RULECAST: EVIDENCE-CONSTRAINED PARTNER-ROLE FORECASTING FOR TEMPORAL LINK PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing temporal link predictors aggregate source history and score candidate compatibility, which overlooks the structural regularity of future partner roles. We propose a fundamentally different factorization: future links arise from evidence-constrained partner-role forecasting. First, the source's historical interactions predict the expected behavioral role of its next partner. Second, candidate nodes are grounded by candidate behavior and query-specific historical evidence. Our RULECAST model implements this paradigm with time-varying soft roles and a protocol encoder that forecasts role distributions. It employs specialized experts to model role compatibility, history matching, reciprocity, closure, and periodicity, alongside a candidate-aware routing mechanism that activates only valid evidence-supported experts. A lightweight identity residual preserves transductive node characteristics. Evaluated on ten temporal graph benchmarks, RULECAST consistently improves MRR and improves AP and ROC-AUC on most datasets. Ablations verify that role forecasting and evidence grounding are mutually essential, validating our new paradigm for temporal link prediction.

1 INTRODUCTION

Temporal interaction networks record who interacts with whom and when in communication, online platforms, recommendation systems, and transportation. A central prediction problem is to anticipate the next destination of an interaction as a source's local context evolves. For a source and a candidate set at a query time, temporal link prediction asks which candidate will be the next interaction partner after observing the preceding events. Existing models encode history with temporal attention (Xu et al., 2020), node memories (Rossi et al., 2020), anonymous walks (Wang et al., 2021), lightweight mixers (Cong et al., 2023), or patched sequences (Yu et al., 2023). Most then pass the resulting representations to a shared pair decoder, leaving the source's next-partner demand implicit. CRAFT (Yi et al., 2025a) is explicitly target-aware: it matches each candidate directly against the source history. Its final score still combines heterogeneous reasons for interaction within one compatibility function. It also does not explicitly forecast what kind of partner the source is likely to seek next.

Candidates in the same query can be grounded by different historical evidence. A recent contact, a reverse-direction contact, a local structural connection, or behavior similar to the source's earlier partners supplies a distinct route for grounding a common forecast of the next partner's behavioral role. A single latent compatibility score leaves three decisions implicit: which behavioral role the source expects its next partner to occupy, which candidates are supported by the available evidence, and how that evidence should shape the final score.

This separation makes expert availability part of the model: repeat evidence requires a prior pair event, and periodic evidence requires at least two prior pair events. RULECAST encodes these query-specific support conditions explicitly.

RULECAST realizes this two-stage factorization through three coupled levels. The first two form the forecasting stage: strict-history behavior determines each node's time-varying behavioral role, and the behavioral roles of the source's past partners, together with their temporal context, define an interaction protocol that predicts the behavioral role of the next partner. The third level grounds this forecast for each candidate using candidate behavior and query-time pair history. Role compatibility

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.

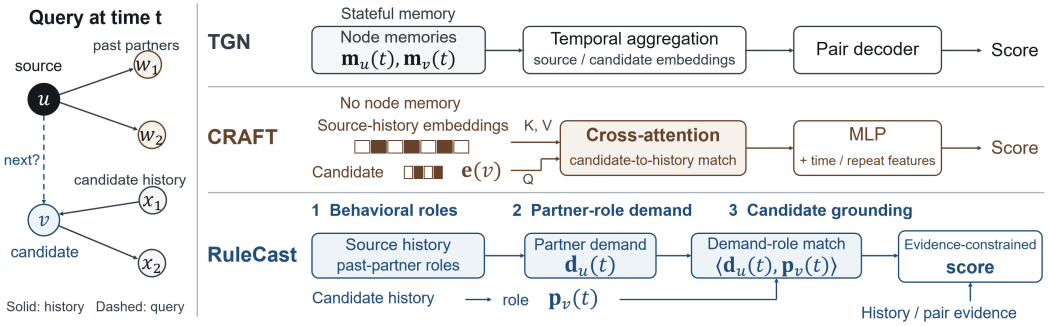

Figure 1: Prediction views for a source–candidate query. Solid edges denote past interactions; the dashed edge denotes the query. TGN builds temporal embeddings from node memories. CRAFT directly matches candidate embeddings to source history without node memory. RULECAST forecasts partner-role demand and scores its compatibility with the candidate's behavioral role, alongside supported historical and pair evidence.

evaluates the forecast against the candidate's behavioral role, while history matching, repetition, reciprocity, closure, and periodicity provide supported contextual evidence. A candidate-conditioned router mixes only the supported experts. The name RULECAST reflects this procedure: it casts a forecast over future partner roles and applies query-specific evidence rules to ground each candidate. Figure 2 shows the complete scoring path. We use *predictive signature* for an observable, score-relevant pattern in the event history that contributes predictive evidence to scoring.

Figure 1 illustrates this factorization relative to memory-based and history-matching prediction.

Our contributions are:

- We introduce a behavior-first factorization of temporal link prediction into evidence-constrained partner-role forecasting and candidate grounding. A source's history forecasts the expected behavioral role of its next partner, and each candidate is then grounded by its behavior and query-specific evidence. This moves the reusable predictive object from a source representation used for pairwise matching to an interaction protocol over future partner roles.
- We introduce evidence-constrained routing. Expert support is defined for each source–candidate query, unavailable experts receive zero routing probability, and the router combines only scores backed by the observed history.
- We represent historical partners by their behavior at the interaction time and use candidate-aware behavioral history matching as one grounding route. This separates the central partner-role forecast from CRAFT-style candidate-to-history matching.
- We evaluate RULECAST on ten temporal-graph benchmarks against reproduced CRAFT and established baselines. Component controls and evidence-conditioned analyses examine identity reliance, source demand, candidate grounding, and routing behavior.

2 RELATED WORK

Temporal node and sequence encoders. TGAT learns functional time encodings for temporal attention (Xu et al., 2020). TGN combines node memories with graph operators (Rossi et al., 2020), and CAWN retrieves anonymous temporal motifs (Wang et al., 2021). GraphMixer replaces recurrent and attention modules with MLP components (Cong et al., 2023). DyGFormer models long first-hop histories through co-occurrence encoding and patching (Yu et al., 2023). These methods provide strong history encoders. RULECAST changes the final decision: the source protocol predicts a partner role, and separate experts score distinct pair signatures.

Pair-aware and target-aware prediction. TNCN maintains temporal common-neighbor information efficiently (Zhang et al., 2024). TPNet unifies relative encodings through temporal walk

matrices (Lu et al., 2024). CRAFT directly queries a source's recent interactions with the destination representation (Yi et al., 2025a). RULECAST is also pair-conditioned, but separates source demand from candidate grounding. Different candidates can therefore rely on different supported predictive signatures.

Patterns, motifs, and experts. TGACN recognizes link-evolution patterns by appending a query link to historical links (Liu & Huang, 2024). NeurTWs learns motif-aware temporal representations (Jin et al., 2022), and TempME discovers temporal motifs that explain a trained predictor (Chen & Ying, 2023). Mixture models have also appeared in link prediction. Link-MoE routes static node pairs to link predictors (Ma et al., 2024), and ESSEN uses thermodynamic experts for temporal-network decoding (Huang et al., 2023a). RULECAST combines three elements that these methods treat separately: dynamic partner-role demand, predictive signatures with query-specific support, and evidence-route intervention tests.

Evaluation. EdgeBank and harder negative sampling show that repeated edges and easy negatives can inflate dynamic link prediction results (Poursafaei et al., 2022). TGB standardizes large-scale evaluation (Huang et al., 2023b), and TGB-Seq emphasizes sequential dynamics and unseen edges (Yi et al., 2025b). We follow the benchmark-specific evaluators and negative-sampling protocols.

3 PROBLEM FORMULATION

Let $\mathcal{E} = \{e_k = (u_k, v_k, t_k, \mathbf{x}_k)\}_{k=1}^{N}$ be events sorted by non-decreasing time. The strict history at query time t is

$$\mathcal{H}_t = \{e_k \in \mathcal{E} : t_k < t\}. \tag{1}$$

Events sharing timestamp t are excluded together. Given source u and candidate list $\mathcal{C}(u,t) = (v_0 = v^+, v_1^-, \ldots, v_K^-)$, a model scores each entry with $s(u,v,t)$ to rank the true destination v^+ against sampled negatives. Candidate sums count list entries with their sampling multiplicity.

