TEMPORAL SHEAF DIFFUSION: MEMORY UNDER CHANGING GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

When relational geometry changes, should temporal memory preserve past comparisons or reinterpret history using current ones? For a prescribed evolving sheaf, we characterize the exact finite-polynomial historical projection through signal moments and a positive-definite Gram state, deriving continuous dynamics, exact held-history updates, and a compressed representation using $2N - 1$ operator moments for N temporal modes. A three-node graph example separates historical regularization, current-geometry filtering, and compression of filtered inputs, showing why time-varying pointwise filtering can leave the polynomial space. Synthetic reconstruction shows that the appropriate geometry depends on the meaning of the observed changes, with little benefit from the coupled solve over filter-then-compress, while a compact Temporal Sheaf Diffusion predictor is competitive on temporal knowledge graphs.

1 INTRODUCTION

Temporal graph learning combines changing node values with changing relations. A memory that respects the comparisons made when observations arrived need not equal one that applies today's comparisons to the entire past. We ask what state is needed to retain historical geometry during temporal compression, and how this differs from the memory used by a practical predictor.

Cellular sheaves express relational geometry through endpoint maps into shared edge spaces. Their Laplacians measure transformed disagreement, allowing connected nodes to agree without having identical values (Hansen & Gebhart, 2020; Bodnar et al., 2022). In *Temporal Sheaf Diffusion* (TSD), persistent node memories determine these maps, and spatial propagation supplies feedback to memory. This interface accommodates continuous dynamics and event-driven observations: history can change how values are compared even on a fixed graph. Figure 1 shows the evaluated recurrence; Appendix A includes the conceptual overview in Fig. 3.

Our central result characterizes polynomial memory regularized by a prescribed evolving sheaf. Its exact state combines signal moments with operator-history moments, from which a positive-definite Gram system recovers the coefficients. We derive continuous dynamics and exact held-history updates. A one-mode example on an ordinary graph distinguishes this solution from filtering signal moments with current geometry and from compressing already filtered observations. A second example shows how historical geometry couples polynomial degrees. For fixed geometry, the three constructions coincide.

GraphSSM introduces the historical projection objective (Li et al., 2024), and CTDG-SSM already notes its distinction from current-geometry regularization (Raghuvanshi et al., 2026, Appendix C.1). Our contribution is the exact finite-polynomial solution: pointwise filtering under changing geometry can leave the chosen polynomial space, whereas the Gram construction retains that constraint. Temporal sheaf maps also have precedents in ST-Sheaf GNN and TSNN (Mostafa et al., 2026; Asif et al., 2026). Our incidence-specific orthogonal maps allow cycle transports beyond a single frame per node (Appendix A.6).

Retaining historical geometry has a computational cost: even with compressed operator moments, coefficient recovery couples spatial coordinates. We also study a compact predictor with nodewise HiPPO-initialized memory (Gu et al., 2020) and finite sheaf propagation. Its learned state implements the memory–geometry feedback interface. A coupled continuous model and a consistent physical-

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.

time scheme clarify the relationship between evolution in physical time and observation-driven computation.

A sensor-reconstruction study tests the exact memories against clean histories: historical geometry helps when sensors physically change orientation, whereas current geometry helps when a calibration correction applies retrospectively. Filter-then-compress gives similar errors in the central setting, separating the benefit of observation-time geometry from that of the coupled solve. We also assess the compact architecture on seven TGB link tasks with controls that share its inputs and scoring head (Huang et al., 2023; Gastinger et al., 2024a).

2 PRELIMINARIES

Temporal graphs. The evaluated model observes snapshots or event batches k at times t_k, with gaps $\Delta_k = t_k - t_{k-1}$. Let $G_k = (V_k, E_k)$ be the propagation graph and $x_{u,k} \in \mathbb{R}^{d_x}$ the available node input. State is indexed by persistent node identity. Link events are predicted before incorporation into state. Physical time t_k is distinct from spatial computation depth within a snapshot.

Sheaves encode how endpoints agree. A cellular sheaf $\mathcal{F}$ on a graph assigns vector spaces to nodes and edges, and a linear restriction map $R_{e\leftarrow u} : \mathcal{F}(u) \to \mathcal{F}(e)$ to each incident node–edge pair. We use node and edge spaces $\mathbb{R}^{d_s}$. A node representation $z_u \in \mathbb{R}^{d_s \times f}$ has f feature channels; stacking the nodes gives a 0-cochain $Z \in \mathbb{R}^{nd_s \times f}$. On an edge $e = \{u, v\}$, the discrepancy is

$$(\delta_{\mathcal{F}} Z)_e = R_{e\leftarrow u} z_u - R_{e\leftarrow v} z_v, \qquad L_{\mathcal{F}} = \delta_{\mathcal{F}}^{\top} \delta_{\mathcal{F}}. \tag{1}$$

The endpoint maps put the two representations into a shared edge space before comparison. Thus, agreement need not require identical node features. Cochains with zero discrepancy on every edge are the sheaf's global sections. Choosing all restriction maps as identities recovers the usual graph Laplacian applied independently to each stalk coordinate (Hansen & Gebhart, 2020; Bodnar et al., 2022).

