PREDICTIVE STATE HAS A SCOPE

Anonymous authors
Paper under double-blind review

ABSTRACT

A temporal model can recognize a relation without remembering its history. We study *predictive state scope*: where different histories are written and which states future predictions can read. Shared state pools observations but can merge histories with distinct futures, creating predictive aliasing; local states retain separate histories but receive fewer observations. We introduce *Factor-Addressed Predictive State* (FA-State), a jointly trained neural architecture coupling a shared prediction path with pair-addressed recurrent states. FA-State shares how history is encoded and updated across addresses while preserving where relation-specific history is written and retrieved. Bayes log-risk and random-effects analyses characterize the cost of predictive aliasing and the finite-sample trade-off between shared and local states. Across temporal graph forecasting tasks, FA-State improves prediction with multiple backbones. Holding trained weights and online bank contents fixed, deranging only the retrieval addresses systematically degrades prediction, showing that trained predictors use the correspondence between historical content and its address. Controlled streams and recurrence analyses further reveal how available history shapes the shared–local trade-off. These findings establish state scope as a design choice in temporal representation learning itself: what a model remembers depends on where it writes history and how it reads it back.

1 INTRODUCTION

Where should a temporal model remember an interaction? A source node can trade with one partner, repeatedly collaborate with another, and stop interacting with a third. A single node state must make these histories available to later queries. If its updates merge histories that imply different futures for the same queried partner, identifying that partner at readout is too late: the distinguishing information has already disappeared. This is *predictive aliasing*: the learned compression and access rule jointly determine which historical distinctions remain available to prediction.

Temporal models organize history through node memory, temporal neighborhoods, or structured fact or entity states (Rossi et al., 2020; Xu et al., 2020; Mei et al., 2020; Henaff et al., 2017). Target-aware models supply endpoint identities during updates (TGNv2) or construct pair-conditioned representations at query time (DyGFormer) (Tjandra et al., 2024; Yu et al., 2023). We study the *scope of predictive state*: which histories update a state and which states a query can read. FA-State couples this routing with shared learning rules and separately stored relation states. If an accessible representation merges histories with different future distributions, that merger has an exact cost under Bayes log-loss.

Keeping relation histories separate introduces a second question: how much evidence does each address receive? Local state can retain predictive detail that sharing loses, but sparse and unseen relations still need pooled information. The design challenge is to retain local predictive evidence alongside shared prediction. A random-effects analysis makes this estimation trade-off explicit and motivates a controlled study that varies relation-specific dependence and recurrence independently.

Factor-Addressed Predictive State (FA-State) separates parameter sharing from state sharing (Figure 1). It shares how histories are encoded and updated, while keeping their evolving states address-specific. Each candidate retrieves its directed-pair history and combines its contribution with the shared score. The shared route pools evidence across relations and supplies predictions for unseen addresses. Both routes learn jointly from the prediction objective.

Our contributions are threefold:

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.

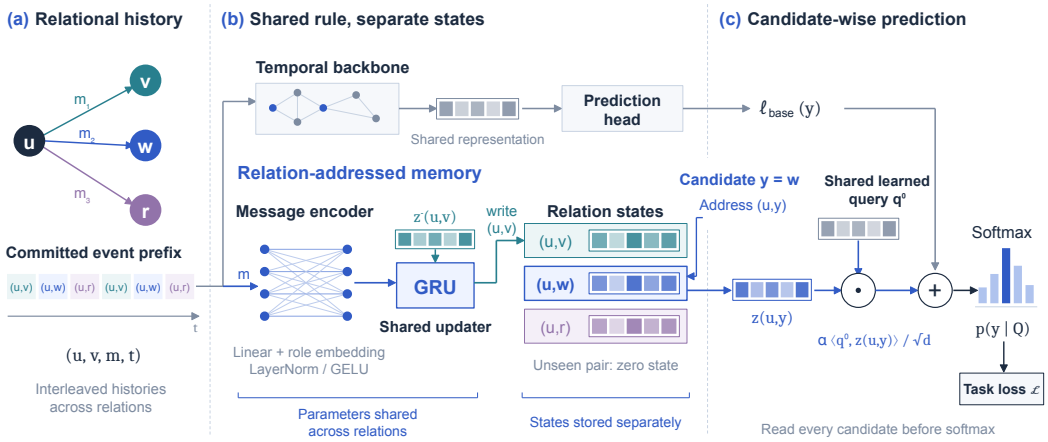


Figure 1: **FA-State combines shared prediction with pair-specific recurrent state.** (a) Observed events form a committed prefix. (b) A message encoder and GRU share parameters across addresses, while each directed pair retains its own state; an event at (u, v) updates that pair. (c) Candidate y reads $z_{(u,y)}$, illustrated with $y = w$. In this online node-affinity realization, a shared learned vector q^0_η scores the retrieved content. Addressed and backbone logits combine before softmax and learn jointly. Unseen addresses return zero. Colors identify relation histories; vector entries and output bars are schematic. Section 3 describes task realizations.

1. We identify *predictive state scope* as a representation-design axis for relational event streams, connect predictive aliasing to Bayes log-risk, and characterize a shared–local estimation trade-off.
2. We introduce FA-State, a jointly trained shared-plus-addressed architecture with shared history encoders, directed-dyad states, and candidate-wise retrieval.
3. We demonstrate the predictive value of addressed history across temporal backbones and tasks, establish trained predictors' use of content–address correspondence, and localize addressing gains by historical recurrence.

2 Predictive State Scope

State scope raises two questions: which predictive distinctions survive compression, and how much evidence supports each state. We examine them through Bayes log-risk and a finite-sample shared–local estimation model.

2.1 Relational Histories and Query Addresses

Let $H_t = (e_1, \ldots, e_{t-1}) \in \mathcal{H}$ be a strict event prefix, with $e_i = (u_i, v_i, m_i, \tau_i) \in \mathcal{E}$ recording source, destination, message, and timestamp. A query specifies an address $A \in \mathcal{A}$ and a future outcome Y_A.

Definition 1 (State scope)**.** *A state scope is $\mathcal{S} = (\mathcal{M}, \sigma_{\mathrm{w}}, \sigma_{\mathrm{r}})$: $\mathcal{M}$ is the memory-address universe, $\sigma_{\mathrm{w}} : \mathcal{E} \to 2^{\mathcal{M}}$ assigns each event the addresses it updates, and $\sigma_{\mathrm{r}} : \mathcal{A} \to 2^{\mathcal{M}}$ assigns each query the addresses it can read. Source and directed-dyad scopes use $\{u\}$ and $\{(u, v)\}$, respectively. Set-valued routes allow shared and relational states to coexist.*

A scope and its learned updates induce an accessible representation $R_{\phi,\mathcal{S}}(H_t, A)$, abbreviated R_ϕ for a fixed scope. Histories with the same accessible representation for a query form an equivalence cell. *Predictive aliasing* occurs when the merged histories retain predictive distinctions: $Y \not\perp H \mid R_\phi(H, A), A$. For shared state $C = \phi(H_t)$, retaining C while adding fine-scope content gives the refinement $\Psi = (C, Z_f)$.

2.2 BAYES RISK OF AN ALIASED REPRESENTATION

Let $\mathcal{R}^*(X)$ denote the minimum expected log loss achievable by an unrestricted readout conditioned on X.

Proposition 1 (Aliasing risk)**.** *For deterministic $C = \phi(H)$ and finite relevant conditional entropies,*

$$\mathcal{R}^*(C, A) - \mathcal{R}^*(H, A) = I(Y; H \mid C, A) \tag{1}$$

$$= \mathbb{E}[D_{\mathrm{KL}}(p(Y \mid H, A) \,\|\, p(Y \mid C, A))] \geq 0. \tag{2}$$

The gap is zero if and only if the representation's equivalence cells are predictively homogeneous almost surely.

Equation 2 turns a standard proper-scoring identity (Gneiting & Raftery, 2007) into a diagnostic of state scope by connecting an access rule to the future distributions it merges. Predictive-state representations likewise characterize history through future predictions (Littman et al., 2001; Singh et al., 2004); our focus is the relational index at which that history is stored and retrieved. Because $\Psi = (C, Z_f)$ retains the shared state, it representation-refines C and weakly lowers population Bayes log-risk. Shared-only and scoped-only designs remain incomparable unless one representation reconstructs the other, and recurrence governs their finite-sample trade-off.

Let $Z_A = \psi(H, A)$ denote content retrieved from the correct address. Then

$$\mathcal{R}^*(C, A) - \mathcal{R}^*(C, Z_A, A) = I(Y; Z_A \mid C, A). \tag{3}$$

If an occupancy variable O_A records whether the address has appeared in the prefix, the content beyond existence is

$$I(Y; Z_A \mid C, O_A, A). \tag{4}$$

Conditioning on O_A removes information already available from address existence. Comparing correct content with $Z_{\pi(A)}$ under a prescribed derangement rule π probes the value of binding content to its address. Correlated addresses may leave $Z_{\pi(A)}$ predictively informative, so the comparison concerns the incremental value of the correct correspondence.

We study this correspondence through two complementary comparisons. Separately trained access rules assess the predictive value of a design; frozen-model interventions retain the weights and state bank and change only retrieval to reveal how a trained predictor uses it (Section 5.3).

2.3 THE FINITE-SAMPLE SHARED–SCOPED FRONTIER

Finer state scope can lower population aliasing while reducing the number of observations available per scope. A stylized random-effects model makes the trade-off explicit. Let an address have a latent deviation $\theta_a \sim \mathcal{N}(0, \tau^2)$ and observations

$$X_{a,i} = \mu + \theta_a + \epsilon_{a,i}, \qquad \epsilon_{a,i} \sim \mathcal{N}(0, \sigma^2), \tag{5}$$

with known shared mean μ and iid historical and future noises independent of θ_a. After $n_a > 0$ observations, the posterior-mean predictor of $X_{a,*} = \mu + \theta_a + \epsilon_{a,*}$ is

$$\widehat{X}_{a,*} = \mu + w_a(\bar{X}_a - \mu), \qquad w_a = \frac{n_a \tau^2}{\sigma^2 + n_a \tau^2}.$$

For an unseen address, $\widehat{X}_{a,*} = \mu$.

Proposition 2 (A stylized shared–local frontier)**.** *Let $R(\widehat{X}) = \mathbb{E}[(X_{a,*} - \widehat{X})^2]$, with expectation over the address effect, observed samples, and future noise. For $n_a > 0$, the unshrunk local mean beats the known shared mean if and only if $n_a \tau^2 > \sigma^2$. The shrinkage predictor strictly dominates both when $n_a > 0$ and $\tau^2, \sigma^2 > 0$, and recovers the shared predictor when $n_a = 0$.*

Here τ^2 is static address heterogeneity and n_a the effective number of observations available to estimate it. This stylized model isolates the estimation side of the frontier: heterogeneity favors local state, recurrence makes it reliable, and shrinkage interpolates between local and shared prediction. The threshold compares sharing with an *unshrunk* local mean; Section 4 adds the dynamic-dependence axis. Complete risks and proofs appear in Appendix A.