Each predictive-signature expert $m \in \mathcal{M}$ contains three components: a historical evidence view, an availability predicate, and a scorer.

4 RULECAST

4.1 BEHAVIORAL STATE AND TIME-VARYING BEHAVIORAL ROLES

The behavioral state is a compact, query-relative summary of four dynamic regularities. **Activity scale and relationship reach** capture how much a node has interacted and how broadly it has connected. **Temporal cadence** captures whether it is recently active and whether its interactions follow a concentrated or variable rhythm. **Relational organization** captures persistence with prior partners and directional balance. **Short-term momentum** captures whether current activity is increasing or fading relative to the preceding window. Together, these four axes provide a dynamic, label-agnostic description of node behavior.

Let $\mathcal{H}_i(t) = \{(n_{ij}, \tau_{ij}, d_{ij})\}_{j=1}^{N_i}$ denote the events incident to i in $\mathcal{H}_t$, ordered by time. Here n_{ij} is the other endpoint and $d_{ij} = 1$ when i is the source of the event (and 0 otherwise). We compute the corresponding behavior vector $\mathbf{b}_i(t) \in \mathbb{R}^9$ from this strict history:

$$\mathbf{b}_i(t) = \big[\log(1+N_i), \log(1+U_i), \log(1+\delta_i/\sigma), \mu_i^{\log\Delta}, s_i^{\log\Delta}, \rho_i^{\mathrm{rep}}, \rho_i^{\mathrm{out}}, \log(1+A_i), g_i\big]. \tag{2}$$

The first two entries are the event count N_i and the number U_i of distinct historical partners. The temporal-cadence entries use recency $\delta_i = t - \tau_{iN_i}$ and the mean and standard deviation of normalized log gaps $\log(1 + (\tau_{ij} - \tau_{i,j-1})/\sigma)$. Relational organization is measured by the repeated-partner ratio $\rho_i^{\mathrm{rep}} = 1 - U_i/N_i$ and the outgoing-event ratio $\rho_i^{\mathrm{out}} = N_i^{-1}\sum_j d_{ij}$. Finally, A_i counts events in a recent window and g_i compares that count with the preceding window. We set the state to zero for an empty history and set the two gap moments to zero when fewer than two events are present. Appendix A gives the complete window and normalization definitions. The global time scale σ and feature normalization statistics are estimated from training events, while each state itself uses events strictly before t. These nine coordinates are used together: they form one normalized behavioral

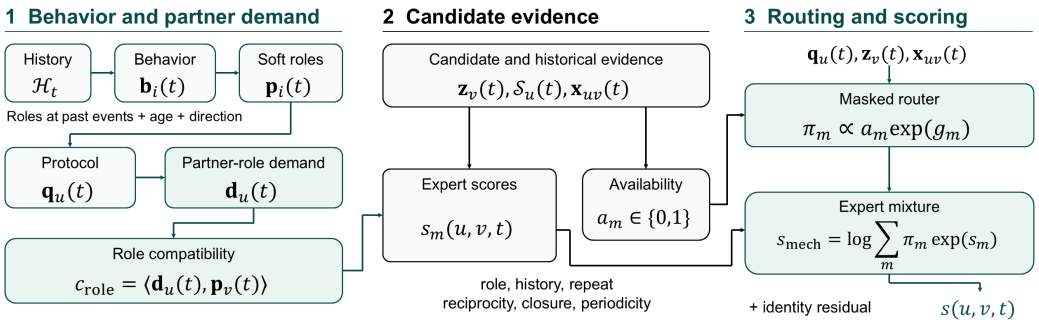

Figure 2: RULECAST workflow. The nine strict-history statistics are jointly assembled into $\mathbf{b}_i(t)$ and mapped to a time-varying soft role distribution $\mathbf{p}_i(t)$; the source protocol then predicts partner-role demand in the same role space. Behavioral and pair evidence provide expert scores and availability masks. A candidate-conditioned router combines available experts, and an identity residual completes the score. Query arguments are omitted in the compact expressions.

state rather than nine independent predictors, so activity, reach, cadence, relational organization, and short-term momentum jointly define the role input.

An MLP maps the full normalized state $\widetilde{\mathbf{b}}_i(t)$ to a time-varying behavioral role:

$$\mathbf{z}_i(t) = f_b(\widetilde{\mathbf{b}}_i(t)), \qquad \mathbf{p}_i(t) = \mathrm{softmax}((W_r \mathbf{z}_i(t) + \mathbf{c}_r)/\tau_r)\,, \tag{3}$$

where $\mathbf{z}_i(t) \in \mathbb{R}^d$, $W_r \in \mathbb{R}^{R \times d}$, $\mathbf{c}_r \in \mathbb{R}^R$, and $\mathbf{p}_i(t) \in \Delta^{R-1}$. Behavioral roles are time-varying distributions, not fixed node types. We use *partner role* to mean the behavioral role occupied by an interaction partner; partner-role demand is therefore a forecast over this same role space. Role labels are assigned only after inspecting their behavior profiles. All reported experiments use the fixed role temperature $\tau_r = 1.0$; it is not tuned per dataset.

4.2 INTERACTION PROTOCOL AS PREDICTIVE PARTNER-ROLE DEMAND

Let $\mathcal{S}_u(t) = \{(w_\ell, t_\ell, d_\ell)\}_{\ell=1}^{L}$ be u's most recent interactions before t, ordered from old to new. Importantly, the ℓ-th partner is represented by its behavioral role at the instant before that interaction, $\mathbf{p}_{w_\ell}(t_\ell^-)$, rather than at query time. Its protocol token is

$$\mathbf{h}_\ell^{\text{in}} = W_{\text{in}}\big[W_p \mathbf{p}_{w_\ell}(t_\ell^-) \,\|\, f_t(\log(1 + (t - t_\ell)/\sigma)) \,\|\, \mathbf{e}_{d_\ell}\big] + \mathbf{c}_{\text{in}}. \tag{4}$$

The three concatenated components have width d_p; $W_{\text{in}} \in \mathbb{R}^{d \times 3d_p}$ and $\mathbf{c}_{\text{in}} \in \mathbb{R}^d$. A GRU returns its last valid hidden state $\mathbf{q}_u(t)$; an explicit learned vector represents empty history. The resulting *predictive demand* summarizes which behavioral role the next partner is likely to occupy:

$$\mathbf{d}_u(t) = \mathrm{softmax}(f_d([\mathbf{q}_u(t) \,\|\, \mathbf{z}_u(t)]))\,. \tag{5}$$

The basic role compatibility with candidate v is

$$c_{\text{role}}(u, v, t) = \langle \mathbf{d}_u(t), \mathbf{p}_v(t) \rangle, \qquad s_{\text{role}}(u, v, t) = \beta \log \max\{c_{\text{role}}(u, v, t), \epsilon\} + b. \tag{6}$$

Here $\epsilon = 10^{-8}$. The scalar $\beta \in \mathbb{R}$ is initialized to 1 and learned jointly with the bias b.

4.3 CANDIDATE EVIDENCE EXPERTS AND AVAILABILITY

RULECAST uses the predictive-signature set

$$\mathcal{M} = \{\text{role}, \text{hist}, \text{rep}, \text{rec}, \text{clo}, \text{per}\},$$

where closure is optional and is disabled in the large-scale results in Section 5. The role expert evaluates demand–candidate role compatibility. Behavioral roles summarize node-level dynamics independently of the candidate. The history and pair experts then provide candidate-conditioned

evidence for grounding a particular candidate under the source's predicted demand. The history expert provides candidate-aware grounding without reverting to raw identity matching. For candidate v, it attends to the latent behavior states of u's historical partners:

$$\eta_{\ell v}(t) = \mathrm{softmax}_{\ell: a_\ell = 1} \frac{\langle W_Q \mathbf{z}_v(t), W_K \mathbf{z}_{w_\ell}(t_\ell^-) \rangle}{\sqrt{d}}, \tag{7}$$

$$\bar{\mathbf{z}}_{u \to v}(t) = \sum_{\ell=1}^{L} \eta_{\ell v}(t)\, W_V \mathbf{z}_{w_\ell}(t_\ell^-), \tag{8}$$

where a_ℓ is the history-token mask. For empty history, set all $\eta_{\ell v} = 0$ and $\bar{\mathbf{z}}_{u \to v} = \mathbf{0}$. The score is

$$s_{\mathrm{hist}}(u, v, t) = f_{\mathrm{hist}}\big([\mathbf{z}_v(t) \,\|\, \bar{\mathbf{z}}_{u \to v}(t) \,\|\, \mathbf{z}_v(t) \odot \bar{\mathbf{z}}_{u \to v}(t) \,\|\, \gamma_{uv}^{\mathrm{hist}}(t)]\big), \tag{9}$$

with $\gamma_{uv}^{\mathrm{hist}}(t) = \langle W_Q \mathbf{z}_v(t), \bar{\mathbf{z}}_{u \to v}(t) \rangle / \sqrt{d}$. The matched objects are behavior states at historical event times. This expert provides candidate-conditioned behavioral support alongside the role score.

