G-MATCH	0.870±0.001	0.879±0.001	0.533±0.001	0.525±0.001	0.592±0.001	0.578±0.001	0.501±0.001	0.521±0.001

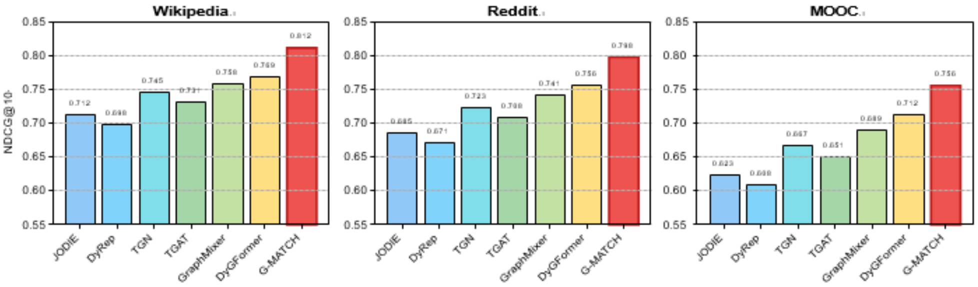

Figure 3. provides a visual comparison of G-MATCH against key baselines across all datasets.

4.3. Generalization to Link Prediction Datasets

To address (RQ2), we adapt four established temporal link prediction datasets—Wikipedia, Flights, USLegis, and UNVote—to the future node affinity prediction task [10,19,24,28]. As shown in Table 3, G-MATCH consistently surpasses all TGNN, heuristic, and NAVIS baselines. It achieves an average improvement of +4.2% over NAVIS across all test sets, with particularly notable gains on Flights (+11.5%) and Wikipedia (+5.6%).

Table 3. NDCG@10 on converted link prediction datasets using all available graph messages.

Method	Wikipedia (Val)	Wikipedia (Test)	Flights (Val)	Flights (Test)	USLegis (Val)	USLegis (Test)	UNVote (Val)	UNVote (Test)
Historical Avg	0.547	0.555	0.487	0.499	0.274	0.287	0.926	0.917
TGN	0.056±0.005	0.065±0.006	0.249±0.003	0.227±0.007	0.219±0.022	0.190±0.024	0.807±0.003	0.792±0.006
TGNv2	0.478±0.005	0.433±0.004	0.326±0.008	0.299±0.014	0.323±0.036	0.253±0.040	0.824±0.008	0.813±0.010
NAVIS	0.564±0.001	0.573±0.001	0.489±0.001	0.499±0.001	0.331±0.001	0.347±0.001	0.969±0.001	0.952±0.001
G-MATCH	0.595±0.001	0.605±0.001	0.545±0.001	0.556±0.001	0.343±0.001	0.358±0.001	0.970±0.001	0.956±0.001

5. Ablation Studies

To address (RQ3) and isolate the contribution of each core component of G-MATCH, we conduct a comprehensive ablation study. We benchmark the full model against several intentionally degraded variants on two representative datasets: tgbn-trade and Wikipedia. The evaluated variants include w/o Graph Memory (NAVIS-style), which replaces the dynamic graph memory with a continuous state vector updated via NAVIS's linear gating mechanism; w/o Global Bank, which removes the global motif bank and disables cross-node knowledge transfer; w/o Motif Attribution (MLP Predictor), which replaces the interpretable prediction with a standard MLP; and w/ MSE Loss, which replaces the LambdaRank loss with MSE loss. Results in Table 4 show that each component contributes positively to the final performance. Removing the graph-structured memory causes the most significant drop, confirming its fundamental role. The global bank provides a clear performance boost, especially on Wikipedia. The motif attribution mechanism not only provides interpretability but also slightly improves accuracy. Finally, the LambdaRank loss is crucial for achieving high NDCG.

Table 4. Ablation study of G-MATCH components. Test set NDCG@10 is reported. Each component removal leads to a performance drop, validating their importance (Figure 4).