3 FACTOR-ADDRESSED PREDICTIVE STATE

3.1 WRITING RELATION HISTORY

FA-State combines shared prediction with states organized by relational addresses. For a prediction query Q and candidate outcome y, the factor address $a(Q, y)$ identifies the history exposed to that candidate. Our directed-dyad realizations use $a(Q, y) = (u, y)$: each candidate reads the history of its own source–destination pair.

An active address a maintains an episodic state $z_a \in \mathbb{R}^{d_a}$. In the online realization, a learned message projection and GRU update the state after the associated events become available:

$$z_a^+ = U_\eta\big(z_a^-, g_\eta(m_a, \mathrm{role}(a))\big)\,, \tag{6}$$

In the online realization, m_a is the address's last message in the committed batch; $\mathrm{role}(a)$ denotes a source, destination, or directed-pair role. The updater shares parameters across addresses, and its activations encode each address's observed history. An unseen address reads as zero. Query scoring precedes the commitment of the current batch, and the candidate mask follows the same committed prefix.

3.2 READING CANDIDATE-SPECIFIC STATE

Candidate-wise retrieval connects each outcome to its history before the task softmax, making addressed evidence part of prediction formation. For a query Q with candidate set $\mathcal{Y}_Q$, the addressed route exposes the candidate-indexed collection

$$\mathcal{Z}_Q = \big(z_{a(Q,y)}\big)_{y \in \mathcal{Y}_Q}.$$

For the dyad-only realization, the query-level accessible representation is $(C, Q, \mathcal{Y}_Q, \mathcal{Z}_Q)$: candidate identities remain associated with their retrieved states. Hybrid variants additionally expose their entity states. Retaining C ensures that adding addressed content preserves the information available to shared prediction, conditional on the query and candidate set.

The candidate logit combines shared prediction with addressed content:

$$\begin{aligned}\ell(y \mid H, Q) = \ell_{\mathrm{base}}(y \mid C, Q) &+ \frac{\alpha_a}{\sqrt{d_a}}\langle q_\eta(Q, y), z_{a(Q,y)}\rangle \\ &+ \frac{\alpha_e}{\sqrt{d_a}}\langle q_\eta(Q, y), z_{Q,y}^{\mathrm{entity}}\rangle,\end{aligned} \tag{7}$$

where q_η is normalized and the gains are nonnegative learned scalars. The entity term is enabled for entity or hybrid variants.

For node-affinity prediction, we use a state-only readout: $q_\eta(Q, y) = q_\eta^0$ is one learned vector shared across sources and candidates. All candidate-specific information in this branch comes from the addressed history; TGN, TGAT, or TGNv2 supplies the backbone logits. For temporal-KG prediction, $Q = (u, r, t)$ specifies a source, relation type, and time. The address remains (u, y), and q_η is projected from LogCL's decoder query. A recurrent encoder summarizes that directed pair's history of relation-type tokens. Node-affinity prediction updates an online bank; temporal-KG prediction recomputes these summaries from stored historical snapshot bags.

3.3 JOINT PREDICTION AND LEARNING

Both routes learn jointly through the combined logits in Equation 7. Node-affinity variants use the same conditional cross-entropy within each comparison. The LogCL realization retains its static-graph encoder and jointly optimizes entity prediction and contrastive learning. Appendix B specifies the batching, message aggregation, gradient truncation, and task-specific state encoders.

4 CONTROLLED DEPENDENCE AND SHARED BACKOFF

4.1 DESIGN

We predict binary marks in directed event streams. Before symmetric noise, a seen pair's next mark uses its previous mark with probability δ and an evolving source-level bit otherwise; unseen pairs use

the shared bit. Thus δ controls pair-specific dependence, while ρ controls resampling of previously observed pairs. Models receive only preceding observed events, not generator latents or axis values.

The four-way grid compares shared-only, scoped-only, shared-plus-correct-scoped, and shared-plus-wrong-scoped predictors on identical episodes at the same encoder width. Wrong-scoped retrieval uses an episode-fixed derangement of the pair space; unobserved mapped pairs have empty history. Correct and wrong predictors use global-scalar and input-dependent gates, respectively. Appendix B.1 specifies the generator, grid, and readouts.

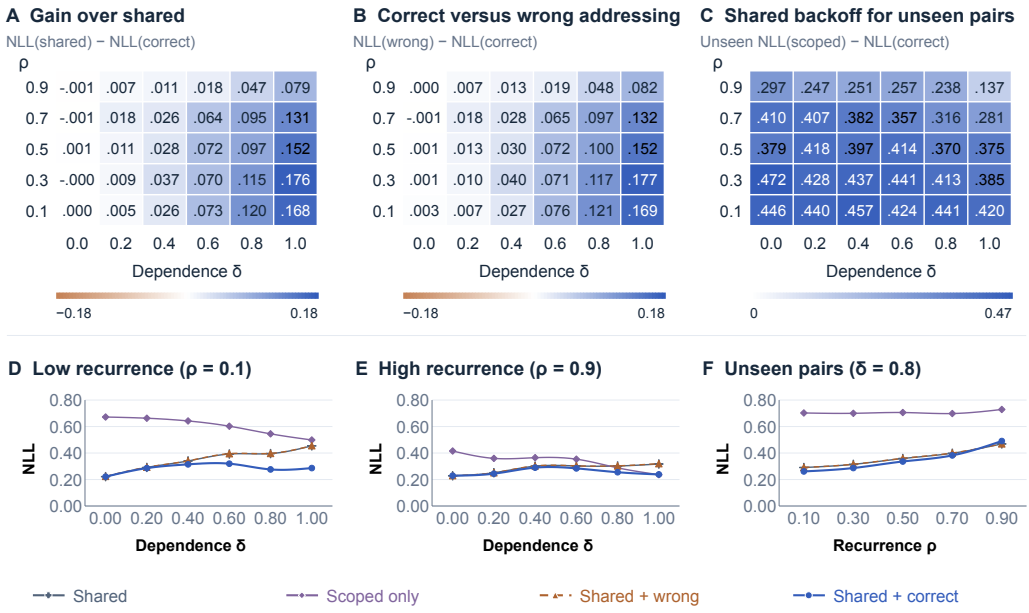


Figure 2: **The controlled stream exposes both sides of the scope trade-off.** A–C show NLL gains over shared or wrong-scoped prediction and shared backoff for unseen pairs; D–F show absolute NLL at the labeled settings. Values average three seeds. Correct and wrong predictors use global-scalar and input-dependent gates, respectively.

4.2 Local signal and pooled evidence

In the four-way comparison (Figure 2), correct shared-plus-scoped state beats shared-only and wrong-address state in all six cells with $\delta \geq .6$, $\rho \geq .7$, by 0.0724 and 0.0738 mean NLL, respectively. At $\delta = 0$, shared-minus-correct is -0.00033: pair history brings little gain without direct predictive dependence.

With a scalar gate in both readouts, correct addressing also wins in all 75 seed–cell comparisons with $\delta > 0$. The mean wrong-minus-correct NLL gap is 0.0730 in the high-dependence, high-recurrence region and 0.00055 at $\delta = 0$ (Appendix B.2).

Shared backoff improves unseen-pair NLL over scoped-only in all six high-dependence, low-recurrence cells, by 0.421 on average. At high recurrence, the local route becomes more competitive as dependence grows, while shared prediction supports unseen pairs. Recurrence changes available history; the correct–wrong gap need not grow monotonically. This dynamic grid complements Proposition 2 without estimating its static Gaussian threshold.

Across widths 8–48, shared-plus-correct retains a 0.060–0.079 NLL gain over shared-only (Appendix J). With matched scalar gates, the wrong-minus-correct gap is 0.0587–0.0805, positive in all 12 seed–width comparisons.

Table 1: Controlled state-scope results (NLL), averaged over three seeds and all grid cells in each region. High dependence: $\delta \geq .6$; low/high recurrence: $\rho \leq .3$ / $\rho \geq .7$.

Regime / evaluation	Shared	Scoped	Wrong addr.	Correct addr.
$\delta = 0$ / all queries	**.2317**	.6010	.2328	.2320
High δ, low ρ / unseen	.3171	.7046	.3182	**.2839**
High δ, high ρ / all	.3533	.3459	.3547	**.2809**

5 Experiments on Real Relational Streams

5.1 Tasks and evaluation

We first study future partner-distribution prediction on `tgbn-trade` from TGB (Huang et al., 2023), then study future genre-distribution prediction on `tgbn-genre`. The primary evaluation measures prediction among destinations already observed with the source in its causal prefix. This occupancy-matched candidate set focuses the comparison on historical content among known relations. We optimize conditional cross-entropy on this set and report matched NDCG@10, conditional cross-entropy, and NDCG@10 over the full candidate set. The event-stream backbones are recurrent-memory TGN (Rossi et al., 2020) and temporal-attention TGAT (Xu et al., 2020), using the TGM implementations (Chmura et al., 2025). We then evaluate temporal knowledge-graph forecasting on ICEWS18 with LogCL (Chen et al., 2024), using raw and filtered ranking metrics. Unless stated otherwise, learned results average three seeds; tables report mean $\pm$ sample standard deviation.

5.2 FA-State improves prediction across temporal backbones

FA-State improves partner-distribution prediction with both recurrent-memory TGN and temporal-attention TGAT (Table 2). Over their shared-only baselines, the matched-NDCG gains are 0.0601 and 0.0462, and the full-candidate gains are 0.0491 and 0.0406. Figure 3 summarizes the cross-setting gains.

We compare FA-State (dyad) with an occupancy route supplying an existence marker and a wrong-address control retrieving another observed partner's content from the same source. FA-State and the wrong-address control match architecture, parameter count, objective, and selection rule, and are trained separately.

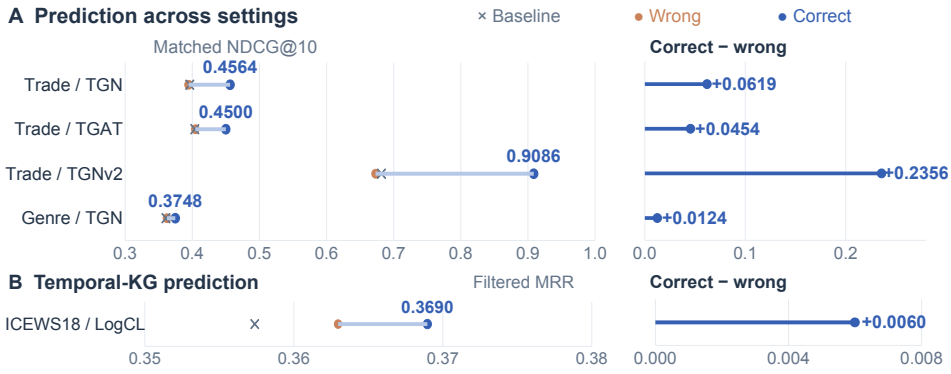


Figure 3: **FA-State improves prediction across settings.** Three-seed means and correct-minus-wrong gains for separately trained access rules. "Correct" denotes FA-State; "Wrong" denotes its wrong-address control. A uses matched NDCG@10 and B filtered MRR, on separate scales.