The remaining experts provide repeat, reciprocal, closure, and periodic evidence for candidate grounding. For a directed pair (u, v), let $n_{u \to v}(t)$ be its strict historical count and $\Delta_{u \to v}(t)$ the gap since its last event. Repeat and reciprocity use

$$\mathbf{x}_{\mathrm{rep}} = [\log(1 + n_{u \to v}), \exp(-\Delta_{u \to v}/(10\sigma))], \tag{10}$$

$$\mathbf{x}_{\mathrm{rec}} = [\log(1 + n_{v \to u}), \exp(-\Delta_{v \to u}/(10\sigma))]. \tag{11}$$

For homogeneous graphs, $c_{uv}(t)$ counts recent shared neighbors. For bipartite source–item graphs, endpoints have disjoint neighbor types, so direct common neighbors are invalid. We instead count non-degenerate three-hop paths $u \to i \leftarrow u' \to v$ with $u' \neq u$ and $i \neq v$. Closure uses the log path count and a freshness statistic based on the oldest edge of each path (Appendix A).

If $n_{u \to v} \geq 2$, define the median prior pair gap $\widehat{P}_{uv}$ and

$$p_{uv}^{\mathrm{per}}(t) = \exp\Big(-\Big|\log(1 + \Delta_{u \to v}/\sigma) - \log(1 + \widehat{P}_{uv}/\sigma)\Big|\Big). \tag{12}$$

Small MLPs map each signature's evidence to $s_m(u, v, t)$. Their support is explicit:

$$\begin{aligned} a_{\mathrm{role}} = 1, \quad a_{\mathrm{hist}} = \mathbb{I}[|\mathcal{S}_u(t)| > 0], \quad a_{\mathrm{rep}} = \mathbb{I}[n_{u \to v} > 0], \quad a_{\mathrm{rec}} = \mathbb{I}[n_{v \to u} > 0], \\ a_{\mathrm{clo}} = \mathbb{I}[c_{uv} > 0], \quad a_{\mathrm{per}} = \mathbb{I}[n_{u \to v} \geq 2]. \end{aligned} \tag{13}$$

The role expert is always available. Every other expert receives routing mass only when its required history exists.

The non-role terms in this decomposition are evidence signatures derived from strict temporal history and query-time pair records, rather than additional role labels. Repeat, reciprocity, closure, and periodicity describe complementary ways in which a candidate is supported by the observed interaction record; the role pathway remains the source-side forecast of the next partner's behavioral state.

4.4 Evidence-Constrained Candidate Routing

The router is conditioned on both source protocol and candidate evidence:

$$\mathbf{g}_{uv}(t) = f_g([\mathbf{q}_u(t) \,\|\, \mathbf{z}_v(t) \,\|\, \mathbf{x}_{uv}(t)]). \tag{14}$$

Here $\mathbf{x}_{uv}(t) \in \mathbb{R}^7$ collects the pair statistics

$$[\log(1 + n_{u \to v}),\, r_{u \to v},\, \log(1 + n_{v \to u}),\, r_{v \to u},\, \log(1 + c_{uv}),\, r_{uv}^{\mathrm{clo}},\, p_{uv}^{\mathrm{per}}],$$

where n denotes a directed interaction count, r its exponentially decayed recency, c_{uv} the shared-neighbor or bipartite path count, and p_{uv}^{per} the periodicity match. A missing statistic is set to zero. Availability masks router logits and supplies the regularization target. For $m \in \mathcal{M}$, the masked distribution and mixture score are

$$\pi_m(u, v, t) = \frac{a_m(u, v, t) \exp(g_m(u, v, t))}{\sum_{j \in \mathcal{M}} a_j(u, v, t) \exp(g_j(u, v, t))}, \tag{15}$$

$$s_{\mathrm{mech}}(u, v, t) = \log \sum_{m \in \mathcal{M}} \pi_m(u, v, t) \exp(s_m(u, v, t)). \tag{16}$$

Algorithm 1 Evidence-constrained scoring with RULECAST

Input: Minibatch $\mathcal{B} = \{(u_b, v_b^+, t_b, \mathcal{C}_b)\}_{b=1}^{B}$; strict temporal index $\mathcal{I}$; history length L; parameters Θ
Output: Candidate scores $\mathcal{Z}$ and training loss $\mathcal{L}$

Initialize: $\mathcal{Z} \leftarrow \varnothing$, $\mathcal{L}_{\text{rank}} \leftarrow 0$
for all $(u, v^+, t, \mathcal{C}) \in \mathcal{B}$ **do**
  $\mathcal{S}_u(t) \leftarrow \text{RECENTHISTORY}(\mathcal{I}, u, t, L)$ using events with timestamp $< t$
  Compute four-axis normalized behavior states, $\mathbf{z}_u(t)$, historical partner roles $\{\mathbf{p}_{w_\ell}(t_\ell^-)\}_{\ell=1}^{L}$, and masks
  Encode $\mathbf{q}_u(t)$ and partner-role demand $\mathbf{d}_u(t)$ using Eqs. 4–5
  for all $v \in \mathcal{C}$ **do**
    Compute candidate state $\mathbf{z}_v(t)$ and role $\mathbf{p}_v(t)$
    Compute $s_{\text{role}}(u, v, t)$ and $s_{\text{hist}}(u, v, t)$ using Eqs. 6–9
    Query pair evidence $\mathbf{x}_{uv}(t)$ and availability $\{a_m(u, v, t)\}_{m \in \mathcal{M}}$
    Evaluate each supported expert score $s_m(u, v, t)$
    Compute $\pi_m(u, v, t)$ and $s_{\text{mech}}(u, v, t)$ using Eqs. 15–16
    Compute $s(u, v, t)$ using Eq. 17; add (u, v, t, s) to $\mathcal{Z}$
  end for
  Accumulate the K ranking terms from candidate positions 0 and $1{:}K$, scaled by $1/(BK)$
end for
Compute batch role and router terms; set $\mathcal{L}$ using Eq. 19
return $\mathcal{Z}, \mathcal{L}$

Because $a_{\text{role}} = 1$, Eq. 15 is always defined. The log-mixture allows a strong supported expert to dominate without making expert scores directly comparable probabilities.

The mask excludes unavailable expert scores and logits from the mixture. The shared router uses the observed candidate evidence to allocate mass among supported experts; π_m is its conditional mixing coefficient.

An optional transductive identity residual is

$$s(u, v, t) = s_{\text{mech}}(u, v, t) + \alpha \frac{\langle W_q \mathbf{e}_u, W_k \mathbf{e}_v \rangle}{\sqrt{d}}, \qquad \alpha = \text{softplus}(\widetilde{\alpha}), \tag{17}$$

with $\widetilde{\alpha}$ initialized to -4. The no-ID model removes this term entirely.

4.5 OBJECTIVE AND REGULARIZATION

For K negatives per event, the ranking loss is

$$\mathcal{L}_{\text{rank}} = -\frac{1}{BK} \sum_{b=1}^{B} \sum_{k=1}^{K} \log \text{sigmoid}\big(s(u_b, v_b^+, t_b) - s(u_b, v_{bk}^-, t_b)\big) . \tag{18}$$

We collect the auxiliary role terms in $\mathcal{L}_{\text{role}}$: the same ranking loss applied to s_{role}, a batch-level role-usage KL penalty, and a small mean role-entropy penalty. For routing, let $\bar{\pi}_m$ be mean usage and let $q_m \propto \sum_{b,v} a_m(u_b, v, t_b)$ be empirical availability. We minimize $D_{\text{KL}}(\bar{\pi} \| q)$ and add a signed mean router-entropy term. These terms form $\mathcal{L}_{\text{route}}$ and make candidate-level specialization possible when evidence contexts differ. Appendix B gives the full objective and coefficients.