From sheaf geometry to propagation. Let $\widetilde{L}_{\mathcal{F}} = S L_{\mathcal{F}} S$ be a symmetrically normalized sheaf Laplacian, where S is a block-diagonal degree scaling. It is positive semidefinite, and its energy is

$$\mathcal{E}_{\mathcal{F}}(Z) = \operatorname{tr}(Z^{\top} \widetilde{L}_{\mathcal{F}} Z) = \sum_{e=\{u,v\} \in E} \| R_{e\leftarrow u} S_u z_u - R_{e\leftarrow v} S_v z_v \|_F^2 , \tag{2}$$

where each undirected edge is counted once. For a fixed sheaf, linear heat flow $dZ/ds = -\widetilde{L}_{\mathcal{F}} Z$ decreases this energy. This characterizes the geometry of consistency. The evaluated block instead uses $I - \widetilde{L}_{\mathcal{F}}$ inside a learned residual update, with feature mixing and activation (Bodnar et al., 2022); even its linear form has an amplifying reaction term.

3 MEMORY UNDER CHANGING SHEAF GEOMETRY

We first isolate the compression problem from the learned predictor. Let $X(s) \in \mathbb{R}^p$, $p = nd_s$, be a signal in a fixed identified cochain space, and let $\mathcal{L}(s) \succeq 0$ be a *prescribed* sheaf Laplacian path. Assume both are bounded and piecewise continuous on a finite horizon. The path is held fixed during minimization.

3.1 HISTORICAL GEOMETRY COUPLES TEMPORAL COEFFICIENTS

Let $\pi_t(s) \in \mathbb{R}^N$ contain the shifted Legendre polynomials orthonormal under $\mu_t(ds) = ds/t$ on $[0, t]$, with $t > 0$. A coefficient matrix $H \in \mathbb{R}^{p \times N}$ reconstructs the past as $H\pi_t(s)$. For $\alpha \geq 0$, consider

$$J_t(H) = \frac{1}{t} \int_0^t \left\{ \|X(s) - H\pi_t(s)\|^2 + \alpha \left[H\pi_t(s)\right]^{\top} \mathcal{L}(s) [H\pi_t(s)] \right\} ds. \tag{3}$$

The second term favors agreement under the comparisons that held at time s. For example, reconstructing archived sensor measurements under changing pairwise calibrations calls for historical geometry when the reconstruction must respect the calibration valid at each observation. On a small

fixed node set, the coupled solve provides an exact reference for this objective. This is the historical graph-regularized projection principle of GraphSSM (Li et al., 2024), with a sheaf operator specifying endpoint comparisons. CTDG-SSM already highlights the difference between historical and current Laplacians in this penalty (Raghuvanshi et al., 2026, Appendix C.1). We give an explicit state and exact update for this finite-polynomial historical problem.

Exact characterization. Set $Q(s) = I_p + \alpha\mathcal{L}(s)$ and define $C(t) \in \mathbb{R}^{p\times N}$ and $K(t) \in \mathbb{R}^{pN\times pN}$ by

$$C(t) = \frac{1}{t}\int_0^t X(s)\pi_t(s)^\top ds, \qquad K(t) = \frac{1}{t}\int_0^t [\pi_t(s)\pi_t(s)^\top] \otimes Q(s)\, ds. \tag{4}$$

Use the LegS generator and input vector, indexed from zero,

$$(A_H)_{ij} = \begin{cases} -\sqrt{(2i+1)(2j+1)}, & i > j, \\ -(i+1), & i = j, \\ 0, & i < j, \end{cases} \qquad (b_H)_i = \sqrt{2i+1}, \quad 0 \le i, j < N. \tag{5}$$

Set $R_H = A_H + \frac{1}{2}I_N$ and $\tau = \log(t/t_{\mathrm{ref}})$, $t_{\mathrm{ref}} > 0$.

Theorem 3.1 (Exact historical polynomial memory)**.** *For the prescribed bounded, piecewise-continuous signal and geometry above, the unique minimizer of Eq.* (3) *is*

$$\operatorname{vec} H_{\mathrm{hist}}(t) = K(t)^{-1}\operatorname{vec} C(t), \qquad K(t) \succeq I_{pN}, \tag{6}$$

using columnwise vectorization. The states satisfy Eq. (7) *almost everywhere. If X, Q are held on each interval $[t_k, t_{k+1})$, Eq.* (8) *is exact and preserves $K \succeq I$, starting from history-consistent states at $t_0 > 0$.*

Proof outline. Expanding J_t gives Hessian $2K$, while orthonormality and $Q \succeq I$ imply the lower bound. Differentiating the moving-basis integrals gives the ODE; integrating it under held inputs gives the recurrence (full proofs: Theorem B.5 and Proposition B.6). For a fixed geometry path, changing the signal from X to Y changes the coefficients by at most $\|X - Y\|_{L^2(\mu_t)}$ in Frobenius norm.

The ordinary signal moments C alone generally do not determine the historical projection. For example, when the geometry changes, the (i, j) block of K contains a weighted history of $Q(s)$ with weight $\pi_{t,i}(s)\pi_{t,j}(s)$. Different polynomial degrees can therefore be coupled even though the basis is orthonormal without that weight.

Finite-state continuous and discrete dynamics. The closed state equations in logarithmic time are

$$\begin{aligned} \frac{dC}{d\tau} &= CA_H^\top + X(t)b_H^\top, \\ \frac{dK}{d\tau} &= (R_H \otimes I_p)K + K(R_H^\top \otimes I_p) + (b_H b_H^\top) \otimes Q(t). \end{aligned} \tag{7}$$

This follows by differentiating the two integrals in Eq. (4); the half-identity shift accounts for the two basis factors in K. For $X = X_k$ and $Q = Q_k$ held on $[t_k, t_{k+1})$, let $\eta_k = \log(t_{k+1}/t_k)$, $E_k = e^{\eta_k A_H}$, $J_k = e^{\eta_k R_H}$, and $v_k = A_H^{-1}(E_k - I_N)b_H$. Then

$$\begin{aligned} C_{k+1} &= C_k E_k^\top + X_k v_k^\top, \\ K_{k+1} &= (J_k \otimes I_p)K_k(J_k^\top \otimes I_p) + (I_N - J_k J_k^\top) \otimes Q_k. \end{aligned} \tag{8}$$

No small-step approximation is required for held histories. On a first held interval $(0, t_0]$ with values X_0, Q_0, initialize $C(t_0) = X_0 e_0^\top$ and $K(t_0) = I_N \otimes Q_0$, where $e_0 = (1, 0, \ldots, 0)^\top$.