The matched-NDCG gains over wrong addressing are 0.0619 for TGN and 0.0454 for TGAT, with positive effects in every seed and test year. Occupancy and wrong-address state remain close to their corresponding baselines. Conditional cross-entropy also improves, indicating gains in both partner ordering and the predicted distribution across the two backbone families.

Table 2: Future partner-distribution prediction on `tgbn-trade`. "Matched" conditions on causally observed partners. † uses previous ground-truth target distributions rather than raw event messages.

Backbone	Variant	Matched NDCG↑	Cond. CE↓	Full NDCG↑
TGN	Baseline	.3963±.0055	4.2042±.0549	.3671±.0027
	Occupancy	.3932±.0040	4.2197±.0385	.3686±.0042
	Wrong-address control	.3945±.0032	4.2127±.0396	.3687±.0066
	FA-State (dyad)	**.4564±.0079**	**4.0264±.0152**	**.4162±.0190**
TGAT	Baseline	.4038±.0049	4.2044±.0373	.3824±.0073
	Occupancy	.3972±.0077	4.1935±.0264	.3759±.0067
	Wrong-address control	.4046±.0059	4.1977±.0543	.3836±.0047
	FA-State (dyad)	**.4500±.0074**	**4.0768±.0060**	**.4229±.0071**
TGNv2	Baseline	.6818±.0045	3.7436±.0842	.6405±.0043
	Wrong-address control	.6730±.0060	3.7441±.1191	.6382±.0052
	FA-State (dyad)	**.9086±.0052**	**2.7493±.0743**	**.8982±.0071**
Label history†	Moving average	.8287	4.1078	.8232
	Persistence	.8668	5.2333	.8562

Using previous ground-truth target distributions, moving average and persistence reach matched NDCG@10 of 0.8287 and 0.8668, respectively; FA-State with TGNv2 reaches 0.9086 from raw event messages.

Identity-aware updates. An independent implementation of TGNv2 (Tjandra et al., 2024) uses its paper-reported `tgbn-trade` configuration, with learned cosine identification of both endpoints in every memory update. All three variants share the same TGNv2 backbone, matched objective, optimizer, candidate set, and validation rule. FA-State and its wrong-address control additionally match factor architecture and parameter count. Correct addressing improves matched NDCG@10 over wrong addressing by 0.2356 and over TGNv2 alone by 0.2269, with positive effects in all three seeds and all nine seed–test-year cells. The hierarchical seed/source intervals are $[0.2101, 0.2631]$ and $[0.2018, 0.2532]$, respectively. Appendix F reports the per-seed results and configuration.

The improvement spans most test queries: Figure 4A shows a broad shift above the diagonal when comparing correct and wrong addressing for each matched source–year query, with scores averaged across runs.

5.3 The trained predictor depends on correct retrieval

To isolate a trained predictor's use of content–address correspondence, we intervene on retrieval without retraining. For each of three 100-dimensional identity-aware TGN models, we freeze the weights, reconstruct each model's online state bank from its observed prefix, and score the same test queries with correct and deranged retrieval. Backbone logits and bank contents are identical between these two predictions. This compact identity-aware configuration is distinct from the TGNv2 comparison in Table 2.

Changing only the read address lowers matched NDCG@10 by 0.0519 on average, with positive effects in every seed and all nine seed–test-year cells (Figure 4B). The paired 95% interval is $[0.0452, 0.0589]$. Deranged retrieval also increases conditional cross-entropy by 0.1151, with interval $[0.1050, 0.1252]$. Five predetermined derangement rules give mean NDCG effects between 0.0508 and 0.0525. The learned predictor therefore uses the correspondence between historical content and its retrieval address. Appendix D specifies the intervention and its sensitivity analysis.

5.4 FA-State improves genre and temporal-KG prediction

FA-State also improves prediction of a user's future distribution over music genres on `tgbn-genre`. We train shared-only TGN and parameter-matched correct- and wrong-address variants on this interaction system (Table 3). Correct addressing improves the primary matched NDCG@10 by 0.0139 over the baseline and by 0.0124 over wrong addressing; both paired effects are positive in all three seeds. Paired source/seed resampling gives intervals $[0.0010, 0.0336]$ and $[0.0011, 0.0322]$,

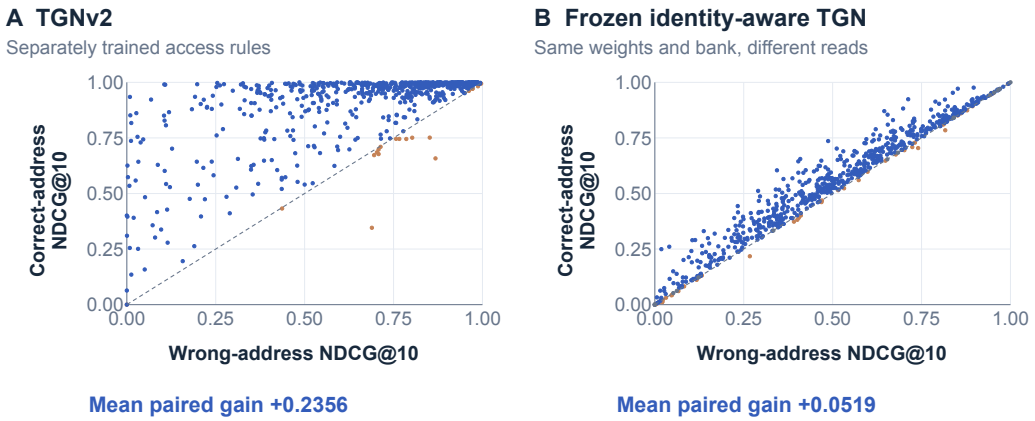


Figure 4: **Correct addressing improves queries and matters to the trained predictor.** Each point shows three-run mean NDCG@10 for one of 664 matched trade queries. A compares separately trained TGNv2 access rules; B changes only retrieval in compact identity-aware TGN, holding weights, bank contents, and backbone logits fixed. Blue/orange points favor correct/wrong addressing.

respectively (Appendix C.1). Conditional cross-entropy also favors correct over wrong addressing in all three seeds, by 0.0395 on average. Full-candidate NDCG also improves in mean by 0.0091, with greater seed variability.

Table 3: Future genre-distribution prediction on `tgbn-genre`.

Variant	Matched NDCG↑	Cond. CE↓	Full NDCG↑
Baseline	.3608±.0002	4.2142±.0023	.3489±.0017
Wrong-address control	.3623±.0017	4.2106±.0059	.3519±.0024
FA-State (dyad)	**.3748±.0177**	**4.1711±.0529**	**.3580±.0182**

Temporal-KG forecasting. FA-State improves temporal-KG forecasting with LogCL on ICEWS18 (Table 4). Its dyadic state improves filtered MRR over baseline by 0.0116 and over wrong addressing by 0.0060; the latter difference is positive in all three seeds. Combining entity and correctly addressed dyad state obtains the best mean across all reported metrics: filtered MRR rises from 0.3574 to 0.3718 and Hits@10 from 0.5786 to 0.5979.

Table 4: LogCL on ICEWS18. Hits use filtered ranking; hybrid combines entity and dyad states.

Variant	Raw MRR	Filtered MRR	Hits@1	Hits@3	Hits@10
Baseline	.3382±.0029	.3574±.0032	.2460±.0040	.4036±.0031	.5786±.0017
Wrong address	.3431±.0018	.3630±.0021	.2505±.0016	.4102±.0034	.5864±.0028
FA-State (dyad)	.3480±.0031	.3690±.0036	.2544±.0045	.4194±.0026	.5965±.0042
FA-State (hybrid)	**.3507±.0017**	**.3718±.0018**	**.2580±.0029**	**.4211±.0014**	**.5979±.0010**

5.5 History and task structure shape addressing gains

On ICEWS18, the addressing gain concentrates in relations with usable history (Figure 5). Both bins with at least four prior snapshots show positive mean effects in every seed, with the largest effect at 4–11 snapshots. Their query-weighted contributions recover the overall +0.00599 MRR gain after the offsets from sparse and unseen pairs.

Content value and address dependence. On `tgbl-wiki`, the single-seed occupancy-matched comparison improves MRR by 0.0259 over baseline and 0.0079 over occupancy alone. This supports content value beyond existence, while the smaller correct-minus-wrong effect remains uncertain

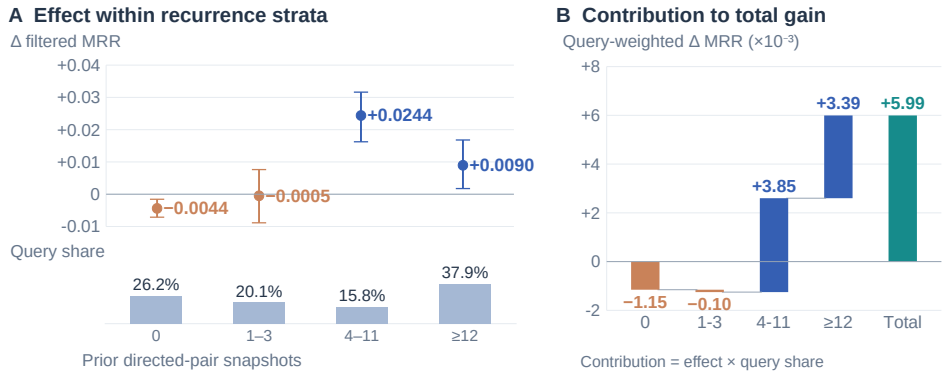


Figure 5: **Repeated relational history carries the gain.** Correct-minus-wrong filtered MRR on ICEWS18. A shows stratum effects and query shares; B decomposes the total gain into query-weighted contributions. Points average three seeds; error bars are paired 95% bootstrap intervals over seeds and directed pairs.

(its source-cluster interval crosses zero) and does not extend to cross-entropy. This task contrast distinguishes content value from address dependence; Appendix K gives the full results.

6 Related Work

Predictive sufficiency and structured memory. Predictive-state representations and causal states characterize history by its implications for future observations (Littman et al., 2001; Singh et al., 2004; Shalizi & Crutchfield, 2001); predictive rate–distortion formalizes compression that preserves future information (Marzen & Crutchfield, 2016). Structured memories organize state around grounded facts (Mei et al., 2020), addressable slots or semantic keys (Graves et al., 2014; Henaff et al., 2017), and persistent concept keys with evolving values (Zhang et al., 2017). Addressable virtual nodes decompose global memory in static message passing (Marcoccia, 2026).