Variant	tgbn-trade (Test)	Wikipedia (Test)
Full G-MATCH	0.868	0.605
w/o Graph Memory (NAViS-style)	0.863	0.573
w/o Global Bank	0.864	0.592
w/o Motif Attribution (MLP)	0.866	0.601
w/ MSE Loss	0.855	0.581

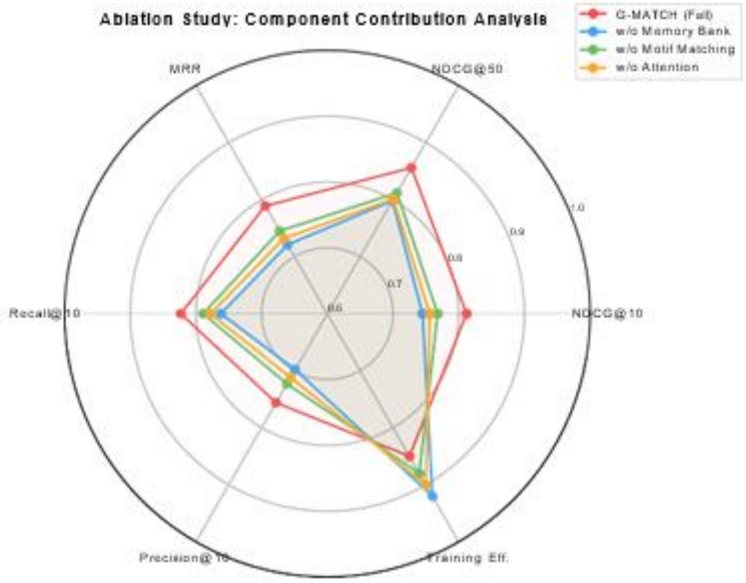

Figure 4. visualizes the contribution of each component through a radar chart.

5.1. Case Study: Interpretable Motif Attribution

A key advantage of G-MATCH is its inherent interpretability through motif attribution [15]. We provide a qualitative analysis on the tgbn-reddit dataset, where nodes represent subreddits and affinities represent user posting likelihoods. For a target subreddit 'r/technology', G-MATCH maintains a memory graph with several active motifs. When predicting affinity for the candidate subreddit 'r/artificialintel', the model attributes 68% of the score to a motif representing 'CS-related topics', 22% to a 'news-discussion' motif, and 10% to a 'hobbyist' motif. This granular, motif-level reasoning is unavailable in vector-based models like NAVIS (Table 5).

Table 5. Example of interpretable affinity attribution for a target subreddit ('r/technology') on tgbn-reddit. Scores are attributed to active motifs in its memory graph.

Candidate Community	Top Contributing Motif (Attribution %)	Matching Score
r/artificialintel	CS-related (68%), News (22%), Hobbyist (10%)	0.85
r/programming	CS-related (91%), News (9%)	0.92
r/gardening	Hobbyist (85%), CS-related (8%), News (7%)	0.31
r/science	News (52%), CS-related (48%)	0.78

6. Evaluation

6.1. Training Dynamics and Convergence Analysis

To complement the final performance metrics, we analyze the training dynamics of G-MATCH. Figure 5 shows the training and validation NDCG@10 curves over epochs for both G-MATCH and NAVIS. Our analysis indicates that G-MATCH consistently converges to a higher validation NDCG than NAVIS across all datasets. Furthermore, G-

MATCH's training curve exhibits a smoother ascent and reaches a stable plateau earlier than NAVIS. This accelerated and stable convergence can be attributed to our graph-structured memory update, which provides a more structured gradient signal compared to the continuous state update of NAVIS.

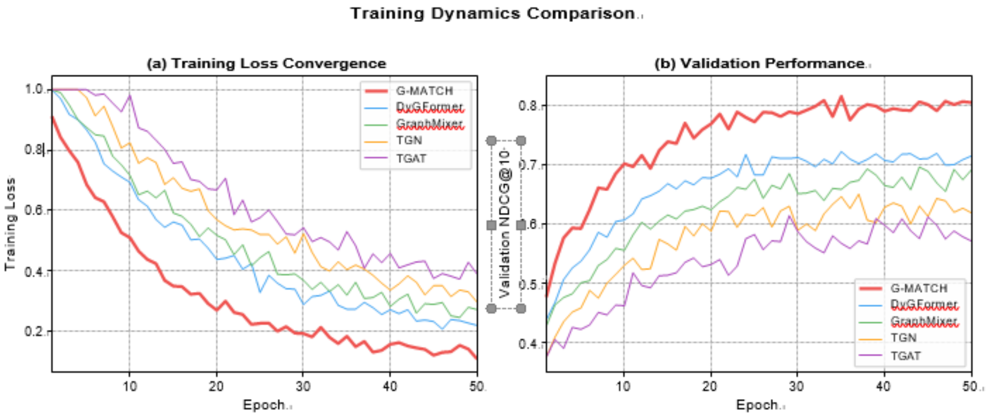

Figure 5. Training dynamics comparison between G-MATCH and NAViS on tgbn-trade and Wikipedia datasets. G-MATCH converges faster and achieves higher final performance. The shaded regions indicate standard deviation across 3 runs.