The state need not retain the observation sequence or the full Gram matrix: products of the first N polynomials have degree at most $2N - 2$, so $2N - 1$ operator-valued moments suffice to reconstruct every block of K (Proposition B.7). The exact state can therefore be stored as $(C, \mathcal{M}_0, \ldots, \mathcal{M}_{2N-2})$, where $\mathcal{M}_r = t^{-1}\int_0^t Q(s)\pi_{t,r}(s)\, ds$. Persistent geometry storage is $O(p^2N)$ rather than $O(p^2N^2)$, excluding transition/product tables and coefficient-solve workspace. Reconstructing $K_{ij} = \sum_r \gamma_{ijr}\mathcal{M}_r$ with the basis-product coefficients of Proposition B.7 then permits recovery of H_{hist} through the coupled pN-dimensional system $K \operatorname{vec} H_{\mathrm{hist}} = \operatorname{vec} C$. These equations provide an exact reference recursion for historical compression.

3.2 THREE WAYS TO COMBINE GEOMETRY AND COMPRESSION

The historical objective couples geometric regularization and polynomial approximation in one constrained problem. Two other natural constructions apply geometry before or after temporal compression. They answer different reconstruction questions.

Current geometry: compress, then filter. Replacing $\mathcal{L}(s)$ by $\mathcal{L}(t)$ in Eq. (3) gives

$$H_{\mathrm{cur}}(t) = Q(t)^{-1}C(t). \tag{9}$$

This reinterprets the retained history under today's comparisons. The same resolvent form appears in CTDG-SSM's filtered reconstruction objective (Raghuvanshi et al., 2026), although its loss differs from the additive penalty here (Appendix B.6). Smoothly varying geometry contributes $-Q^{-1}\dot{Q}H_{\mathrm{cur}}$ to the coefficient dynamics; geometry jumps require the corresponding coefficient transfer (Proposition B.8).

Input geometry: filter, then compress. Alternatively, one can regularize each observation when it arrives and store ordinary polynomial moments of that filtered signal:

$$H_{\mathrm{in}}(t) = \frac{1}{t}\int_0^t Q(s)^{-1}X(s)\pi_t(s)^\top ds, \qquad \frac{dH_{\mathrm{in}}}{d\tau} = H_{\mathrm{in}}A_H^\top + Q(t)^{-1}X(t)b_H^\top. \tag{10}$$

This formula defines the canonical LegS compression of filtered inputs, with history-consistent initialization and no geometry-dependent coefficient resets. GraphSSM's filtered-input update order motivates this construction (Li et al., 2024); the equation above fixes its clock and state semantics explicitly. It does not generally minimize the historical objective.

Why pointwise filtering does not solve the constrained problem. Let $\Pi_t X$ be ordinary polynomial projection. The pointwise expression $Q(s)^{-1}(\Pi_t X)(s)$ in GraphSSM's projection derivation (Li et al., 2024, Appendix C.1, Eq. (28)) need not remain in the finite polynomial space when Q varies. It is then not an admissible reconstruction for that derivation's historical objective. Equations (4)–(8) retain this constraint explicitly through the Gram state. This correction concerns the finite-polynomial projection factorization.

An ordinary-graph example. On $[0, 1]$, take a three-node path 1–2–3 until time $1/2$, and a triangle afterward. Use symmetric normalized Laplacians without self-loops or a degree offset, $\alpha = 1$, one constant polynomial mode ($N = 1$), and the constant signal $X(s) = v = (1, 0, -1)^\top/\sqrt{2}$. The vector v is a shared eigenvector with eigenvalues 1 and $3/2$, so $Q(s)$ acts by 2 and $5/2$ on this mode. Direct integration gives

$$H_{\mathrm{hist}}(1) = \frac{4}{9}v, \qquad H_{\mathrm{cur}}(1) = \frac{2}{5}v, \qquad H_{\mathrm{in}}(1) = \frac{9}{20}v. \tag{11}$$

The pointwise-filtered ordinary projection is a step function taking values $v/2$ and $2v/5$, outside the constant-polynomial space. The distinction therefore occurs even with one temporal mode on an ordinary graph. If the graph were a triangle throughout, $C(1)$ and $Q(1)$ would be unchanged but $H_{\mathrm{hist}}(1)$ would instead be $2v/5$: signal moments and current geometry alone do not determine the historical optimum.

For constant geometry, all three constructions coincide because $K = I_N \otimes Q$. Each sheaf eigenmode of eigenvalue λ is then attenuated by $(1 + \alpha\lambda)^{-1}$, preserving $\ker \mathcal{L}$ (global sections in the corresponding normalization coordinates). With evolving geometry, the choice is semantic: retain the comparisons made at observation time, reinterpret history under current comparisons, or compress observations already filtered on arrival. Appendix B.6 gives a two-node, two-mode example where historical regularization also couples polynomial degrees.

4 TEMPORAL SHEAF DYNAMICS AND THE EVALUATED PREDICTOR

The projection in Section 3 takes geometry as prescribed. We now let memory determine the geometry, giving a coupled dynamical interface and then a trainable recurrence. These constructions share endpoint comparisons and temporal feedback.