Target-aware temporal graph representations. Temporal graph models encode interactions through point-process dynamics (Trivedi et al., 2019), coupled recurrent trajectories (Kumar et al., 2019), memory and temporal attention (Rossi et al., 2020; Xu et al., 2020), or MLP-based history summaries (Cong et al., 2023). Structural context comes from causal anonymous walks (Wang et al., 2021) and DyGFormer's co-occurrence encoding of paired historical neighborhoods (Yu et al., 2023). Target-aware designs include query-time pair structure in TNCN (Zhang et al., 2025), source–target identifiable TGN updates in TGNv2 (Tjandra et al., 2024), and candidate matching against recent source interactions in CRAFT (Yi et al., 2025).

Task-specific local and global history. For node-affinity prediction, NAVIS combines adaptive state-space updates, global virtual state, and a ranking objective (Feldman et al., 2026). In temporal knowledge graphs, DaeMon maintains query-aware path memory (Dong et al., 2023), and LogCL combines query-related local and global history encoders (Chen et al., 2024). FA-State studies predictive state scope as a common design axis through shared learning rules and address-specific states.

7 Conclusion

FA-State makes predictive state scope an explicit design choice, combining shared prediction with address-specific history through shared learning rules. Our analyses and experiments reveal the complementary roles of pooled evidence and local history across temporal backbones and tasks, while frozen retrieval interventions establish trained predictors' use of content–address correspondence. For directed dyads, gains depend on available history and task structure; learned and higher-order scopes remain future directions. Predictive state has a scope. Choosing that scope is part of learning what to remember.

Reproducibility statement

Appendices B–C.1 specify the update schedule, candidate construction, configurations, selection criteria, and paired uncertainty estimates. The experiment records provide per-seed metrics, validation-selected epochs, rank frequencies, and query-level evaluation scores; the controlled generator and complete training settings appear in Appendix B.1, with derivations in Appendix A.

AI use statement

Generative AI assisted with theoretical model development, the design of selected experiments, portions of code implementation, and language polishing of the manuscript. The authors take responsibility for the final manuscript and accompanying artifacts.

References

Wei Chen, Huaiyu Wan, Yuting Wu, Shuyuan Zhao, Jiayaqi Cheng, Yuxin Li, and Youfang Lin. Local–global history-aware contrastive learning for temporal knowledge graph reasoning. In *2024 IEEE 40th International Conference on Data Engineering*, pp. 733–746, 2024. doi: 10.1109/ICDE60146.2024.00062.

Jacob Chmura, Shenyang Huang, Ali Parviz, Farimah Poursafaei, Michael Bronstein, Guillaume Rabusseau, Matthias Fey, and Reihaneh Rabbany. TGM: A modular framework for machine learning on temporal graphs. In *ICML Workshop on Championing Open-source Development in Machine Learning (CODEML)*, 2025. URL https://openreview.net/pdf?id=7Bzm7GrP4d.

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.

Hao Dong, Zhiyuan Ning, Pengyang Wang, Ziyue Qiao, Pengfei Wang, Yuanchun Zhou, and Yanjie Fu. Adaptive path-memory network for temporal knowledge graph reasoning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 2086–2094, 2023. doi: 10.24963/ijcai.2023/232.

Or Feldman, Krishna Sri Ipsit Mantri, Moshe Eliasof, and Chaim Baskin. Revisiting node affinity prediction in temporal graphs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.06940.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://sites.stat.washington.edu/people/raftery/Research/PDF/Gneiting2007jasa.pdf.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rJTKKKqeg.

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*, 2023. URL https://arxiv.org/abs/2307.01026.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1269–1278, 2019. doi: 10.1145/3292500.3330895. URL https://snap.stanford.edu/jodie/.

Michael L. Littman, Richard S. Sutton, and Satinder P. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Félix Marcoccia. Designing a good virtual node: Addressable and cardinality-preserving global memory for message passing architectures. *arXiv preprint arXiv:2608.02709*, 2026. URL https://arxiv.org/abs/2608.02709. Preliminary work.

Sarah E. Marzen and James P. Crutchfield. Predictive rate–distortion for infinite-order markov processes. *Journal of Statistical Physics*, 163:1312–1338, 2016. doi: 10.1007/s10955-016-1535-3.

Hongyuan Mei, Guanghui Qin, Minjie Xu, and Jason Eisner. Neural datalog through time: Informed temporal modeling via logical specification. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 6808–6819, 2020. URL https://proceedings.mlr.press/v119/mei20a.html.

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

Cosma Rohilla Shalizi and James P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104:817–879, 2001. doi: 10.1023/A:1010388907793.

Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, pp. 512–519, 2004.

Temporal Graph Benchmark. Dynamic node property prediction, 2026. URL https://tgb.complexdatalab.com/docs/nodeprop/. Accessed September 5, 2026.

Benedict Aaron Tjandra, Federico Barbero, and Michael Bronstein. Enhancing the expressivity of temporal graph networks through source–target identification. *arXiv preprint arXiv:2411.03596*, 2024. URL https://arxiv.org/abs/2411.03596.

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://openreview.net/forum?id=KYPz4YsCPj.

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://arxiv.org/abs/2002.07962.

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*, 2025. URL https://arxiv.org/abs/2505.19408.

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. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d611019afba70d547bd595e8a4158f55-Abstract.html.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 765–774, 2017. doi: 10.1145/3038912.3052580. URL https://archives.iw3c2.org/www2017/proceedings/proceedings/p765.pdf.

Xiaohui Zhang, Yanbo Wang, Xiyuan Wang, and Muhan Zhang. Efficient neural common neighbor for temporal graph link prediction. *arXiv preprint arXiv:2406.07926*, 2025. URL https://arxiv.org/abs/2406.07926.

A PROOFS

A.1 ACCESSIBLE HISTORIES AND PREDICTIVE ALIASING

Under Definition 1, memory address μ receives $H_t^\mu = (e_i)_{i<t:\,\mu\in\sigma_{\mathrm{w}}(e_i)}$. The memory universe may include shared, endpoint, and relational addresses; implementations can instantiate only active addresses.

Definition 2 (Representation-induced equivalence)**.** *For a fixed query address a, histories h and h' are equivalent under ϕ when their accessible representations agree:*

$$h \sim_{\phi,a} h' \quad\Longleftrightarrow\quad R_\phi(h,a) = R_\phi(h',a). \tag{8}$$

For a coarse shared design, $R_\phi(h,a) = (\phi(h),a)$. The equivalence cells describe distinctions unavailable to any downstream readout.

Definition 3 (Predictive aliasing)**.** *The design R_ϕ aliases the future when $Y \not\perp H \mid R_\phi(H,A), A$. Equivalently, on a positive-probability set of accessible states and addresses, the conditional future law $p(Y \mid H, A)$ is not almost surely constant over the histories mapped to that state.*

A.2 ALIASING RISK (PROPOSITION 1)

Within an equivalence cell, the optimal coarse predictor is the mixture

$$p(y \mid C=c, A=a) = \int p(y \mid h,a)p(h \mid c,a)\,\mathrm{d}h. \tag{9}$$

Aliasing is therefore a property of the representation and query, rather than a synonym for small hidden width.

For log loss, the Bayes-optimal readout conditioned on a random variable X is $p(Y \mid X)$ and its risk is $H(Y \mid X)$. Since C is a deterministic function of H,

$$\begin{aligned}\mathcal{R}^*(C,A) - \mathcal{R}^*(H,A) &= H(Y \mid C,A) - H(Y \mid H,A)\\ &= I(Y;H \mid C,A).\end{aligned}$$

Expanding conditional mutual information gives the expected KL expression in Equation 2. It is zero exactly when $p(Y \mid H,A) = p(Y \mid C,A)$ almost surely, which is predictive homogeneity of the representation cells up to null sets. Equation 3 follows by the same argument after conditioning on (C,A) and adding Z_A.

A.3 SCOPE AND REPRESENTATION REFINEMENT

Definition 4 (Routing refinement)**.** *A routing scope $\mathcal{S}_f$ refines $\mathcal{S}_c$ when a quotient map $\kappa : \mathcal{M}_f \to \mathcal{M}_c$ consistently merges its write and read addresses: $\kappa[\sigma_{\mathrm{w}}^f(e)] = \sigma_{\mathrm{w}}^c(e)$ and $\kappa[\sigma_{\mathrm{r}}^f(a)] = \sigma_{\mathrm{r}}^c(a)$, where $\kappa[B] = \{\kappa(\mu) : \mu \in B\}$.*

Directed-dyad scope refines source scope through $\kappa(u,v) = u$.

Routing refinement describes how event subhistories are assigned to memory addresses. Representation refinement describes the information available after those subhistories have been compressed. A representation $\Psi = \psi(H)$ refines $C = \phi(H)$ when a measurable map γ satisfies $C = \gamma(\Psi)$. In particular, retaining a coarse route while adding fine-scope content gives $\Psi = (C, Z_f)$ and $C = \mathrm{proj}_C(\Psi)$.

Proposition 3 (Risk monotonicity under representation refinement)**.** *If $C = \gamma(\Psi)$ and the relevant conditional entropies are finite, then under population log loss*

$$\mathcal{R}^*(C,A) - \mathcal{R}^*(\Psi,A) = I(Y;\Psi \mid C,A) \geq 0. \tag{10}$$

For the retained shared-plus-scoped representation $\Psi = (C, Z_f)$, this becomes

$$\mathcal{R}^*(C,A) - \mathcal{R}^*(C,Z_f,A) = I(Y;Z_f \mid C,A) \geq 0. \tag{11}$$

Because $C = \gamma(\Psi)$ is deterministic, conditioning on (Ψ, C, A) is equivalent to conditioning on (Ψ, A). Hence

$$\begin{aligned}\mathcal{R}^*(C, A) - \mathcal{R}^*(\Psi, A) &= H(Y \mid C, A) - H(Y \mid \Psi, A) \\ &= H(Y \mid C, A) - H(Y \mid \Psi, C, A) \\ &= I(Y; \Psi \mid C, A) \geq 0.\end{aligned}$$

For $\Psi = (C, Z_f)$, the chain rule gives $I(Y; \Psi \mid C, A) = I(Y; Z_f \mid C, A)$. The population ordering follows from retaining C; shared-only and scoped-only representations remain unordered, and finite-sample estimation motivates their recurrence-dependent trade-off.

A.4 Binding-information decomposition

Proposition 4 (Binding-information decomposition)**.** *Let $B = (C, O, A)$ contain the base state, all occupancy or availability information exposed identically to both paths, and the query. For a fixed bank-generating representation and joint distribution, define correct retrieval Z_A and deranged retrieval $Z_{\pi(A)}$.*

$$\begin{aligned}&\mathcal{R}^*(B, Z_{\pi(A)}) - \mathcal{R}^*(B, Z_A) \\ &\quad = I(Y; Z_A \mid B) - I(Y; Z_{\pi(A)} \mid B).\end{aligned} \tag{12}$$

For either retrieval Z, the Bayes log-risk is

$$\mathcal{R}^*(B, Z) = H(Y \mid B, Z) = H(Y \mid B) - I(Y; Z \mid B).$$

Applying this identity to $Z_{\pi(A)}$ and Z_A and subtracting gives Equation 12. The derivation requires the base information B and bank-generating representation to be fixed. For a fully accessible candidate-state tuple, a reindexing that is invertible given the common conditioning information preserves its conditional information and has zero Bayes-risk gap. A fixed candidate-wise readout can still be sensitive to that reindexing. Thus the decomposition describes information available to an unrestricted predictor, while the architecture comparisons and frozen-model interventions assess how learned predictors use the content–address correspondence.