4.6 COMPLEXITY

For supplied candidate lists, let B be the batch size, $C = K + 1$ the candidate count, L the history length, d the hidden width, and $M = |\mathcal{M}|$. With R, $d_p = O(d)$ and fixed-dimensional pair inputs, the neural cost is

$$O\big(B(L + C)d^2 + BCLd + BCMd\big) .$$

The terms cover encoders, GRU, attention projections and scorer, and router hidden layers; candidate–history attention; and pair-expert and router outputs, respectively. The $BCLd$ term matches candidate-to-history attention with the same C, L, d. The identity residual adds $O(|\mathcal{V}|d)$ parameters and $O(BCd^2)$ projection work; without it, parameter count is independent of node count.

The temporal index occupies $O(|\mathcal{E}| + |\mathcal{V}|)$ space. For maximum node-history length Δ, behavior and directed-pair lookups cost $O(B(L + C)\log(1 + \Delta))$ per batch. Periodicity additionally costs

$O(\sum_{b,v} n_{u_b \to v}(t_b))$ to compute medians of pair gaps. These costs precede neural scoring; the batch builder deduplicates candidates within each query. Closure is disabled in the main experiments. With H recent events per node, closure adds $O(BC(H + \log(1 + \Delta)))$ for homogeneous graphs or $O(BCH(H + \log(1 + \Delta)))$ for exact bipartite enumeration.

5 Experimental Protocol

5.1 Research Questions and Data

We evaluate three questions. **RQ1** asks whether source-side role demand and evidence-aware candidate grounding improve ranking. **RQ2** asks whether the improvement comes from the factorization rather than a larger decoder. **RQ3** asks whether routed experts respond to their associated observable signature evidence.

We evaluate Wikipedia, Reddit, UCI, LastFM, Flights, ML-20M, YouTube, tgbl-review, tgbl-flight, and tgbl-coin. The suite covers classic temporal-graph benchmarks, TGB link prediction, and TGB-Seq datasets with strong sequential and unseen-edge dynamics (Huang et al., 2023b; Yi et al., 2025b). We keep every timestamp group in a single split. RuleCast uses the scalable `full_match` configuration in all main results: dynamic roles, source protocol, role/history/repeat/reciprocity/periodic experts, evidence-constrained routing, and a small identity residual. Closure is disabled.

5.2 Baselines and Controls

Table 1 includes 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 SGNN-HN (Yi et al., 2025a). We reproduce CRAFT with the same loader, temporal split, negative-sampling rule, candidate count, and metrics as RuleCast. All ten RuleCast entries in the main tables use the completed three-seed runs; Table 2 reports AP and ROC-AUC for CRAFT and RuleCast.

We use three controls to isolate the components: **Uniform-mechanism** mixes the same signature experts uniformly over available evidence, removing candidate routing; **No-ID / ID-only** separates transferable behavior from transductive identity; and **Direct pair controls** retain pair-history or identity matching without the role–protocol pathway. All learned controls use the same hidden width, timestamp-group split, sampled candidates, seed set, and early-stopping rule. The direct pair controls separate the role forecast from gains due only to a larger pair decoder.

Table 1: Test MRR (%). CRAFT is reproduced under the same protocol as RuleCast. All entries report the mean and standard deviation over three seeds. Review, TGB-F, and Coin denote tgbl-review, tgbl-flight, and tgbl-coin.

Model	Wiki	Reddit	UCI	LastFM	Flights	ML-20M	YouTube	Review	TGB-F	Coin
JODIE	76.48±1.72	77.16±1.27	51.43±1.74	18.49±2.87	20.37±4.18	21.16±0.73	41.67±2.86	41.43±0.15	–	–
DyRep	67.42±4.21	72.33±2.14	14.00±2.71	17.76±6.56	15.62±0.47	19.00±1.69	35.12±4.13	40.06±0.59	55.60±1.40	45.20±4.60
TGAT	72.72±1.50	78.03±0.25	31.50±0.48	24.10±1.26	53.90±1.07	10.47±0.20	43.56±2.53	19.64±0.23	–	60.92±0.57
TGN	82.22±0.39	79.25±0.24	45.29±6.43	25.59±4.43	64.24±2.11	23.99±0.20	55.16±5.89	37.48±0.23	70.50±2.00	58.60±3.70
CAWN	86.07±0.08	87.66±0.04	66.96±0.48	35.10±0.34	80.06±2.10	12.31±0.02	47.55±1.08	19.30±0.10	–	–
GraphMixer	73.57±1.48	70.87±0.38	59.58±1.00	26.76±1.42	42.26±0.49	21.97±0.17	58.87±0.12	36.89±1.50	–	75.57±0.27
DyGFormer	88.69±0.04	88.70±0.05	76.61±0.26	46.50±0.51	84.13±3.46	–	46.08±3.44	22.39±1.52	–	75.17±0.38
SGNN-HN	82.22±0.35	87.42±0.04	57.23±0.39	40.72±0.26	80.80±0.14	33.12±0.01	59.64±0.22	35.62±0.53	89.03±0.03	78.46±0.40
CRAFT	88.21±0.36	89.33±0.05	74.08±1.48	55.12±0.12	83.08±0.13	44.94±0.28	47.55±0.43	32.32±0.35	86.74±0.24	81.18±1.81
RuleCast	**92.19±0.14**	**94.69±0.03**	**86.65±0.10**	**91.39±0.15**	**91.37±0.10**	**54.48±0.21**	**63.10±0.21**	**49.79±0.13**	**90.20±0.04**	**91.46±0.02**

Table 2: Test AP and ROC-AUC (%). Both methods use the same data splits, negative-sampling rule, and evaluation metrics. All entries report the mean and standard deviation over three seeds. Dataset abbreviations follow Table 1.

Model	Metric	Wiki	Reddit	UCI	LastFM	Flights	ML-20M	YouTube	Review	TGB-F	Coin
CRAFT	AP	98.17±0.18	**99.33±0.02**	93.83±0.26	93.96±0.01	98.83±0.01	91.39±0.13	89.17±0.20	81.05±0.09	99.24±0.02	97.72±0.08
	ROC-AUC	97.57±0.24	**99.20±0.02**	91.54±0.12	93.38±0.03	98.76±0.01	92.28±0.13	88.43±0.20	78.54±0.09	99.22±0.01	97.22±0.06
RuleCast	AP	**99.20±0.22**	99.21±0.04	**97.77±0.03**	**98.08±0.07**	**99.01±0.06**	**95.28±0.12**	**94.86±0.21**	**91.98±0.09**	**99.38±0.04**	**98.74±0.11**
	ROC-AUC	**98.97±0.35**	98.96±0.06	**97.00±0.07**	**97.38±0.07**	**98.79±0.08**	**95.90±0.13**	**94.27±0.28**	**91.05±0.19**	**99.44±0.03**	**98.35±0.17**

5.3 Component Ablations

Table 3 reports component ablations on Wikipedia, UCI, LastFM, and Flights under the same split, negatives, optimization schedule, checkpoint rule, and closure-free configuration as the full model. All rows use the same three random seeds as the full model. Direct pair controls remove source-side partner-role forecasting; the remaining controls remove history matching, identity, or candidate-conditioned routing. We read these aggregate controls together with the evidence-stratified results below and the direct controls in Appendix H.

Table 3: Component ablations: test MRR (%). Entries report mean ± standard deviation over three seeds. "No history" removes the behavior-aware history-match expert; "uniform router" averages over available experts; direct pair controls remove the role–protocol pathway.

Variant	Wiki	UCI	LastFM	Flights
RuleCast	**92.19±0.14**	**86.65±0.10**	**91.39±0.15**	**91.37±0.10**
No history	92.01±0.08	86.59±0.13	91.21±0.15	91.16±0.03
No identity	91.97±0.14	86.50±0.07	91.24±0.30	90.87±0.05
Uniform router	92.06±0.00	86.60±0.11	90.96±0.01	91.11±0.12
Role+protocol only	64.41±3.59	60.43±0.19	38.30±0.18	59.56±0.01

5.4 Evidence-Stratified Analysis

We evaluate evidence-stratified behavior by grouping queries according to available evidence: repeated versus new, reverse-history versus none, closure versus none, and periodic versus irregular. The reported analyses compare routing allocations, demand–candidate role overlap, and ranking gains across these groups. Router weights and these comparisons summarize how the fitted predictor uses observed evidence.