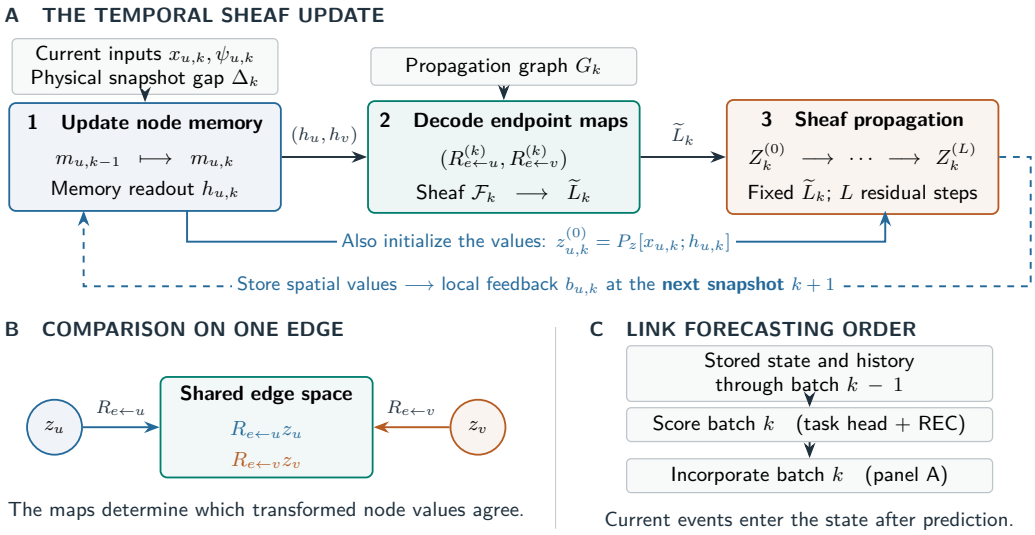

Figure 1: **The evaluated temporal sheaf recurrence.** **(A)** Memories incorporate current inputs, elapsed time, and lagged spatial feedback. Their readouts determine restriction maps and initialize spatial values. One sheaf governs a finite propagation block; stored values inform the next memory update. Physical time conditions memory, while propagation steps measure computational depth. **(B)** Endpoint maps transform values into a shared edge space before comparison. **(C)** Link predictions precede incorporation of each batch. The prediction head includes a separate recurrence-aware path (REC).

4.1 A CONTINUOUS INTERFACE AND CONSISTENT PHYSICAL-TIME STEPS

On a fixed finite graph, decode endpoint maps from memories $h_u = \rho(m_u)$ and set $\mathsf{L}(t,m) = (D+\varepsilon I)^{-1/2}\delta^\top \delta (D+\varepsilon I)^{-1/2}$, where $D = \mathrm{blockdiag}(\delta^\top\delta)$ and $\varepsilon > 0$. A continuously driven TSD system is

$$\begin{aligned} q_u(t,Z) &= [x_u(t); \mathcal{R}_u(t,Z); \psi_u(t)], \\ \dot{m}_u &= r(q_u)\{Am_u + B(q_u)q_u\}, \\ \dot{Z} &= f(t,m,Z) + U(t), \qquad f = \sigma[(I - \mathsf{L}(t,m))(I_n \otimes W_1)ZW_2]\,. \end{aligned} \tag{12}$$

Here $\mathcal{R}_u$ reads local spatial feedback, ψ_u contains graph descriptors, $r \geq 0$ selects a rate, and $U(t)$ is an exogenous spatial input. Under the bounded-input, affine-feedback, and Lipschitz assumptions in Appendix B.1, this system has a unique trajectory on every finite horizon (Theorem B.2; Proposition B.4). The normalized bound $0 \preceq \mathsf{L} \preceq 2I$ first bounds spatial growth independently of memory, then bounds the feedback driving memory. This establishes finite-horizon well-posedness.

A consistent physical-time step solves memory with frozen inputs over $h_k = t_{k+1} - t_k$, then takes a spatial Euler step, including $h_k U(t_k)$. Under joint time/state Lipschitz bounds, its global error is $O(\max_k h_k)$; finitely many prescribed changes require event-aligned steps and locally Lipschitz transfers (Appendix B.2). Stage reversal preserves this limit while changing finite-step states.

4.2 A COMPACT RECURRENCE FOR OBSERVED GRAPHS

Exact historical memory retains operator-history information and recovers coefficients through a coupled spatial system. The evaluated model instead allocates fixed-width memory to each node, learns its transition from prediction loss, and processes relations through sparse sheaf propagation. HiPPO initializes the temporal generator and input bias. This nodewise state is learned for prediction rather than computed from the exact Gram system. Appendix C relates these choices to trainable SSMs and temporal graph memories (Gu et al., 2021; Gu & Dao, 2023; Rossi et al., 2020).

For nodes $u \in \mathcal{U}_k$ in an update batch and its selected neighborhood, current inputs and lagged spatial feedback $b_{u,k-1}$ drive

$$q_{u,k} = [x_{u,k}; b_{u,k-1}; \psi_{u,k}], \quad m_{u,k} = \Psi_\eta(m_{u,k-1}, q_{u,k}; \Delta_k), \quad h_{u,k} = \rho(m_{u,k}). \tag{13}$$

Unselected memories are held. The effective step is selected from $q_{u,k}$ and the gap Δ_k between processed snapshots, then capped. For each edge $e = \{u, v\}$, a shared decoder gives orthogonal endpoint maps

$$(R^{(k)}_{e\leftarrow u}, R^{(k)}_{e\leftarrow v}) = \Phi_\theta(h_{u,k}, h_{v,k}), \qquad R^{(k)}_{e\leftarrow u}, R^{(k)}_{e\leftarrow v} \in \mathbb{R}^{d_s \times d_s}. \tag{14}$$

Their cross-node Laplacian block is $-(R^{(k)}_{e\leftarrow u})^\top R^{(k)}_{e\leftarrow v}$: history changes the transformed disagreement, not only the propagated values. The decoder uses endpoint memories, so different relation labels on the same endpoint pair share maps at an update; relation labels also enter the scoring head.