A.5 Gaussian shared–local frontier (Proposition 2)

For the shared predictor, the future residual is $\theta_a + \epsilon_{a,*}$, so independence gives $R_{\text{shared}} = \tau^2 + \sigma^2$. For $n_a > 0$, $\bar{X}_a = \mu + \theta_a + \bar{\epsilon}_a$, and therefore

$$\mathbb{E}\big[(X_{a,*} - \bar{X}_a)^2\big] = \mathbb{E}\big[(\epsilon_{a,*} - \bar{\epsilon}_a)^2\big] = \sigma^2 + \frac{\sigma^2}{n_a}.$$

The posterior mean is

$$\widehat{X}_{a,*} = \mu + w_a(\bar{X}_a - \mu), \qquad w_a = \frac{n_a\tau^2}{\sigma^2 + n_a\tau^2},$$

and the posterior variance of θ_a is $(\tau^{-2} + n_a\sigma^{-2})^{-1} = \tau^2\sigma^2/(\sigma^2 + n_a\tau^2)$. Adding the irreducible future noise σ^2 gives $R_{\text{shrink}} = \sigma^2 + \tau^2\sigma^2/(\sigma^2 + n_a\tau^2)$. Finally, $R_{\text{local}} < R_{\text{shared}}$ exactly when $\sigma^2/n_a < \tau^2$, or $n_a\tau^2 > \sigma^2$. For positive variances and $n_a > 0$, the posterior variance is smaller than both τ^2 and σ^2/n_a; at $n_a = 0$ it equals τ^2, recovering the shared predictor.

A.6 Dependence axis of the controlled process

Lemma 1 (Predictive separation in the controlled process)**.** *Fix a previously observed query address $a = (u, v)$ and condition on the same source bit $B = b$. Let $L = Y_a^{\text{last}}$ be the previous mark at that address, and let the selector and symmetric flip noise be independent. Under Equation 15, for $l, b \in \{0, 1\}$,*

$$\Pr(Y_t = 1 \mid L = l, B = b, A = a) = p_{\text{flip}} + (1 - 2p_{\text{flip}})\big(\delta l + (1 - \delta)b\big). \tag{13}$$

Consequently,

$$\begin{aligned}&\Pr(Y_t = 1 \mid L = 1, B = b, A = a) \\ &\quad - \Pr(Y_t = 1 \mid L = 0, B = b, A = a) = \delta(1 - 2p_{\text{flip}}).\end{aligned} \tag{14}$$

At the experimental flip rate $p_{\text{flip}} = .05$, the separation is 0.9δ.

Proof. Before the symmetric flip,

$$q_l := \Pr(\widetilde{Y}_t = 1 \mid L = l, B = b, A = a) = \delta l + (1 - \delta)b.$$

The flip channel gives

$$\begin{aligned}\Pr(Y_t = 1 \mid L = l, B = b, A = a) &= (1 - p_{\text{flip}})q_l + p_{\text{flip}}(1 - q_l) \\ &= p_{\text{flip}} + (1 - 2p_{\text{flip}})q_l.\end{aligned}$$

Since $q_1 - q_0 = \delta$, subtraction yields Equation 14. □

B Implementation Details

Temporal execution. The node-affinity implementations process the benchmark's chronological merged edge–label stream in minibatches. A batch's label queries are scored from the already committed prefix before that batch's edges update the backbone, factor states, or candidate mask. The benchmark labels describe future behavior after their query cutoff (Huang et al., 2023; Temporal Graph Benchmark, 2026); events tied at the cutoff can already belong to the committed prefix. The implemented observation protocol is therefore batch-prefix forecasting. Within each comparison, all variants receive the same batches and visible history. ICEWS18 instead uses complete timestamp snapshots, with prediction preceding the current snapshot's update.

Algorithm 1 Online factor state with score-before-commit minibatches

Require: Chronological batches $(\mathcal{E}_b, \mathcal{Q}_b)$, backbone state C, materialized factor bank $\mathcal{Z}$, pending messages P

for batch b in chronological order **do**
  for all queries $Q \in \mathcal{Q}_b$ and candidate outcomes y **do**
    $a \leftarrow a(Q, y)$ ▷ or $\pi_b(a(Q, y))$ for wrong retrieval
    $z \leftarrow \textsc{ReadPending}(\mathcal{Z}, P, a)$ ▷ zero if unseen
    compute $\ell(y \mid H, Q)$ from C and z using Equation 7
  form the task loss from the scored queries, when training
  commit $\mathcal{E}_b$ to the backbone and causal candidate history
  for all addresses a affected by $\mathcal{E}_b$ **do**
    materialize any earlier pending update at a, with stopped gradients
    activate a if unseen; set $P[a] \leftarrow \textsc{LastMessage}(\mathcal{E}_b, a)$
  when training, backpropagate the query loss and update shared parameters
  detach materialized states and pending raw messages between batches

Messages, delayed updates, and gradients. The online encoder linearly projects the raw edge attributes, adds a learned source, destination, or directed-pair role embedding, and applies layer normalization and GELU. A GRUCell maps the resulting message and previous state to the updated state. Repeated occurrences of one address in a minibatch use the last raw message in stream order. Committed messages remain pending until a subsequent read. That read evaluates the GRU update with gradients without changing the stored bank; materializing an older update before its next write stops gradients through the accumulated past. Training thus uses truncated temporal gradients, following the delayed-memory pattern of TGN (Rossi et al., 2020). States are reset for each training epoch. Before validation, parameter-dependent memories are reconstructed by replaying the training prefix with the current parameters held fixed. After model selection, training and validation prefixes are replayed before test scoring.

Candidate-wise readout. For node-affinity prediction, the addressed query is $q_\eta^0 = \mathrm{normalize}(\mathrm{MLP}_\eta(0, 0))$ with zero query dropout. It is identical for all sources and candidates; candidate dependence enters through $z_{(u,y)}$. The backbone supplies the shared logits. The optional online entity route averages the source-role state of u and destination-role state of y. Learned gains are softplus transforms of scalars initialized to -2.

Online entity states. The generic online implementation stores source- and destination-role entity states in a $2|V|$-slot bank, separate from the directed-pair bank. Both banks share the message encoder and GRUCell parameters. A committed event (u, v, m) supplies its raw message to the source-role slot u, destination-role slot $|V| + v$, and directed-pair slot (u, v) under the delayed-update protocol.

Temporal-KG realization. For each entity or directed pair, the raw-history store retains the most recent twelve active timestamp bags of relation tokens. It includes inverse relation tokens for the reversed direction. Relation embeddings of width 32 are mean-pooled within a timestamp bag, and a GRU of width 64 encodes the ordered bags. This encoder is recomputed from the stored prefix during joint training. Its state is scored against the normalized projection of LogCL's decoder query; candidate y retrieves the history of (u, y). The hybrid variant additionally reads the candidate entity's relation history using the same encoder. The full-entity logits receive these additive scores before softmax. Static-graph encoding is retained, and the combined entity scores are trained jointly with LogCL's contrastive objective.

Prefix-conditional wrong-address mapping. Let S_u^b be the sorted partner set of source u in the committed prefix of batch b. For $|S_u^b| > 1$, the wrong-address rule applies a nonzero cyclic offset to this list. The offset is $1 + ((1000003s + 9176u) \bmod (|S_u^b| - 1))$ for permutation seed s. The trade and genre controls use $s = 1901$; the LogCL controls use $s = 1700 + k$ for training seed k. Each seen partner maps to a different seen partner of the same source. Singleton sets retain their original lookup, and unseen addresses read as zero. The generating rule is fixed and outcome-independent; the resulting map $\pi_{u,b}$ can change when the prefix partner set grows. The same within-source rule is applied to the LogCL history store at each snapshot. In these real-stream comparisons, correct and wrong models share update rules, occupied addresses, architecture, and parameter counts, but are trained separately.

Occupancy and identity controls. The occupancy route replaces an observed address state by a fixed all-ones marker and an unseen state by zero. With the shared query q_η^0, this adds the same scalar to every occupancy-matched candidate and cancels from the conditional softmax. Source–target identity controls instead supply identity information through the backbone's event-message function in every variant. Table 5 summarizes shared backoff and the signals supplied by the additional factor route. The scoped-only variant is used in the controlled process; real-data factor models retain their backbone prediction route.

Table 5: Shared backoff and signals supplied by the added factor route. Real-stream correct and wrong variants match architecture and parameter count, not trained parameter values. A dash does not imply that the backbone lacks the information.

Variant	Shared	Occupancy	Content	Correct binding
Baseline/shared	✓	–	–	–
Occupancy only	✓	✓	–	–
Wrong address	✓	✓	✓	–
Correct address	✓	✓	✓	✓
Scoped only	–	✓	✓	✓

Controlled-process predictors. The binary-mark predictors share a recurrent history encoder and combine endpoint and directed-pair sequences. Appendix B.1 specifies their active readouts, generator, and complete training settings.

State representation. Node-indexed shared state occupies $O(|V|d_s)$ values; a sparse factor bank stores $O(|\mathcal{A}_{\text{active}}|d_a)$ values. The generic online implementation uses a dynamic pair-to-slot map. For the bounded genre task, we preallocate $2^{19} = 524{,}288$ slots for the $992 \times 513 = 508{,}896$ possible user–genre addresses and use a causal activity mask; unseen addresses still read as zero. This dense specialization preserves the learned modules, delayed updates, and prefix-conditional mapping. With message width d_m, a dense GRUCell update costs $O(d_a^2 + d_a d_m)$, and the inner-product readout costs $O(d_a)$ per candidate.

B.1 Controlled-process configuration

Episodes and observations. Each episode starts with independently sampled $b_u \sim \text{Bernoulli}(1/2)$, empty histories, and a fresh fixed-point-free permutation of the directed non-self pair space. The recurrence branch samples uniformly from previously observed pairs with probability ρ; otherwise, a

pair is sampled uniformly from all directed non-self pairs. The all-pair rule is also used before the first observation. Because this branch can itself select a previously observed pair, ρ is a sampling-mixture parameter rather than the realized repeat fraction.