Figure 3 reports the resulting allocations on 2,048 held-out Flights queries, with 101 candidates per query. With only source history available, the role and history scores receive mean weights of 0.49 and 0.51. Repeat, reciprocity, and periodic contexts instead distribute substantial mass across the three relational scores (0.27–0.34); the reciprocity score reaches 0.34 in reciprocity contexts. The companion demand plot shows a clear ranking separation: across four demand-confidence quartiles, realized next partners have mean role-overlap scores of 0.084–0.164, compared with 0.002–0.003 for sampled candidates. These descriptive plots summarize the fitted predictor conditional on the observed evidence.

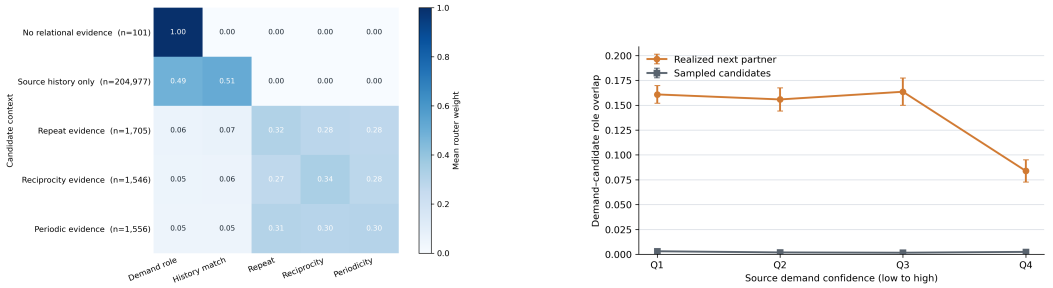

Figure 3: Evidence-conditioned analysis on Flights. **Left:** mean router allocation over held-out candidate contexts. **Right:** role-overlap for realized next partners and sampled candidates across source-demand-confidence quartiles.

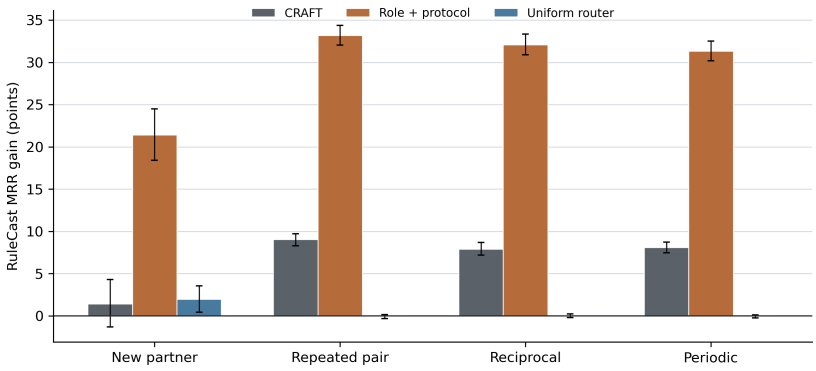

Figure 4: Flights MRR gain by evidence stratum. RuleCast gains $1.4/9.0/7.9/8.1$ points over CRAFT for new/repeated/reciprocal/periodic evidence. Gains are measured against CRAFT, role+protocol-only, and uniform routing on aligned held-out queries; strata may overlap.

5.5 Identity Controls

We compare the full model with no-ID and identity-only controls under the same split, candidates, and evaluator; the complete results are in Appendix H. The no-ID control removes transductive embeddings, while ID-only isolates their contribution without behavioral demand or evidence experts.

5.6 Runtime

Recorded server runs are shorter than CRAFT on the five directly matched datasets; Table 4 reports the end-to-end wall-clock measurements. RuleCast takes 0.16–$0.61\times$ the CRAFT time under the matched protocol, consistent with the bounded-history and cached-state design used for candidate scoring.

Table 4: Recorded end-to-end runtime on the server (minutes). The ratio is RuleCast divided by CRAFT.

Dataset	RuleCast	CRAFT	Ratio
Wikipedia	6.2	14.0	$0.44\times$
Reddit	23.0	86.7	$0.27\times$
UCI	2.6	4.4	$0.61\times$
LastFM	56.5	231.3	$0.24\times$
Flights	185.8	1,189.8	$0.16\times$

6 Discussion

The direct controls isolate the role–protocol factorization from decoder capacity, while the expert set covers complementary predictive signatures. Learned roles are time-varying latent states, and bounded histories with cached states keep scoring efficient. Behavioral axes and time scale are hand-designed, so scores should be read as predictive evidence within the benchmark protocol.

7 Conclusion

RuleCast changes the predictive object in temporal link prediction. Instead of using source history only as context for candidate matching, it forecasts the behavioral role of the next partner and grounds each candidate through supported predictive signatures. Across ten datasets, this factorization improves MRR over CRAFT while retaining the same asymptotic candidate–history attention cost. The results show that partner-role demand provides a useful bridge between source dynamics and candidate-specific evidence.

AI USE STATEMENT

Generative AI tools were used to assist with literature retrieval and discovery and to polish the writing of this manuscript. The authors independently checked the relevant sources, reviewed all AI-assisted text, and take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Sections 3–4 and Algorithm 1 specify the model, objective, temporal split, candidate construction, negative-sampling protocol, metrics, and comparison controls. The appendix gives strict-time query properties, expert specifications, dataset-specific settings, shared hyperparameters, and ablation configurations. Each run records the dataset, split boundaries, sampler, random seed, validation-selected checkpoint, and final metrics; aggregate tables use the same three seeds for all compared learned variants. These details provide the information needed to reproduce the reported experiments.

REFERENCES

Jialin Chen and Rex Ying. TempME: Towards the explainability of temporal graph neural networks via motif discovery. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://openreview.net/forum?id=6OOgw4boZI.

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 *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ayPPc0SyLv1.

Qiyao Huang, Yingyue Zhang, Zhihong Zhang, and Edwin R. Hancock. ESSEN: Improving evolution state estimation for temporal networks using von neumann entropy. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

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

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

Srijan Kumar, Xikun Zhang, and Jure Leskovec. JODIE: Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. doi: 10.1145/3292500.3330895.

Bingqing Liu and Xikun Huang. Link-aware link prediction over temporal graph by pattern recognition. *arXiv preprint arXiv:2402.07199*, 2024. URL https://arxiv.org/abs/2402.07199.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4482.

Li Ma, Haoyu Han, Juanhui Li, Harry Shomer, Hui Liu, Xiaofeng Gao, and Jiliang Tang. Mixture of link predictors on graphs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=X3oeoyJlMw.

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2386.

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

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. DyRep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 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 *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2101.05974.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJeW1yHYwH.

Lu Yi, Runlin Lei, Fengran Mo, Yanping Zheng, Zhewei Wei, and Yuhang Ye. Future link prediction without memory or aggregation. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-0084.

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

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2960.

Xiaohui Zhang, Yanbo Wang, Xiyuan Wang, and Muhan Zhang. Efficient neural common neighbor for temporal graph link prediction. *arXiv preprint arXiv:2406.07926*, 2024. URL https://arxiv.org/abs/2406.07926.

A Implementation Details

Behavior windows and dynamic axes. The nine-dimensional state is organized into activity scale and relationship reach (N_i, U_i), temporal cadence $(\delta_i, \mu_i^{\log\Delta}, s_i^{\log\Delta})$, relational organization $(\rho_i^{\mathrm{rep}}, \rho_i^{\mathrm{out}})$, and short-term momentum (A_i, g_i). It uses event endpoints, directions, and timestamps; no dataset-specific node metadata is required.

For $N_i > 0$, let $U_i = |\{n_{ij} : 1 \le j \le N_i\}|$, $\delta_i = t - \tau_{iN_i}$, and

$$\ell_{ij} = \log\left(1 + \frac{\tau_{ij} - \tau_{i,j-1}}{\sigma}\right) \quad (2 \le j \le N_i).$$

For $N_i \ge 2$, $\mu_i^{\log\Delta} = (N_i - 1)^{-1} \sum_{j=2}^{N_i} \ell_{ij}$ and $s_i^{\log\Delta} = \sqrt{(N_i - 1)^{-1} \sum_{j=2}^{N_i} (\ell_{ij} - \mu_i^{\log\Delta})^2}$. We use $\rho_i^{\mathrm{rep}} = 1 - U_i/N_i$ and $\rho_i^{\mathrm{out}} = N_i^{-1} \sum_{j=1}^{N_i} d_{ij}$. Let σ be the median positive difference between distinct training timestamps, or 1 if none exists, and $W = \max(10\sigma, 1)$. With $A_i = |\{j : t - W \le \tau_{ij} < t\}|$ and $A_i^{\mathrm{prev}} = |\{j : t - 2W \le \tau_{ij} < t - W\}|$,

$$g_i = \frac{A_i - A_i^{\mathrm{prev}}}{A_i + A_i^{\mathrm{prev}} + 1}.$$

For $N_i = 0$, all nine entries are zero; for $N_i = 1$, the two gap moments are zero. We estimate a per-dimension mean $\bar{b}_r$ and standard deviation s_r from source and destination states at up to 25,000 training events, and use $\widetilde{b}_{ir}(t) = (b_{ir}(t) - \bar{b}_r)/\widehat{s}_r$, where $\widehat{s}_r = s_r$ when $s_r \ge 10^{-6}$ and $\widehat{s}_r = 1$ otherwise.