Initialize $z^{(0)}_{u,k} = P_z[x_{u,k}; h_{u,k}]$ and perform L residual steps under the same sheaf:

$$Z^{(\ell+1)}_k = Z^{(\ell)}_k + \tau_\ell \, \sigma\Big[(I - \widetilde{L}_k)(I_{n_k} \otimes W_1) Z^{(\ell)}_k W_2\Big], \quad \ell = 0, \dots, L-1, \quad \tau_\ell > 0. \tag{15}$$

The matrices W_1, W_2 mix stalk coordinates and feature channels. Geometry and weights are reused within the block; training dropout is omitted here. The τ_ℓ are computational-depth steps independent of Δ_k. With identity activation and unit mixing, the field is $Z - \widetilde{L}_k Z$, containing both reaction and heat diffusion. The reaction term can amplify values; Eq. (43) bounds growth over the finite propagation block.

4.3 Feedback and prediction order

Store terminal spatial values and updated memories. A local aggregation of the stored values supplies the next batch's feedback, completing

$$\text{memory} \longrightarrow \text{endpoint maps} \longrightarrow \text{sheaf propagation} \longrightarrow \text{memory feedback}. \tag{16}$$

For link forecasting, score batch k from the state preceding that batch, then incorporate its events (Algorithm 1). The head includes a separate recurrence-aware path (REC) using observed interaction counts and recency. The neural backbone is held between observations; query-time dependence enters REC. Appendix A specifies the recurrence, inputs, local graph, and scoring head.

The normalized operator satisfies the spectral bound, the memory transition is exact for held inputs on its selected clock, and propagation obeys the conditional depth bound. Appendix C develops the computational rationale for these choices; Appendix D analyzes their temporal properties.

5 Synthetic reconstruction: which geometry should memory retain?

We test the exact memories on a sensor array whose comparisons change with sensor orientation. Twenty sensors observe a smooth two-dimensional field with sensor-specific deviations and noise; half change orientation midway through the window. In scenario A, a physical reorientation changes subsequent readings. In scenario B, a calibration correction means that the final orientation applies to the entire clean history. Both scenarios supply the same old-to-new geometry path. At window end, we reconstruct the clean history with $N = 16$ modes, tuning α separately for each method and scenario on ten draws and evaluating on 50 fresh draws (Appendix F).

At a 45° change and 0 dB SNR, historical memory reduces pre-change relative error from 0.2156 to 0.1807 in A, a 16.2% reduction relative to current filtering. The preference reverses in B: current filtering gives 0.1681, versus 0.2024 for historical memory. The paired error differences (historical minus current) are -0.0349 $[-0.0393, -0.0305]$ and $+0.0343$ $[0.0299, 0.0386]$, respectively (95% bootstrap intervals). Whole-window errors have the same ordering. At 10 dB, the pre-change gaps shrink to 0.0016–0.0019; the benefit depends on how much regularization the noise warrants.

Filter-then-compress reaches 0.1807 in A and 0.2024 in B at the 0 dB setting. Thus the crossover supports choosing geometry according to the reconstruction semantics, without establishing a reconstruction advantage for the coupled Gram solve. Numerical checks separately verify the exact recursions against direct integration, with relative discrepancies below 4×10^{-13} for the Gram and operator-moment states (Appendix F.3).

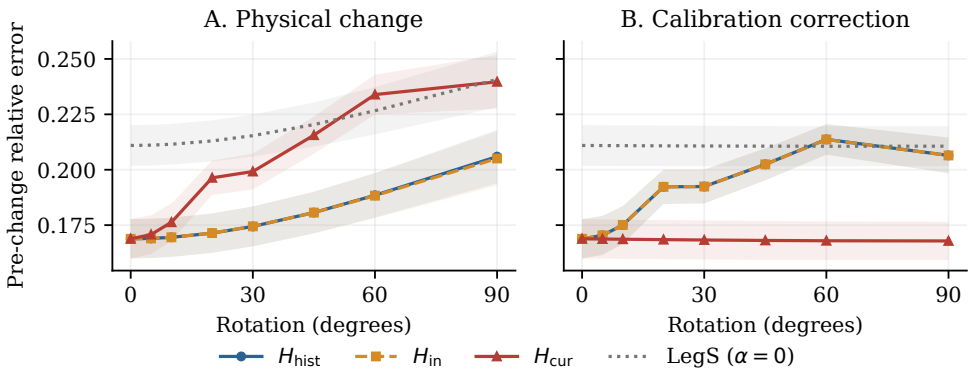

Figure 2: Recovering the history before the change, at 0 dB SNR. Historical geometry helps for physical changes (A); current geometry helps for retrospective calibration (B). Lines are mean relative reconstruction errors over 50 draws; bands are mean ± 1.96 standard errors. H_{hist} and H_{in} largely overlap. Each point uses regularization selected on separate whole-window validation histories.

6 A PRACTICAL ARCHITECTURE STUDY

Protocol. We assess the compact TSD predictor on seven TGB link tasks using official splits, evaluators, and negative sets (Huang et al., 2023; Gastinger et al., 2024a). Each snapshot is scored before its events enter the state; evaluation streams through validation and test without parameter updates. All headline link configurations include the recurrence-aware scorer (REC), with task-specific relation and node-type inputs. Table 1 reports three- or five-seed results and separately lists the descriptive node tasks. Training suffixes, validation-prefix selection, and compute are specified in Appendix E.2.