At step t, let $a_t = (u, v)$ be the sampled directed pair and $y_{a_t}^{\text{last}}$ its last observed mark, when available. The next mark uses that pair history or the current source bit, followed by independent symmetric flip noise:

$$\begin{aligned} \widetilde{y}_t &= \begin{cases} y_{a_t}^{\text{last}}, & a_t \text{ seen and } D_t = 1, \\ b_u, & \text{otherwise}, \end{cases} & D_t &\sim \text{Bernoulli}(\delta), \\ y_t &= \widetilde{y}_t \oplus \epsilon_t, & \epsilon_t &\sim \text{Bernoulli}(p_{\text{flip}}), \quad p_{\text{flip}} = .05. \end{aligned} \tag{15}$$

Prediction uses only the preceding events. After observing the mark y_t for (u, v), the process sets $b_u \leftarrow y_t$ and $b_v \leftarrow y_t$, appends $(y_t, v, 0)$ to the source history and $(y_t, u, 1)$ to the destination history, and appends $(y_t, v, 0)$ to the directed-pair history. A token records the observed mark, partner identity, and endpoint role. Histories retain their twelve most recent tokens; generator bits and axis parameters are not model inputs.

Table 6: Controlled-process data and training settings. The width sweep jointly varies the GRU state, projection, and gate hidden widths; token and query-endpoint embeddings remain 16-dimensional.

Setting	Value
Entities / directed non-self pairs	$12/132$
Events per episode / history length	$128/12$
Training / validation / test episodes	$24/8/8$
Training / validation / test examples	$3{,}072/1{,}024/1{,}024$
Symmetric mark-flip probability	.05
Dependence grid δ	$\{0, .2, .4, .6, .8, 1\}$
Recurrence grid ρ	$\{.1, .3, .5, .7, .9\}$
Optimization and data seeds	$1, 2, 3$
Token embedding / GRU hidden width	$16/24$
Projection dropout	.1
Optimizer / learning rate	AdamW / .002
Weight decay / batch size	$.0001/256$
Gradient clipping	Global norm 1
Epoch maximum / stopping patience	$6/2$
Checkpoint selection	Minimum validation binary NLL
Width-sweep coordinates	$\delta = .8,\ \rho = .7$
Width-sweep hidden dimensions	$\{8, 16, 24, 48\}$

Encoder and predictors. Learned mark, partner, and endpoint-role embeddings are summed before a GRU shared by endpoint and pair histories. Its final valid output represents each sequence; an empty history encodes to zero. The branch projections concatenate these encodings with learned query-endpoint embeddings and apply a linear map, GELU, and dropout before a scalar head. Write the shared and scoped logits as s and r, and their hidden representations as h_s and h_r. Shared-only and scoped-only predictors output s and r, respectively. In the four-way comparison, the correct-scoped predictor uses $s + \text{sigmoid}(\beta)r$ with one learned scalar β, while the wrong-scoped predictor uses $s + \text{sigmoid}(G([h_s, h_r]))r$ with a two-layer GELU MLP G. All modes instantiate the same modules, including inactive branches, while their active readouts differ.

Training and splits. Training minimizes binary cross-entropy with shuffled minibatches. Validation is evaluated after every epoch; an NLL improvement must exceed 10^{-6} to reset stopping patience. The selected checkpoint is evaluated on independently generated test episodes. Within each grid cell and seed, all predictors share the same three datasets. The width sweep likewise shares datasets across widths and predictors, using episodes generated independently of the grid. For zero-based grid indices i, j and seed k, the generator seed is $10^6 s + 10^4 i + 100j + k$, where $s = 1, 2, 3$ denotes training, validation, and test. The corresponding width-sweep seeds are $4{\times}10^6 + k$, $5{\times}10^6 + k$, and $6{\times}10^6 + k$.

B.2 Address comparison with scalar gates

Both predictors use $s + \mathrm{sigmoid}(\beta)r$, with one learned scalar β. The wrong-address predictor reads the episode-fixed deranged pair history in place of the queried pair's history. Query endpoints, entity histories, labels, and the generator remain unchanged; an unobserved mapped pair has empty history. Each pair starts from identical parameters and uses the same generated splits, minibatch-order seed, optimizer, and validation selection rule, then trains independently. The adaptive-gate MLP remains instantiated but receives no loss gradient in either predictor.

The comparison covers all 30 grid cells and the four-width slice in Table 6, each with three seeds: 102 pairs and 204 training runs. The width data remain independent of the grid data. The four-way comparisons in Figure 2 and Table 1 use CPU runs. Here both members of each pair are trained on the same GPU (A40 for seed 1; A100 for seeds 2 and 3), using PyTorch 2.1.2 and CUDA 12.1, with deterministic algorithms enabled and TF32 disabled.

Table 7 reports every cell. Correct addressing improves NLL in all 75 seed–cell comparisons with $\delta > 0$. At $\delta = 0$, the mean gap is 0.00055; the positive-seed counts for increasing ρ are 3/3, 2/3, 2/3, 2/3, and 1/3. Table 8 gives the width slice, where all 12 seed–width comparisons favor correct addressing.

Table 7: Scalar-gate address comparison over the complete grid. Entries are wrong-minus-correct test NLL, mean ± sample standard deviation across three paired training/data seeds. Positive values favor correct addressing.

	Recurrence mixture ρ				
Dependence δ	.1	.3	.5	.7	.9
0.0	$+.0011 \pm .0010$	$+.0003 \pm .0007$	$+.0014 \pm .0020$	$-.0002 \pm .0011$	$+.0001 \pm .0021$
0.2	$+.0062 \pm .0035$	$+.0118 \pm .0027$	$+.0119 \pm .0043$	$+.0164 \pm .0018$	$+.0071 \pm .0010$
0.4	$+.0267 \pm .0086$	$+.0391 \pm .0137$	$+.0293 \pm .0075$	$+.0270 \pm .0093$	$+.0133 \pm .0079$
0.6	$+.0734 \pm .0079$	$+.0707 \pm .0247$	$+.0721 \pm .0182$	$+.0632 \pm .0154$	$+.0189 \pm .0053$
0.8	$+.1213 \pm .0249$	$+.1164 \pm .0068$	$+.0992 \pm .0075$	$+.0965 \pm .0155$	$+.0477 \pm .0135$
1.0	$+.1688 \pm .0231$	$+.1764 \pm .0151$	$+.1520 \pm .0342$	$+.1317 \pm .0145$	$+.0799 \pm .0269$

Table 8: Scalar-gate address comparison across hidden widths at $\delta = .8$, $\rho = .7$. Values are mean ± sample standard deviation over three paired seeds; the last column summarizes paired NLL differences.

Width	Correct NLL	Wrong NLL	Wrong − correct
8	$.3658 \pm .0068$	$.4245 \pm .0082$	$+.0587 \pm .0014$
16	$.3277 \pm .0164$	$.4061 \pm .0137$	$+.0783 \pm .0076$
24	$.3208 \pm .0200$	$.3975 \pm .0069$	$+.0767 \pm .0137$
48	$.3169 \pm .0240$	$.3974 \pm .0072$	$+.0805 \pm .0175$

C Experimental Protocol

Tasks and observations. We use the chronological splits supplied by TGB for trade and genre forecasting (Huang et al., 2023). Trade predicts the next year's destination distribution over 255 nations; genre predicts a user's genre affinities over the next week, with 513 destination classes and 992 user nodes. Models receive raw edge attributes and the committed history specified in Appendix B. ICEWS18 uses its chronological temporal-KG split and both forward and inverse entity-prediction queries. Its history encoder receives relation tokens from earlier snapshots.

Occupancy-matched targets. For source u, let S_u^b contain destinations observed in the committed prefix before batch b. For the benchmark future target $p_t(y \mid u)$, define its retained mass and

conditional target by

$$m_{u,t} = \sum_{y \in S_u^b} p_t(y \mid u), \qquad \widetilde{p}_t(y \mid u) = \frac{\mathbf{1}[y \in S_u^b] p_t(y \mid u)}{m_{u,t}}. \tag{16}$$

A row is eligible when $|S_u^b| \geq 2$ and $m_{u,t} > 10^{-12}$. All variants in a node-affinity comparison minimize the same conditional cross-entropy,

$$\mathcal{L}_{\text{matched}} = -\frac{1}{|\mathcal{Q}_{\text{eligible}}|} \sum_{(u,t) \in \mathcal{Q}_{\text{eligible}}} \sum_{y \in S_u^b} \widetilde{p}_t(y \mid u) \log \frac{\exp \ell_t(u, y)}{\sum_{v \in S_u^b} \exp \ell_t(u, v)}. \tag{17}$$

Matched NDCG@10 uses the same eligible rows and candidate sets, with linear relevance gains. Full NDCG@10 uses all benchmark classes and the unconditioned target. Both metrics are averaged over label rows. The full-candidate score of a matched-trained model therefore measures transfer of that model to the unrestricted risk set. Original full-objective experiments are identified separately from this common-objective comparison.

Table 9: Coverage of the occupancy-matched node-affinity protocol. Candidate count and retained target mass are means over eligible rows.

Dataset	Split	Eligible / all rows	Candidates	Retained mass
Trade	Validation	850/850	142.6741	1.000000
Trade	Test	664/665	143.6913	1.000000
Genre	Validation	36,513/36,620	158.7962	.979328
Genre	Test	32,074/32,151	163.2486	.982224

Optimization and model selection. We use optimization seeds 1, 2, and 3 for the main real-data comparisons. Table 10 gives the node-affinity configurations. All four use Adam, batch size 200, factor state and message widths 64, zero factor query dropout, and gradient clipping at norm 1. TGN and TGAT use backbone dropout .1. The TGAT encoder has two attention heads per layer. Node-affinity checkpoints maximize matched validation NDCG@10 after fixed-parameter prefix replay. All variants within a configuration share the optimizer, candidate construction, validation schedule, and stopping rule. Genre uses eight training epochs and validation at epochs $1, 2, 4, 6, 8$; its patience counts three scheduled validation checks. The TGNv2 comparison completes 750 epochs and selects the best matched-validation checkpoint. Its StepLR scheduler halves the learning rate every 250 epochs. LogCL selects checkpoints by validation filtered MRR and uses entity prediction with the contrastive objective. Test evaluation follows checkpoint selection.

Table 10: Node-affinity training configurations. Dimensions are memory/embedding/time, with the additional identity width shown for TGNv2. TGAT has no recurrent node-memory dimension. Patience counts validation checks; validation runs every epoch except for the stated genre schedule.

Setting	Dimensions	Neighbors	Adam LR	Epoch cap / patience
TGN, trade	100/100/100	10	10^{-4}	60/15
TGAT, trade	$--/172/100$	20×2 layers	10^{-4}	60/15
TGNv2, trade	784/784/784; ID 784	25	10^{-3}	750/750
TGN, genre	100/100/100	10	10^{-4}	8/3

The LogCL configuration uses Adam with learning rate .001, two graph layers of width 200, the UVRGCN encoder and ConvTransE decoder, and ten-snapshot backbone histories during both training and evaluation. Static-graph encoding and contrastive learning are enabled, with contrastive temperature .03. Its factor encoder uses relation-token width 32, state width 64, and twelve active timestamp bags. Training has a 40-epoch maximum and patience 10, with validation after every epoch.