Expert inputs. The role expert uses Eq. 6. The history expert receives the candidate latent state, the attended historical-partner latent state, their elementwise product, and the scaled attention score in Eq. 9. Repeat and reciprocity each receive a log count and exponentially decayed recency, with zero recency for an absent pair. Closure receives a log path count and path freshness when enabled. Periodicity receives the direct log count and Eq. 12. Each non-role expert is a two-layer GELU MLP. The router receives source protocol context, candidate behavior latent, and pair statistics.

Closure freshness. Let $\mathcal{P}_{uv}(t)$ contain the supported two-edge shared-neighbor paths or bipartite three-edge paths formed from recent-neighbor records. For each path P, let τ_e be the latest recorded time of its edge e. The implementation uses

$$T_{uv}^{\mathrm{clo}} = \max_{P \in \mathcal{P}_{uv}(t)} \min_{e \in P} \tau_e, \qquad r_{uv}^{\mathrm{clo}} = \exp\left(-\frac{t - T_{uv}^{\mathrm{clo}}}{10\sigma}\right).$$

This measures the freshness of the path with the most recent oldest edge. Set both closure inputs to zero when $\mathcal{P}_{uv}(t)$ is empty.

Main-experiment configuration. All main results use `full_match` with hidden width $d = 64$, $R = 16$ roles, token-component width $d_p = 32$, history length $L = 30$, and dropout 0.1. We train with AdamW (learning rate 10^{-3}, weight decay 10^{-5}), one sampled training negative per positive, gradient-norm clipping at 5, and a budget of 30 epochs with patience 5 on validation MRR. Batch sizes are listed in Table 7. The role-ranking, router-balance, and router-entropy coefficients are 0.03, 0.08, and -0.001; the identity-residual log-scale and router-role bias are initialized to -4 and 0, respectively.

B Full Training Objective

For a minibatch of B positive events and K sampled negatives per event, let $\mathcal{C}_b = (v_b^+, v_{b1}^-, \ldots, v_{bK}^-)$ retain candidate multiplicities. The ranking term in the main text is denoted by $\mathcal{L}_{\mathrm{BPR}}$. The auxiliary role-ranking term has the same form after replacing s by s_{role}:

$$\mathcal{L}_{\text{role-rank}} = -\frac{1}{BK} \sum_{b=1}^{B} \sum_{k=1}^{K} \log \operatorname{sigmoid}\big(s_{\mathrm{role}}(u_b, v_b^+, t_b) - s_{\mathrm{role}}(u_b, v_{bk}^-, t_b)\big).$$

Let $\mathbf{u}_R = (1/R, \ldots, 1/R)$ and let the mean role usage over the B sources and $B(K+1)$ candidates be

$$\bar{\mathbf{p}} = \frac{1}{B(K+2)} \sum_{b=1}^{B} \left(\mathbf{p}_{u_b}(t_b) + \sum_{v \in \mathcal{C}_b} \mathbf{p}_v(t_b) \right).$$

Writing $H(\mathbf{r}) = -\sum_j r_j \log r_j$, the role regularizers are $D_{\mathrm{KL}}(\bar{\mathbf{p}} \| \mathbf{u}_R)$ and the mean entropy over the same role distributions, denoted $\overline{H}_{\mathrm{role}}$. For the router, define

$$\bar{\pi}_m = \frac{1}{B(K+1)} \sum_{b=1}^{B} \sum_{v \in \mathcal{C}_b} \pi_m(u_b, v, t_b), \qquad q_m = \frac{\sum_{b=1}^{B} \sum_{v \in \mathcal{C}_b} a_m(u_b, v, t_b)}{\sum_{j \in \mathcal{M}} \sum_{b=1}^{B} \sum_{v \in \mathcal{C}_b} a_j(u_b, v, t_b)}.$$

The corresponding mean router entropy is

$$\overline{H}_{\mathrm{route}} = \frac{1}{B(K+1)} \sum_{b=1}^{B} \sum_{v \in \mathcal{C}_b} H(\boldsymbol{\pi}(u_b, v, t_b)).$$

The complete optimized objective is

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\mathrm{BPR}} &+ \lambda_{\mathrm{aux}} \mathcal{L}_{\text{role-rank}} + \lambda_{\mathrm{rb}} D_{\mathrm{KL}}(\bar{\mathbf{p}} \| \mathbf{u}_R) + \lambda_{\mathrm{re}} \overline{H}_{\mathrm{role}} \\ &+ \lambda_{\mathrm{pb}} D_{\mathrm{KL}}(\bar{\boldsymbol{\pi}} \| \mathbf{q}) + \lambda_{\mathrm{pe}} \overline{H}_{\mathrm{route}}. \end{aligned} \tag{19}$$

For the main configuration, $(\lambda_{\mathrm{aux}}, \lambda_{\mathrm{rb}}, \lambda_{\mathrm{re}}, \lambda_{\mathrm{pb}}, \lambda_{\mathrm{pe}}) = (0.03, 0.01, 0.001, 0.08, -0.001)$. The role terms encourage balanced global usage and sharp per-node assignments; the negative router-entropy coefficient favors broader allocations over supported experts. The target $\mathbf{q}$ makes the balance penalty relative to observed support frequency. It is a soft preference, while Eq. 15 enforces support exactly. For numerical stability, the implementation clips mean usages and mean availabilities below 10^{-8} and renormalizes each before computing KL; entropy evaluation uses the same probability floor.

C EXPERT EVIDENCE SPECIFICATION

Table 5 lists the query-time inputs for each signature. Closure is disabled in the main experiments. Section 4.6 includes the shared encoder, router, and expert costs. Here $K_{uv} = n_{u\to v}(t)$ and $\ell_\Delta = \log(1+\Delta)$. Lookup costs assume node and pair indices are already constructed; role inputs share the behavior and protocol computations described above.

Table 5: Evidence specification for each expert. All quantities are computed from $\mathcal{H}_t$. "Lookup" is the retrieval cost before the shared neural scoring layers.

Expert	Score-relevant evidence	Availability condition	Lookup
Role	$\mathbf{d}_u(t), \mathbf{p}_v(t)$	always	$O(1)$
History	$\mathbf{z}_v(t)$ and $\{\mathbf{z}_{w_\ell}(t_\ell^-)\}_{\ell=1}^L$	$\lvert\mathcal{S}_u(t)\rvert > 0$	$O(L\ell_\Delta)$
Repeat	$\log(1+n_{u\to v}), \exp(-\Delta_{u\to v}/(10\sigma))$	$n_{u\to v} > 0$	$O(\ell_\Delta)$
Reciprocity	$\log(1+n_{v\to u}), \exp(-\Delta_{v\to u}/(10\sigma))$	$n_{v\to u} > 0$	$O(\ell_\Delta)$
Closure (optional)	path count and oldest-edge freshness	$c_{uv} > 0$	$O(H+\ell_\Delta)$ / $O(H(H+\ell_\Delta))$
Periodicity	$\log(1+n_{u\to v}), p_{uv}^{\mathrm{per}}(t)$	$n_{u\to v} \geq 2$	$O(\ell_\Delta + K_{uv})$

D STRICT-TIME QUERY CONSTRUCTION

For every query (u, v, t), all node and pair histories are accessed with a left binary search at t. Thus no event at timestamp t enters query-time behavior, protocol, or pair evidence. The feature normalizer is fitted once from training-derived states and then fixed for evaluation. Data splitting chooses boundaries only between distinct timestamp groups. Training negatives exclude simultaneous positive destinations. Evaluation uses the dataset-specific CRAFT sampler described in Table 7; supplied fixed negatives are checked against simultaneous positives.