Table 1: Test performance: NDCG@10 for node tasks (tgbn), MRR otherwise; mean $\pm$ sample standard deviation. Wiki, ICEWS, Forum, and Trade use five seeds; the other rows use three. Rank r/N inserts the TSD mean among the $N-1$ listed TGB entries on September 21, 2026, counting named variants and heuristics separately. Budgets differ. Node results are descriptive because schedules differ; Trade reconstructs already observed edges (Appendix E.1).

Dataset	TSD	TSD rank	Leading reference	Test
`tgbl-wiki`	0.762 ± 0.003	5/22	TPNet	0.827
`tkgl-smallpedia`	0.610 ± 0.007	2/8	CEN	0.612
`tkgl-polecat`	0.244 ± 0.003	1/8	TLogic	0.228
`tkgl-icews`	0.329 ± 0.011	1/8	Recurrency (train)	0.211
`tkgl-wikidata`	0.540 ± 0.009	1/3	EdgeBank	0.535
`thgl-software`	0.438 ± 0.002	3/7	STHN	0.731
`thgl-forum`	0.625 ± 0.010	3/6	TGN (edge type)	0.729
Descriptive node tasks: external protocols differ				
`tgbn-trade`	0.664 ± 0.006	—	—	—
`tgbn-genre`	0.448 ± 0.003	—	—	—

Complete-predictor performance. TSD is most competitive on temporal knowledge graphs, reaching 0.244 MRR on POLECAT and 0.329 on ICEWS and approaching CEN on Smallpedia. Its mean advantage over EdgeBank on Wikidata is smaller than its across-seed standard deviation. Wiki, Software, and Forum remain below their leading listed references.

Component comparison. Controls share task inputs, REC, and observation schedules, with different capacities. TSD improves over *core off*, which removes readable recurrent memory and spatial propagation jointly, by 0.027 MRR on Wiki and 0.010 on Forum. Identity maps and smaller ordinary recurrent cores have similar Wiki means and higher Forum means; on ICEWS, identity maps and GRU

both reach 0.334, versus TSD's 0.329. These comparisons establish no consistent predictive return for memory-conditioned maps. Appendix E.4 gives the full control results in Table 7; Appendices E.5 and E.6 report history strata, timing diagnostics, and synthetic studies.

7 Related work

Geometry and temporal memory. Neural Sheaf Diffusion learns endpoint comparisons for graph propagation (Hansen & Gebhart, 2020; Bodnar et al., 2022); connection and heterogeneous variants impose further geometric structure (Barbero et al., 2022; Braithwaite et al., 2024). ST-Sheaf GNN and TSNN extend sheaf geometry over time (Mostafa et al., 2026; Asif et al., 2026). TSD combines persistent selective memory, incidence-specific orthogonal maps, and spatial feedback. Orthogonal and diagonal maps impose different constraints; neither universally contains the other. The distinction from node frames in Appendix A.6 concerns cycle transports rather than whole-network expressivity.

HiPPO and structured SSMs connect polynomial compression to continuous dynamics and trainable recurrence (Gu et al., 2020; 2021; Gu & Dao, 2023). GraphSSM introduces historical graph-regularized projection (Li et al., 2024); CTDG-SSM studies current graph-filtered memory and its correction, explicitly noting the historical/current distinction (Raghuvanshi et al., 2026, Appendix C.1). Section 3 solves the finite-polynomial historical problem by retaining operator history and identifies why pointwise filtering can violate its constraint. The result applies to prescribed positive-semidefinite paths; sheaves interpret those operators through endpoint comparisons.

Temporal prediction and evaluation. TGN and DyRep maintain persistent node histories (Rossi et al., 2020; Trivedi et al., 2019); attention and neighborhood models offer other representations (Xu et al., 2020; Yu et al., 2023; Cong et al., 2023). TGB and TGB 2.0 standardize prediction tasks (Huang et al., 2023; Gastinger et al., 2024a). EdgeBank, TGB-Seq, CRAFT, and modular evaluations motivate separating recurrent interactions from learned representation gains (Poursafaei et al., 2022; Yi et al., 2025b;a; Yang et al., 2024), as our same-head comparisons do. Work on batch forecasting and truncated backpropagation identifies limits on available history and delayed credit (Lampert et al., 2026; Bravo et al., 2024); Appendix D relates these to TSD's update.

8 Discussion

Historical geometric memory requires information about past operators as well as signal values. The moment/Gram construction makes that requirement explicit and supplies an exact recursion whose state size is independent of history length. Its separation from current filtering and compression of filtered inputs already appears on an ordinary graph with one temporal mode. The sensor study gives this choice an empirical interpretation: historical comparisons help recover physically changing readings, while corrected current comparisons help recover a retrospectively calibrated history. Filter-then-compress captures nearly all of the historical solution's benefit in the central setting. The exact Gram state characterizes the objective, but its additional cost needs a task-specific justification. This perspective also guides practical design: nodewise learned memory and sparse sheaf propagation allocate temporal compression and relational comparison to separate modules, whose contribution to prediction can be tested with shared-head controls.

Limitations. The reconstruction study uses prescribed geometry and clean validation histories to select regularization; its results do not establish a forecasting benefit. The consistent physical-time scheme remains unevaluated, and the learned recurrence has no established approximation bound to the exact states. Public-reference budgets and control capacities differ. The fixed context graph uses the first 50k retained training edges, exposing later edges in that prefix to early training updates; validation and test use no future edges. Capped clocks, observation batching, and truncated gradients can remove temporal distinctions or learning paths, but the diagnostics do not establish these as causes of the predictive gaps (Appendix D). Full protocol, numerical, and model comparisons are documented in Appendix E.