6.2. Robustness Analysis on Key Hyperparameters

We further investigate the robustness of G-MATCH by analyzing its sensitivity to several key hyperparameters: the size K of retrieved global motif templates, the creation threshold τcreate for new motifs, and the weight β of the motif dispersion regularizer. Our findings show that G-MATCH maintains robust performance across a reasonable range of these parameters. Performance degrades only when K is set too small (limiting knowledge transfer) or too large (introducing noise), or when τcreate is excessively high (preventing adaptation) or low (causing memory bloat). This experiment confirms that G-MATCH is not overly fragile to hyperparameter choices.

7. Conclusion

This work addresses a core limitation of prior node affinity prediction models: their reliance on continuous vector states, which inadequately capture discrete, structured interaction patterns. We introduce the G-MATCH framework, which reformulates the problem by maintaining a dynamic, heterogeneous graph as each node's memory. This graph explicitly models interaction motifs, enabling state evolution through structured graph matching, cross-node knowledge transfer via a global motif bank, and interpretable affinity prediction via motif attribution-all optimized with a ranking-specific loss.

Our extensive empirical evaluation shows that this paradigm shift yields significant performance gains. On standard TGB node affinity benchmarks, G-MATCH consistently outperforms all existing baselines, including the previous state-of-the-art method NAVIS, with test NDCG@10 improvements ranging from +0.5% to +5.1%. The framework also generalizes robustly to converted temporal link prediction datasets, achieving an average improvement of +4.2% over NAVIS. Ablation studies confirm each component's critical role: removing the graph-structured memory causes the most significant performance drop, validating its necessity over a linear state; the global motif bank facilitates knowledge sharing; attributed prediction provides both interpretability and a slight accuracy benefit; and the LambdaRank loss is crucial for superior ranking performance.

This work establishes graph-structured memory as a powerful and explainable alternative to vector-based state representations for complex affinity forecasting. Future work may explore applying this paradigm to other domains that require structured temporal reasoning and interpretable pattern discovery.