1. Retrieve u's last L events from $\mathcal{H}_t$.
2. Compute normalized behavior states for u, every candidate, and each historical partner at its own event time t_ℓ^-.
3. Encode source protocol and next-partner role demand.
4. Build pair evidence and the signature-availability vector.
5. Score experts, set unavailable router logits to $-\infty$, and compute Eq. 16.
6. Add the optional identity residual and optimize candidate ranking.

E STRICT-HISTORY PROPERTIES

Proposition 1 (query-time isolation). Fix the query, node encoding, graph type, history tie order, training-fitted time scale and normalization, and model parameters. In evaluation mode, the query-dependent behavior states, protocol tokens, pair evidence, and availability vector depend on the event stream only through $\mathcal{H}_t$. Two streams with identical strict prefixes therefore give identical scores for the same candidate at t.

Proof. Left binary search at t returns exactly the records with timestamp smaller than t; recent-history retrieval takes a suffix of that prefix. Behavior counts, gap moments, and window statistics use only these records. Each historical-partner role is evaluated at $t_\ell^- < t$ with the same boundary rule. Pair counts, recencies, periodic gaps, closure paths, and availability predicates likewise use strict prefixes. With preprocessing and parameters fixed, all subsequent neural operations and the identity residual are identical. The same argument holds in training mode conditional on the same dropout masks. □

This property treats a timestamp group as simultaneous. In particular, two events sharing a timestamp cannot provide one another's behavior, repeat, reciprocity, or periodicity evidence. Training-only normalization is fixed before validation and test evaluation; it is not recomputed from validation or test events.

Proposition 2 (availability invariance). Fix a query, its availability vector, and finite expert scores and router logits. Let $\mathcal{A} = \{m \in \mathcal{M} : a_m = 1\}$, which is nonempty because $a_{\mathrm{role}} = 1$. The mixture

score in Eq. 16 is

$$s_{\text{mech}}(u,v,t) = \log \sum_{m\in\mathcal{A}} \exp\big(g_m(u,v,t) + s_m(u,v,t)\big) - \log \sum_{m\in\mathcal{A}} \exp\big(g_m(u,v,t)\big).$$

Holding the supported logits and scores fixed, changes to any unavailable expert's logit or score leave s_{mech} unchanged.

Proof. Substitute Eq. 15 into Eq. 16. Terms with $a_m = 0$ vanish, leaving the two log-sum-exp expressions over $\mathcal{A}$. Neither contains an unavailable expert's logit or score. □

This is a coordinate-level property of the mixture. Candidate features can also affect supported logits through the shared router. The propositions characterize the information boundary and scoring computation.

Mixture bounds and score sensitivity. Since the supported weights sum to one, $\exp(s_{\text{mech}})$ is a weighted mean of $\exp(s_m)$. Hence

$$\min_{m\in\mathcal{A}} s_m \le s_{\text{mech}} \le \max_{m\in\mathcal{A}} s_m.$$

Define $\omega_m = \pi_m \exp(s_m)/\sum_{j\in\mathcal{A}} \pi_j \exp(s_j)$. Direct differentiation of the two log-sum-exp terms gives

$$\frac{\partial s_{\text{mech}}}{\partial s_m} = \omega_m, \qquad \frac{\partial s_{\text{mech}}}{\partial g_m} = \omega_m - \pi_m \quad (m \in \mathcal{A}).$$

Both derivatives are zero for unavailable coordinates. These partial derivatives hold other scores and logits fixed. They show how expert score and routing mass jointly determine local score sensitivity, while π_m alone describes allocation. The bounds apply to the mixture before adding the identity residual.

F Evaluation, Ties, and Model Selection

For a ranking query q with positive score s_q^+ and sampled negative scores $\{s_{qk}^-\}_{k=1}^K$, we use the CRAFT/TGB-Seq tie convention

$$r_q = 1 + \frac{1}{2}\left(\sum_{k=1}^{K} \mathbb{I}[s_{qk}^- > s_q^+] + \sum_{k=1}^{K} \mathbb{I}[s_{qk}^- \ge s_q^+]\right), \qquad \text{MRR} = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \frac{1}{r_q}.$$

Thus a tied negative contributes half a rank position. AP sums precision weighted by recall increments at distinct score thresholds; ROC-AUC uses average ranks of positive and negative binary examples. Ranking uses 100 sampled negatives plus the positive (101 entries) under Table 7. AP and ROC-AUC use one negative sampled under the same dataset-specific rule per positive.

The checkpoint with the largest validation MRR is retained. Training stops after five consecutive validation checks without improvement, subject to a 30-epoch budget. The validation candidate construction is deterministic for a run through its fixed seed and sampling rule; the test split is evaluated only after checkpoint selection.

G Ablation Configurations

Variant	Behavior roles	Protocol	Mechanisms	Candidate router
ID-only	–	–	–	–
Role-only	✓	–	–	–
Role+protocol	✓	✓	–	–
Uniform-mechanism	✓	✓	✓	–
No-history-match	✓	✓	role+rep./recip./period.	✓
RuleCast-noID	✓	✓	✓	✓
RuleCast	✓	✓	✓	✓

The full model additionally enables the identity residual; the no-ID model is otherwise identical. Uniform-mechanism normalizes equally over available experts, so it differs from the full no-ID configuration even when the same experts are active.

H DIRECT CONTROLS

We group three direct controls under one evaluation. The CRAFT-like matcher replaces the source-side role–protocol pathway with direct candidate–history matching. Feature-MLP feeds the normalized nine-dimensional behavior state directly to an MLP matcher and removes the role–protocol factorization and mechanism experts. Identity-only retains trainable node identities while removing dynamic evidence. All rows use the same timestamp-group split, candidate sampler, evaluator, and three random seeds; the identity-only control uses the same closure-free configuration.

Table 6: Direct controls: test MRR (%). Entries are mean ± standard deviation over three seeds on the three common datasets. The best value in each column is bold.

Variant	Wikipedia	UCI	Flights
CRAFT-like direct matcher	90.11±0.04	79.07±0.81	91.11±0.09
Feature-MLP	91.92±0.00	86.29±0.30	90.48±0.04
Identity only	46.33±0.12	11.33±0.05	81.33±0.03
RULECAST	**92.19±0.14**	**86.65±0.10**	**91.37±0.10**

I ADDITIONAL EVIDENCE-CONDITIONED VISUAL ANALYSIS

Figures 5 and 6 extend the main-paper Flights analysis to Wikipedia and UCI. Both show the same two quantities: router allocation by available candidate evidence and source-demand overlap for realized next partners versus sampled candidates. The plots provide dataset-level views of the model's conditional scoring behavior.

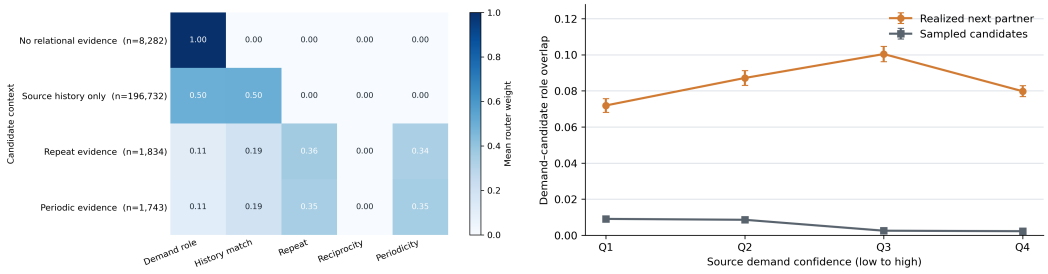

Figure 5: Evidence-conditioned analysis on Wikipedia. **Left:** mean router allocation by candidate context. **Right:** role-overlap for realized next partners and sampled candidates across source-demand-confidence quartiles.

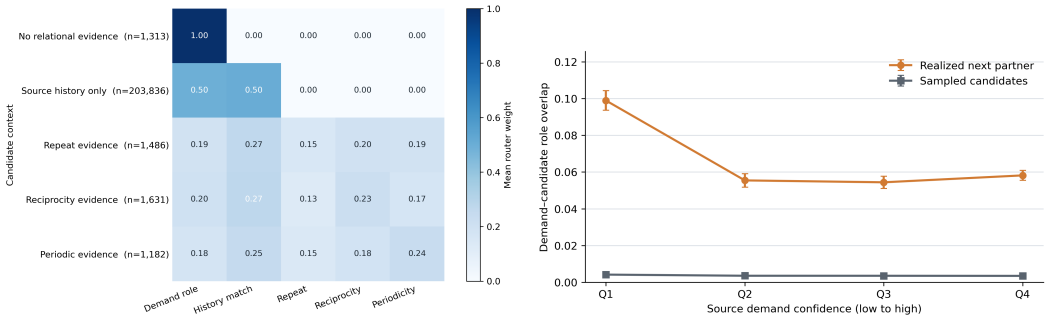

Figure 6: Evidence-conditioned analysis on UCI. **Left:** mean router allocation by candidate context. **Right:** role-overlap for realized next partners and sampled candidates across source-demand-confidence quartiles.