AI USE STATEMENT

The original continuous and discretized updates and their initialization were developed by hand. Claude assisted with developing model implementation details that are not essential to the theoretical construction, and with the choice of the recurrence-aware prediction head. Claude also assisted with implementing the synthetic data generator and reconstruction experiments. OpenAI Codex assisted with manuscript revision, figure and LaTeX preparation, inspection of implementation code and saved experimental outputs, recomputation of descriptive summaries, interpretation of results, and suggestions for follow-up experiments. Codex also assisted with simulated independent reviews, a critical theory audit, correction of mathematical formulations, derivation and checking of proofs and analytic examples, literature research on practical implementation choices, and checking their correspondence with the evaluated model.

REPRODUCIBILITY STATEMENT

Appendix A specifies the evaluated model, state updates, and prediction order. Appendix B gives mathematical assumptions, proofs, and discretization formulas; Appendix C explains the literature-grounded implementation choices and their computational trade-offs; Appendix D analyzes the temporal distinctions and gradient paths those choices can remove. Appendix E.1 describes the data and evaluation protocol, Appendix E.2 reports configurations and selection procedures, and Appendix E.4 provides component comparisons. The reconstruction study reports variability across independently generated histories, with regularization selected on separate draws. The prediction results distinguish across-seed variability and external leaderboard references; additional stratified and synthetic analyses are documented in Appendices E.5 and E.6.

Code and reproduction scripts will be released publicly after acceptance. A clean-environment rerun of the complete synthetic reconstruction study reproduced all 350 regularization selections exactly and all 32 means in Table 15 to four decimal places. The quoted paired contrasts, crossover count, and numerical-equivalence bounds were also reproduced; Appendix F.4 documents the procedure and numerical agreement.

REFERENCES

Md Sadek Hossain Asif, Tanzila Khan, and Md. Mosaddek Khan. Temporal sheaf neural networks with dynamic orthogonal transport, 2026. URL https://arxiv.org/abs/2606.10071. Version 1.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. Sheaf neural networks with connection laplacians. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 28–36. PMLR, 2022.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35:18527–18541, 2022.

Luke Braithwaite, Iulia Duta, and Pietro Liò. Heterogeneous sheaf neural networks. *arXiv preprint arXiv:2409.08036*, 2024.

João Bravo, Jacopo Bono, Pedro Saleiro, Hugo Ferreira, and Pedro Bizarro. Mind the truncation gap: challenges of learning on dynamic graphs with recurrent architectures. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2412.21046.

Ben Chamberlain, James Rowbottom, Maria I. Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. GRAND: Graph neural diffusion. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418, 2021. URL https://proceedings.mlr.press/v139/chamberlain21a.html.

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? *arXiv preprint arXiv:2302.11636*, 2023.

Zifeng Ding, Yifeng Li, Yuan He, Antonio Norelli, Jingcheng Wu, Volker Tresp, Yunpu Ma, and Michael Bronstein. Dygmamba: Efficiently modeling long-term temporal dependency on continuous-time dynamic graphs with state space models. *Transactions on Machine Learning Research*, 2024.

Or Feldman, Krishna Sri Ipsit Mantri, Moshe Eliasof, and Chaim Baskin. Revisiting node affinity prediction in temporal graphs. *arXiv preprint arXiv:2510.06940*, 2025.

Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, Reihaneh Rabbany, and Guillaume Rabusseau. Tgb 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. *arXiv preprint arXiv:2406.09639*, 2024a.

Julia Gastinger, Christian Meilicke, Federico Errica, Timo Sztyler, Anett Schuelke, and Heiner Stuckenschmidt. History repeats itself: A baseline for temporal knowledge graph forecasting. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024b.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33: 1474–1487, 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

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

Moritz Lampert, Christopher Blöcker, and Ingo Scholtes. From link prediction to forecasting: Addressing challenges in batch-based temporal graph learning. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2406.04897.

Ce Li, Rongpei Hong, Xovee Xu, Goce Trajcevski, and Fan Zhou. Simplifying temporal heterogeneous network for continuous-time link prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 1288–1297, 2023.

Jintang Li, Ruofan Wu, Xinzhou Jin, Boqun Ma, Liang Chen, and Zibin Zheng. State space models on temporal graphs: A first-principles study. *arXiv preprint arXiv:2406.00943*, 2024.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 408–417, 2021.

Zixuan Li, Saiping Guan, Xiaolong Jin, Weihua Peng, Yajuan Lyu, Yong Zhu, Long Bai, Wei Li, Jiafeng Guo, and Xueqi Cheng. Complex evolutional pattern learning for temporal knowledge graph reasoning. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 2: short papers)*, pp. 290–296, 2022.

Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4120–4127, 2022.

Xiaodong Lu, Leilei Sun, Tongyu Zhu, and Weifeng Lv. Improving temporal link prediction via temporal walk matrix projection. *Advances in Neural Information Processing Systems*, 37:141153–141182, 2024.

Abeer Mostafa, Raneen Younis, and Zahra Ahmadi. Dynamic sheaf diffusion networks with adaptive local structure for heterogeneous spatio-temporal graph learning, 2026. URL https://arxiv.org/abs/2604.11275. Version 2.

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.

Ayushman Raghuvanshi, Thummaluru Siddartha Reddy, Sundeep Prabhakar Chepuri, and Mahesh Chandran. Learning long range spatio-temporal representations over continuous time dynamic graphs with state space models. *arXiv preprint arXiv:2606.04672*, 2026. URL https://arxiv.org/abs/2606.04672.