C.1 Paired uncertainty estimates

Performance tables report the mean and sample standard deviation across optimization seeds. For paired node-affinity effects, we first align label rows by source and timestamp and compute the within-row difference between variants. Each of 5,000 bootstrap replicates samples source clusters with replacement, using the same sampled sources for every training seed and both variants. Within a seed, the sampled source sums are divided by their sampled row counts, retaining the event-weighted target estimand. We then resample the training seeds with replacement and average their effects equally. The 2.5th and 97.5th percentiles give the reported paired interval. Source cluster multiplicities remain shared across training seeds throughout each replicate.

Genre comparisons contain 32,074 aligned eligible rows per seed, grouped into 567 source clusters. Their bootstrap random seeds are 191901 for correct minus baseline and 191902 for correct minus wrong. The single-seed Wiki comparison in Appendix K instead uses 1,000 paired resamples of source clusters (random seed 8204), with a circular temporal-block sensitivity analysis (block length 200, random seed 8202).

For ICEWS18 recurrence strata, the analogous paired procedure uses directed dyads as the primary clusters and reciprocal-rank differences as its outcome. The query set includes both prediction directions; a displayed fact count corresponds to two directed prediction queries. Recurrence counts previous snapshots containing the directed pair, and strata are constructed from the prefix independently of model scores. Source, unordered-dyad, and timestamp clusterings provide additional sensitivity views. Temporal slices describe where the effect occurs; they are not additional independent training seeds.

C.2 Query-level displays and probability examples

Figure 4 and the complete displays below reuse the existing matched test predictions. Rows are paired by source and timestamp, with identical candidate counts and target mass. In the main figure, each point averages a query's NDCG across seeds 1–3; it does not evaluate an ensemble prediction. Figures 6 and 7 show each seed separately. The 664 queries repeated across three models are not treated as independent replications. A difference within 10^{-10} of zero is counted as a numerical tie. Win fractions, medians, and empirical cumulative distributions are descriptive summaries, not additional hypothesis tests.

For the separately trained TGNv2 access rules, correct addressing wins on 628, 625, and 640 queries (94.6%, 94.1%, and 96.4%), with median NDCG improvements of 0.1834, 0.1645, and 0.1906. In the compact frozen TGN, correct reads improve NDCG on 74.8%, 72.0%, and 77.7% of queries.

The probability examples use seed 1 of the frozen intervention and its primary map, 1901. Among queries in the latest test year, 2015, we compute target entropy $-\sum_y \widetilde{p}(y) \log \widetilde{p}(y)$ and select the rows nearest its 25th, 50th, and 75th percentiles, resolving ties by source ID. The selected sources are 59, 42, and 86. Model predictions and gains do not enter row selection. Saved masked logits are converted to probabilities using their original conditional softmax; recomputed cross-entropies match the stored values. Each display uses the union of the top four candidates under the target, correct prediction, and wrong prediction, ordered by target mass with ties by candidate ID. The remaining mass is shown as "Rest" without renormalizing the displayed subset. The common linear color scale spans 0–55%; displayed NDCG always uses the full matched candidate set.

For Figure 5B, if n_b is the number of queries in stratum b and $\overline{\Delta}_b$ its equal-seed-mean reciprocal-rank difference, the displayed contribution is $(n_b / \sum_j n_j)\overline{\Delta}_b$. The four contributions sum to the overall equal-seed-mean difference. The counts include both prediction directions. This decomposition leaves the original strata and estimand unchanged.

D Frozen Retrieval Intervention

Source model and intervention. We use three validation-selected, correctly addressed models from the compact source–target-identified TGN configuration. Its memory, embedding, time, and identity dimensions are 100, with factor and message dimensions 64. It uses batch size 200, ten

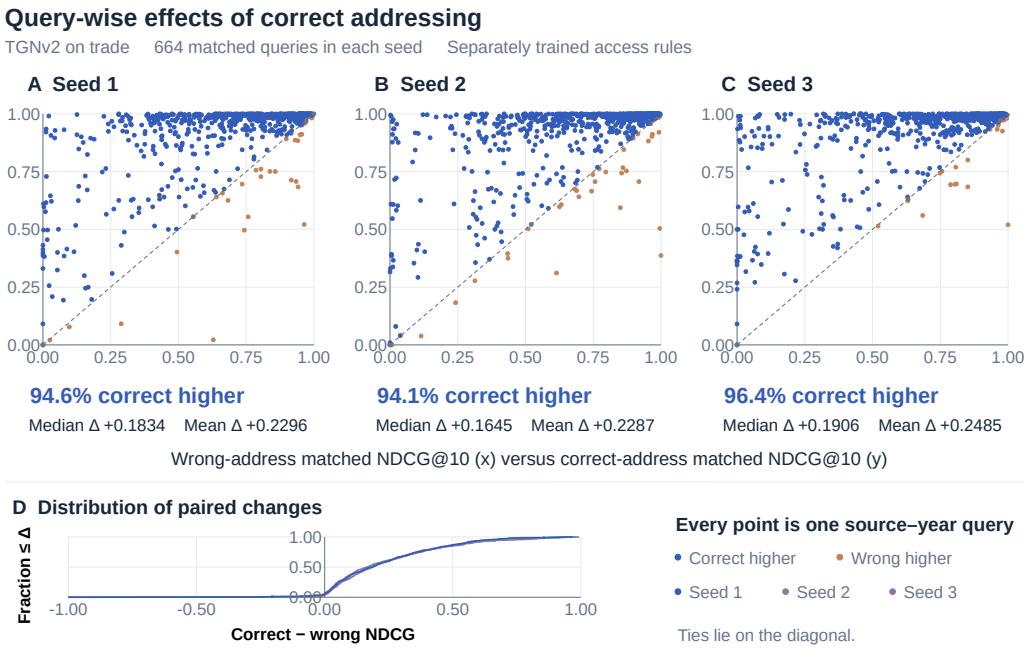


Figure 6: **Complete query-level results for the identity-aware addressing comparison.** A–C pair the separately trained TGNv2 correct- and wrong-address models on the same 664 matched test queries per seed. Points above the diagonal favor correct addressing; orange points favor wrong addressing and gray points are ties. D shows the empirical cumulative distribution of paired NDCG differences.

neighbors, learning rate 10^{-4}, a 60-epoch maximum, and patience 15. This compact source model is distinct from the 784-dimensional TGNv2 comparison in Section 5.2.

At evaluation, learned weights are held fixed and the prefix is replayed to construct one shared online state bank. Each batch is scored once with the correct lookup and again with each wrong-address lookup, using the same backbone logits, candidate set, targets, and state bank. Reads do not mutate the bank; the observed batch is committed once after all lookup variants have been scored. This isolates sensitivity to retrieval while preserving the update trajectory. The primary permutation seed is 1901, with four additional seeds testing mapping sensitivity. All eligible candidate lookups are deranged within their source.

Table 11: Frozen-model retrieval sensitivity on trade. Values are correct minus wrong matched NDCG@10 for three training seeds. Only the inference-time lookup changes across rows.

Map seed	Training seed 1	Training seed 2	Training seed 3	Mean
1901	.04747	.05227	.05606	.05193
2718	.04801	.05270	.05486	.05186
3141	.04789	.05124	.05753	.05222
7919	.04669	.05100	.05481	.05083
104729	.04881	.05281	.05580	.05247

For the primary map, correct retrieval improves matched NDCG@10 by .05193 with paired 95% interval $[.04522, .05894]$. Wrong-minus-correct conditional cross-entropy is .11510 with interval $[.10504, .12520]$. These paired changes measure retrieval sensitivity in a trained model and complement the separately trained architecture controls.

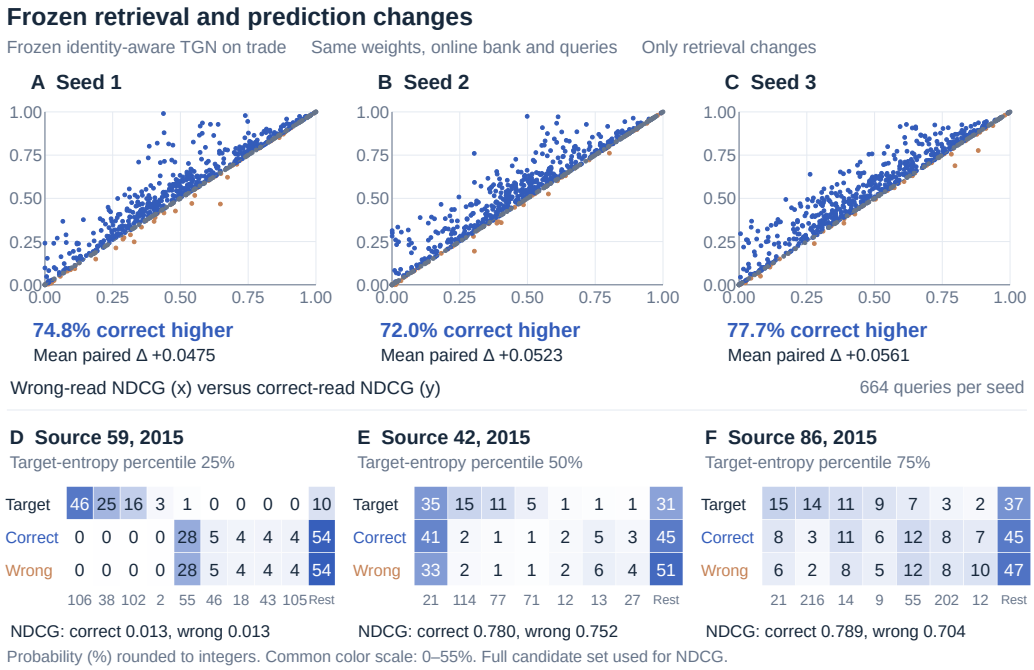


Figure 7: **Complete frozen-retrieval results and probability examples.** Each paired comparison uses the same weights, online bank, backbone logits, and test queries; only the read address changes. A–C show all 664 matched queries per seed. D–F show conditional probability distributions for the three seed-1 queries selected by target entropy. "Rest" pools candidates outside the displayed union of top-four sets. Colors share a linear scale; cells show rounded percentages, and NDCG uses all candidates. Appendix D specifies the compact identity-aware TGN configuration.