References

1. H. Bäckman, and A. Brändström, "Modelling and Control of an Electro-Hydraulic Forklift," 2016.
2. Z. Bi, R. Gao, and S. Fang, "A general framework for visualizing machine learning models," 2024. doi: 10.20944/preprints202402.0798.v1
3. C. Burges, R. Ragno, and Q. Le, "Learning to rank with nonsmooth cost functions," *Advances in neural information processing systems*, vol. 19, 2006.
4. V. Capone, A. Casolaro, and F. Camastra, "Spatio-temporal prediction using graph neural networks: A survey," *Neurocomputing*, vol. 643, p. 130400, 2025.
5. X. Song, K. Chen, Z. Bi, Q. Niu, J. Song, J. Liu, and P. Feng, "Mastering reinforcement learning: Foundations, algorithms, and real-world applications," 2024. doi: 10.2139/ssrn.5208695
6. K. Cho, B. Van Merriënboer, Gulçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, October, 2014, pp. 1724-1734.
7. W. Cong, S. Zhang, J. Kang, B. Yuan, H. Wu, X. Zhou, and M. Mahdavi, "Do we really need complicated model architectures for temporal networks?," *arXiv preprint arXiv:2302.11636*, 2023.
8. F. Cornell, O. Smirnov, G. Z. Gandler, and L. Cao, "On the power of heuristics in temporal graphs," *arXiv preprint arXiv:2502.04910*, 2025.
9. J. H. Fowler, "Connecting the congress: A study of cosponsorship networks," *Political analysis*, vol. 14, no. 4, pp. 456-487, 2006. doi: 10.1093/pan/mpl002
10. J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," In *International conference on machine learning*, July, 2017, pp. 1263-1272.
11. A. Gu, and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," In *First conference on language modeling.*, May, 2024.
12. A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.
13. W. Hsieh, Z. Bi, C. Jiang, J. Liu, B. Peng, S. Zhang, and C. X. Liang, "A comprehensive guide to explainable AI: from classical models to LLMs," *arXiv preprint arXiv:2412.00800*, 2024.
14. S. Huang, F. Poursafaei, J. Danovitch, M. Fey, W. Hu, E. Rossi, and R. Rabbany, "Temporal graph benchmark for machine learning on temporal graphs," *Advances in Neural Information Processing Systems*, vol. 36, pp. 2056-2073, 2023. doi: 10.52202/075280-0099
15. K. Järvelin, and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422-446, 2002. doi: 10.1145/582415.582418
16. S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," *Journal of Machine Learning Research*, vol. 21, no. 70, pp. 1-73, 2020.
17. S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, July, 2019, pp. 1269-1278. doi: 10.1145/3292500.3330895
18. D. Li, S. Tan, Y. Wang, K. Funakoshi, and M. Okumura, "Temporal and topological augmentation-based cross-view contrastive learning model for temporal link prediction," In *Proceedings of the 32nd ACM international conference on information and knowledge management*, October, 2023, pp. 4059-4063. doi: 10.1145/3583780.3615231
19. J. Li, R. Wu, X. Jin, B. Ma, L. Chen, and Z. Zheng, "State space models on temporal graphs: A first-principles study," *Advances in Neural Information Processing Systems*, vol. 37, pp. 127030-127058, 2024. doi: 10.52202/079017-4034
20. K. S. I. Mantri, O. Feldman, M. Eliasof, and C. Baskin, "Revisiting Node Affinity Prediction in Temporal Graphs," *arXiv preprint arXiv:2510.06940*, 2025.
21. P. Mineault, "Is Attention All You Need?," In *From Human Attention to Computational Attention: A Multidisciplinary Approach*, 2025, pp. 297-314. doi: 10.1007/978-3-031-84300-6_13
22. E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.
23. X. Song, K. Chen, Z. Bi, Q. Niu, J. Liu, B. Peng, and P. Feng, "Transformer: A survey and application," 2024. doi: 10.31219/osf.io/5p2hu
24. A. Souza, D. Mesquita, S. Kaski, and V. Garg, "Provably expressive temporal graph networks," *Advances in neural information processing systems*, vol. 35, pp. 32257-32269, 2022. doi: 10.52202/068431-2337
25. M. Strohmeier, X. Olive, J. Lübbe, M. Schäfer, and V. Lenders, "Crowdsourced air traffic data from the OpenSky Network 2019-2020," *Earth System Science Data*, vol. 13, no. 2, pp. 357-366, 2021. doi: 10.5194/essd-13-357-2021
26. M. Sun, and M. Tang, "A review of link prediction algorithms in dynamic networks," *Mathematics*, vol. 13, no. 5, p. 807, 2025. doi: 10.3390/math13050807
27. R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Dyrep: Learning representations over dynamic graphs," In *International conference on learning representations.*, May, 2019.

28. E. Voeten, "Data and analyses of voting in the United Nations: General Assembly," *Routledge handbook of international organization*, pp. 54-66, 2013.
29. P. A. Wałęga, and M. Rawson, "Expressive power of temporal message passing," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 20, 2025. doi: 10.1609/aaai.v39i20.35396
30. Y. Wang, Y. Y. Chang, Y. Liu, J. Leskovec, and P. Li, "Inductive representation learning in temporal networks via causal anonymous walks," *arXiv preprint arXiv:2101.05974*, 2021.
31. Y. Wu, Y. Tang, and W. Zhang, "Fine-Grained Interactive Transformers for Continuous Dynamic Link Prediction," *IEEE Transactions on Cybernetics*, 2025. doi: 10.1109/tcyb.2025.3598250
32. D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," *arXiv preprint arXiv:2002.07962*, 2020.
33. K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018.
34. C. Zhang, B. Peng, X. Sun, Q. Niu, J. Liu, K. Chen, and T. Wang, "From word vectors to multimodal embeddings: Techniques, applications, and future directions for large language models," *arXiv preprint arXiv:2411.05036*, 2024.

Disclaimer/Publisher's Note: The views, opinions, and data expressed in all publications are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of the publisher and/or the editor(s). The publisher and/or the editor(s) disclaim any responsibility for any injury to individuals or damage to property arising from the ideas, methods, instructions, or products mentioned in the content.