The Wikipedia and UCI panels use the same definitions as the Flights analysis in the main text. Their different allocation profiles reflect the evidence available in each graph, while the consistent separation between realized and sampled candidates indicates that the demand–candidate overlap is evaluated relative to each dataset's own history. We include these panels to make the conditional behavior of the same model visible beyond a single benchmark.

J Additional Demand Diagnostics

The following diagnostics use the same held-out Flights checkpoint as the main evidence analysis. The first plot partitions queries into five equal-count bins by the maximum predicted demand probability. Realized next partners maintain higher demand–candidate role overlap than sampled candidates in every bin, while the ranking metrics are shown as descriptive bin-level measurements. The second plot follows one source selected before plotting by the largest mean adjacent Jensen–Shannon shift in its demand distribution; the marked queries show how the predicted role demand and the routed score components change over time.

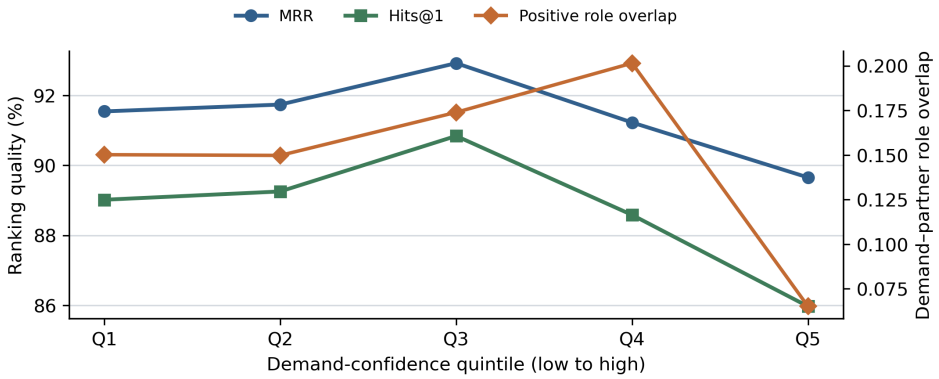

Figure 7: Demand-confidence diagnostic on Flights. Queries are grouped into five equal-count bins by the maximum probability in $\mathbf{d}_u(t)$. The plot reports ranking quality and demand–candidate role overlap for realized next partners and sampled candidates.

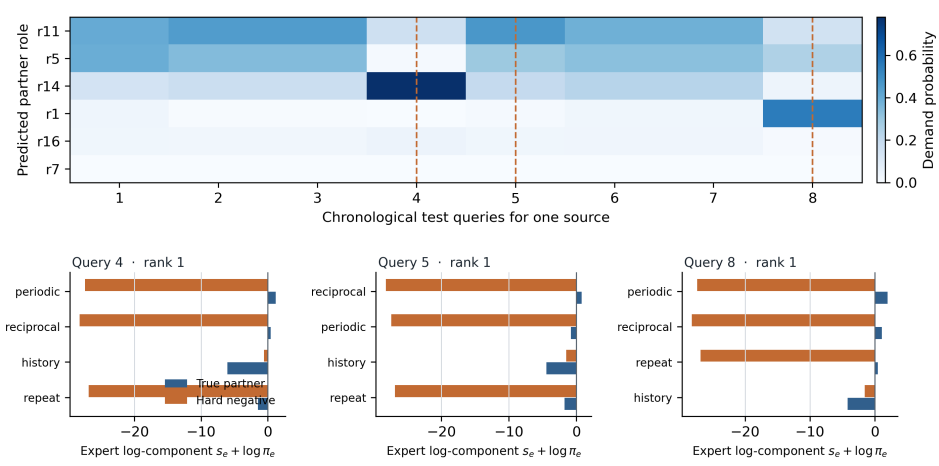

Figure 8: Source-demand evolution and score decomposition on Flights. The heatmap shows the predicted partner-role demand for chronological queries from one source. For three demand-shift events, the lower panels compare the routed expert log-components $s_e + \log \pi_e$ for the true partner and a hard negative; the identity residual is included in the final score but omitted from the bar comparison for readability.

K Dataset-Specific Main-Experiment Settings

The following table records the choices that affect candidate construction. We keep the temporal split, history length, and candidate count fixed across datasets; only the node-space convention and the benchmark-compatible negative sampler vary. This separation lets the main comparison use one model configuration while respecting the evaluator associated with each benchmark family.

Table 7: Dataset-specific main-experiment settings. All datasets use the DyGLib/CRAFT time-quantile split (70%/15%/15%), $L = 30$, and 100 ranking negatives per positive. "Collision" and "random" are the corresponding CRAFT candidate-sampling rules. AP and ROC-AUC use one binary negative per positive.

Dataset	Node spaces	Ranking sampler	Train / eval batch
Wikipedia	bipartite	collision	512 / 512
Reddit	bipartite	collision	512 / 512
UCI	shared	collision	512 / 512
LastFM	bipartite	collision	512 / 512
Flights	shared	collision	512 / 512
ML-20M	bipartite	random	$1024/512$; $512/256$*
YouTube	shared	random	512 / 512
tgbl-review	shared	random	512 / 512
tgbl-flight	shared	random	1024 / 512
tgbl-coin	shared	random	1024 / 512

*ML-20M uses train/evaluation batch sizes $1024/512$ for seed 0 and $512/256$ for seeds 1 and 2.

The collision sampler draws from destination nodes outside the source's strict historical neighbor set, without replacement when the pool is large enough. The random sampler draws from the destination-node pool with replacement. These evaluation samplers retain the CRAFT rules, including possible positive coincidences; training uses the simultaneous-positive filter. Candidate features always use events strictly before the query time.

Table 8: Shared architecture, optimization, and objective settings for the main experiments.

Setting	Value	Setting	Value
Behavior-state dimension	9	Hidden width	64
Number of roles	16	Token-component width d_p	32
History length	30	Dropout	0.1
Optimizer	AdamW	Learning rate	10^{-3}
Weight decay	10^{-5}	Gradient clip	5
Training negatives	1	Ranking negatives	100
Maximum epochs	30	Patience	5
Role auxiliary weight	0.03	Role balance weight	0.01
Role entropy weight	0.001	Router balance weight	0.08
Role temperature	1.0	Router entropy weight	-0.001
ID-scale initialization	-4	Router-role bias initialization	0
Closure expert	off	Behavior-state cache	300,000

The shared settings in Table 8 are applied after the dataset-specific loader has constructed the strict temporal index. In particular, the history length and candidate count are not changed to match a dataset's density. The role temperature, regularization weights, and identity initialization are fixed across the main suite, so differences in the tables are not caused by per-dataset tuning.

For each seed, validation MRR selects the checkpoint before any test query is scored. The bounded behavior-state cache stores repeated (i, t) lookups, while node and pair histories remain sorted for left-boundary retrieval. The batch builder deduplicates candidate nodes within each query and computes pair features directly. These operations preserve the semantics in Appendix E.

The input pipeline queries strict prefixes for each batch. Empty nodes use a zero raw behavior state before normalization; an empty source history uses the learned protocol context. Padding is excluded by the history mask. Feature normalization is fitted once on training-derived states and reused for validation and test. The same state construction and scoring equations apply across the ten datasets.

For reproducibility, a run records the dataset name, split boundaries, sampler type, seed, and selected validation checkpoint together with the final metrics. The reported configuration therefore determines both the numerical objective and the temporal information available to every query.

Boundary cases follow the same rules as ordinary queries. A node with one historical event has no defined gap moments, a pair with one prior interaction has no periodicity support, and an unseen pair receives only the experts whose evidence is available. Events sharing a timestamp are processed as a group, so none can become evidence for another event in that group.

These conventions also apply to validation and test batches. They avoid special-case changes between training and evaluation and make the reported comparisons depend on the model and candidate sampler, rather than on implicit padding or future information.