E Cross-Setting Results by Seed

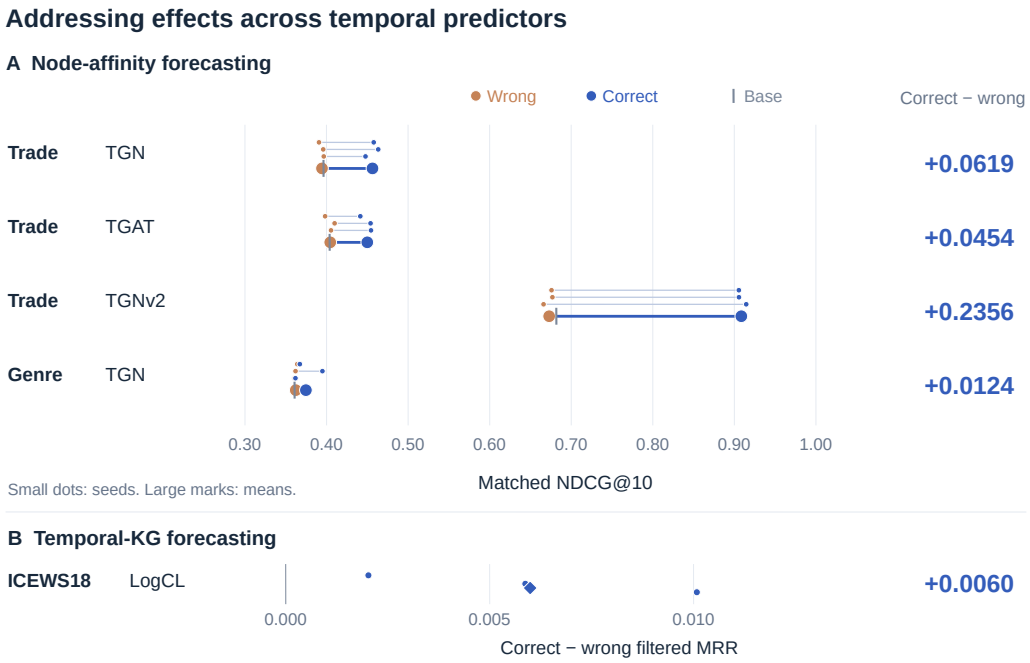


Figure 8: Absolute performance and correct-minus-wrong gains across settings. The seed-level view complements the mean effects in Figure 3. Node-affinity forecasting uses matched NDCG@10; temporal-KG forecasting uses filtered MRR. Each access rule is trained separately.

F Source–Target Identity Control

Table 12 reports every TGNv2 control run. The independently implemented architecture uses trainable cosine node identification, directional source/target message order, cosine time encoding, last aggregation, GRU memory, a two-head one-layer temporal graph attention encoder, and a two-layer decoder. We use the paper-reported trade configuration rather than the public command-line defaults. Correct and wrong FA-State variants have the same factor architecture and parameter count. The wrong-address rule is fixed independently of outcomes, as detailed in Appendix B.

Table 12: Matched NDCG@10 for every TGNv2 control run.

Seed	TGNv2	+ wrong-address control	+ FA-State (dyad)
1	.6783	.6759	.9056
2	.6869	.6770	.9057
3	.6801	.6661	.9146
Mean	.6818	.6730	.9086

The paired correct-minus-wrong effects are $+.2296$, $+.2287$, and $+.2485$ for seeds 1–3. The paired source-cluster and seed bootstrap described in Appendix C.1 gives the interval $[0.2101, 0.2631]$. Correct-minus-TGNv2 is $+.2269$ with interval $[0.2018, 0.2532]$. Both comparisons are positive in all nine seed–test-year cells. All variants complete the same frozen training horizon, use matched validation NDCG for checkpoint selection, and evaluate test labels once after selection.

G GENRE FORECASTING PER-SEED RESULTS

Table 13 reports all nine `tgbn-genre` runs. Candidate identities, target masses, and event rows are matched within each seed; correct and wrong variants use the same factor architecture. Checkpoints are selected by matched validation NDCG before test evaluation.

Table 13: Matched NDCG@10 for every `tgbn-genre` run.

Seed	Baseline	Wrong-address control	FA-State (dyad)
1	.3607	.3642	.3673
2	.3607	.3620	.3950
3	.3611	.3608	.3619
Mean	.3608	.3623	.3748

The paired correct-minus-baseline effects are $+.0066$, $+.0343$, and $+.0009$; correct-minus-wrong effects are $+.0032$, $+.0330$, and $+.0011$. All six effects are positive. Hierarchical seed/source resampling with 5,000 draws yields $[0.0010, 0.0336]$ and $[0.0011, 0.0322]$, respectively. The full-candidate correct-minus-baseline effects are $+.0060$, $+.0275$, and $-.0061$, giving the heterogeneous full-candidate response discussed in Section 5.4.

H RESULTS STRATIFIED BY RECURRENCE

Table 14 expands the ICEWS18 recurrence view from Section 5.5. Intervals use a paired hierarchical bootstrap over optimization seeds and directed-dyad clusters. Repeating the clustering by source, unordered dyad, or timestamp gives the same qualitative separation. The stratification is a descriptive analysis of where the aggregate effect is concentrated.

Weighting each stratum's effect by its query share yields contributions of -0.00115, -0.00010, $+0.00385$, and $+0.00339$ for the four bins, respectively. Their sum is the overall $+0.00599$ filtered-MRR gain.

Table 14: Correct-minus-wrong addressing by the number of prior directed-pair snapshots on ICEWS18. MRR is averaged equally over three seeds.

Prior snapshots	Facts	Δ filtered MRR	Positive seeds	95% interval
0	12,988	$-.0044$	0/3	$[-.0071, -.0016]$
1–3	9,965	$-.0005$	2/3	$[-.0089, +.0076]$
4–11	7,825	$+.0244$	3/3	$[+.0162, +.0316]$
≥ 12	18,767	$+.0090$	3/3	$[+.0017, +.0168]$

I ICEWS18 PER-SEED RESULTS

Table 15: Filtered MRR for every LogCL/ICEWS18 run.

Seed	Baseline	FA-State (dyad)	FA-State (hybrid)	Wrong-address control
1	.3541	.3671	.3738	.3651
2	.3574	.3667	.3702	.3608
3	.3606	.3731	.3715	.3630
Mean	.3574	.3690	.3718	.3630

Correct dyad minus wrong address is $+.0020$, $+.0059$, and $+.0101$ for seeds 1–3. The direction is consistent across seeds. The frozen-model intervention in Section 5.3 provides complementary evidence that holds the learned parameters fixed.

J Capacity Sweep

We vary the predictive-state hidden width at relational dependence $\delta = .8$ and recurrence $\rho = .7$. All four variants use the same width at each point and the readouts in Appendix B. The shared-plus-correct predictor retains its advantage throughout the tested range, while increasing width also improves the shared route. Table 8 reports the same width slice with a scalar gate in both the correct- and wrong-address readouts.

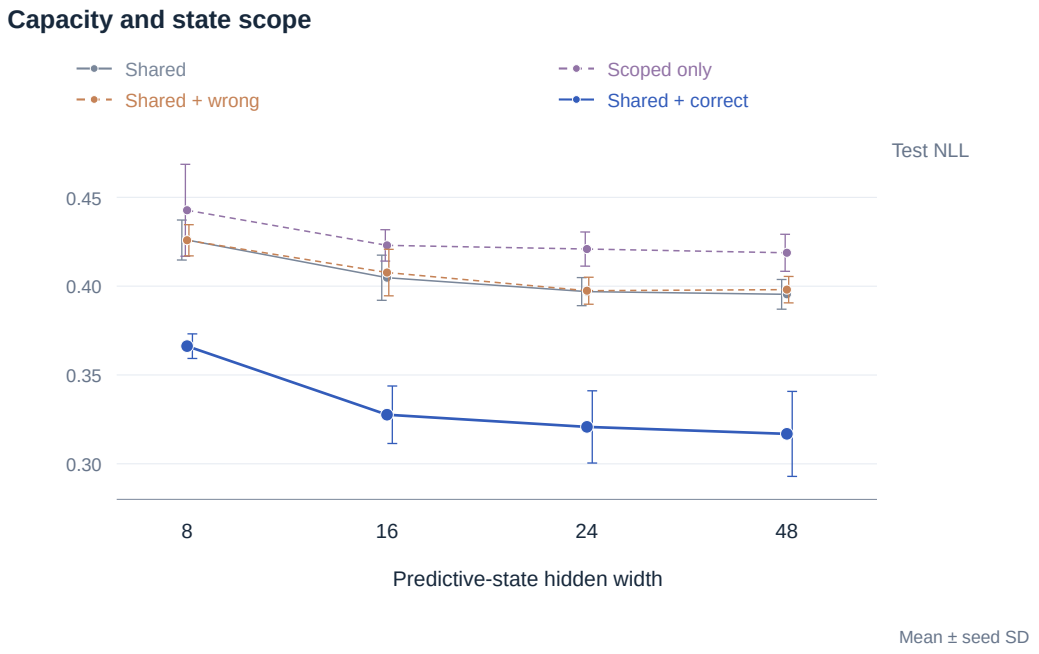


Figure 9: Capacity sweep in the controlled process. Points show mean test NLL, with sample standard deviation over three seeds. The shared-plus-correct advantage persists across the tested hidden widths. Correct and wrong variants retain their global-scalar and input-dependent gates.

K Link-Prediction Boundary Case

The `tgbl-wiki` experiment trains TGN variants with occupancy-matched candidate prediction. Each eligible test event ranks its positive destination against $K = 5$ causally observed negatives from the official TGB candidate set, giving six candidates in total. The full protocol uses the complete benchmark candidate set. These one-seed results complement the multi-seed node-affinity comparisons.

Table 16: Occupancy-matched link prediction on `tgbl-wiki`. Matched MRR and candidate cross-entropy use the same 5,582 eligible test events, each with one positive and five matched negatives; full MRR uses the benchmark protocol. One optimization seed.

Variant	Matched MRR ↑	Candidate CE ↓	Full MRR ↑
Baseline	.7077	1.2556	.0738
Occupancy only	.7257	1.2189	**.0947**
Wrong address	.7278	**1.1933**	.0871
Correct address	**.7336**	1.1947	.0896

Correct addressing improves matched MRR over occupancy and wrong addressing. For correct minus wrong, the paired event-mean effect is $+0.0058$, with a 95% source-cluster bootstrap interval of

$[-0.0019, 0.0137]$ (1,000 draws). A circular moving-block bootstrap of the chronological events, with block length 200 and 1,000 draws, gives $[-0.0011, 0.0138]$. These intervals resample the 5,582 paired test events from one trained seed; they do not measure optimization-seed variability. The effect is positive in six of ten chronological deciles, while wrong addressing has slightly lower cross-entropy. This distinguishes a ranking gain from stable address dependence and complements the stronger separation on node-affinity prediction.

L Limitations and Future Directions

We study directed dyadic addresses; typed factors, higher-order relations, and learned address discovery remain open. Address-indexed storage also introduces a memory–coverage tradeoff, especially for streams with little recurrence. The main real-stream comparisons use three optimization seeds on three public datasets; the supplementary link-prediction boundary case uses one seed. On `tgbn-genre`, correct addressing improves the matched metric in all three seeds, but the effect size varies substantially and the full-candidate gain is positive in only two. Recurrence strata are descriptive, not randomized interventions. Finally, the theory isolates information lost by a given representation and a stylized estimation tradeoff; the experiments assess particular learned architectures rather than a universal separation between shared and addressed memory.