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.

Yulia Rubanova, Ricky T. Q. Chen, and David K. Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

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.

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

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

Yuxin Yang, Hongkuan Zhou, Rajgopal Kannan, and Viktor Prasanna. Towards ideal temporal graph neural networks: Evaluations and conclusions after 10,000 GPU hours. *Proceedings of the VLDB Endowment*, 18(4):956–969, 2024. doi: 10.14778/3717755.3717758. URL https://www.vldb.org/pvldb/vol18/p956-yang.pdf.

Lu Yi, Runlin Lei, Fengran Mo, Yanping Zheng, Zhewei Wei, and Yuhang Ye. Future link prediction without memory or aggregation. *arXiv preprint arXiv:2505.19408*, 2025a. URL https://arxiv.org/abs/2505.19408.

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 *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/db5ca61dbc08cf5143c05ad2d1b0b2ca-Paper-Conference.pdf.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *arXiv preprint arXiv:2303.13047*, 2023.

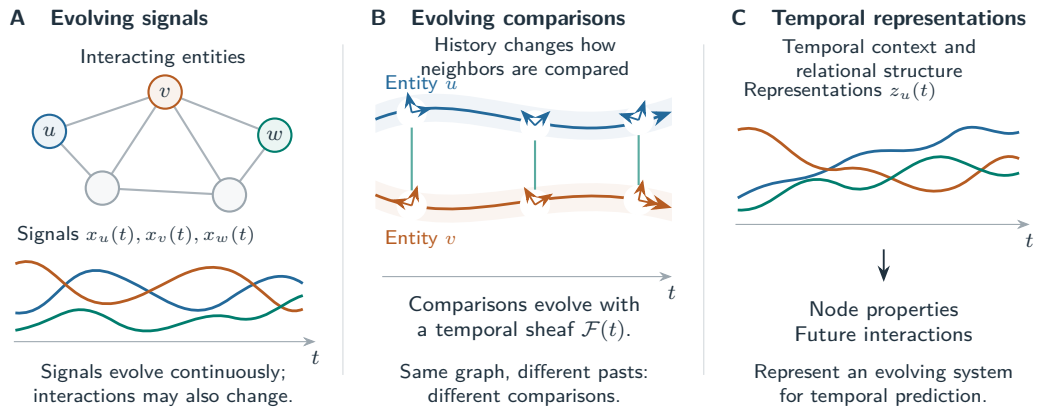

Figure 3: **What Temporal Sheaf Diffusion represents. (A)** Interacting entities carry signals that can evolve continuously in time. **(B)** Their histories shape a time-varying sheaf: the coordinate glyphs illustrate evolving endpoint comparisons, so identical present connections can have different meanings after different histories. **(C)** Propagation under these comparisons yields history-aware representations for temporal prediction and feedback to temporal evolution. Curves illustrate the continuous-time principle, not measured trajectories; topology changes or event-driven observations may introduce discontinuities. Fig. 1 details the sampled implementation used in our experiments.

A Temporal Sheaf Diffusion: Specification and Properties

This appendix specifies the model used in the experiments. Appendix C motivates its finite-state memory, update schedule, and spatial computation as practical realizations of the temporal sheaf framework. Section 4 introduces the recurrence; this appendix specifies its execution. Throughout, m denotes the persistent raw SSM state and $h = \rho(m)$ its readout; k indexes observed batches and ℓ indexes spatial computation.

A.1 Local graph, descriptors, and spatial feedback

The event implementation for link tasks starts from the nodes active in batch k and expands them by the configured number of closure hops. Expansion uses the fixed context graph when available and the current event graph otherwise. The propagation graph has $V_k = \mathcal{U}_k$ and is the union of localized context and current-event edges, made bidirectional with duplicates and self-edges removed. Directions and relation labels used for prediction are retained separately. The fixed context and task observation schedules are specified in Appendix E.1, including the training-prefix context convention; the causal state update alone does not imply that context construction is strictly online within the training split.

Let $z^{\text{store}}_{u,k-1} \in \mathbb{R}^{d_s f}$ be node u's stored spatial vector, which remains unchanged when the node is inactive. The feedback uses the preceding event graph, restricted to the current local node set. With $\mathcal{N}^{(k)}_{k-1}(u)$ denoting that neighborhood,

$$b_{u,k-1} = P_b\left(\frac{z^{\text{store}}_{u,k-1} + \sum_{v\in\mathcal{N}^{(k)}_{k-1}(u)} z^{\text{store}}_{v,k-1}}{1 + |\mathcal{N}^{(k)}_{k-1}(u)|}\right). \tag{17}$$

Here P_b is an affine projection to $\mathbb{R}^{d_z}$. When no previous spatial state exists, feedback is zero. With no preceding local edges, the readout uses the node's own stored value. The graph used to read feedback is therefore distinct from the graph on which the current sheaf is decoded.

The base local descriptor is $[\deg_{G_k}(u); 1]$: every node in the local update set, including closure nodes, receives an active flag of one. When enabled, the mean of incident relation embeddings is appended to this descriptor, so $q_{u,k} = [x_{u,k}; b_{u,k-1}; \psi_{u,k}]$. Inputs $x_{u,k}$ may include learned node or node-type embeddings in addition to supplied features. These input choices and the dimensions vary by task (Appendix E.2). The concrete edge decoder has no separate edge-feature argument: relation

information can reach it through the endpoint memories, but the aggregate does not identify which relation belongs to which neighbor. Distinct relations on the same endpoint pair share the same maps within an update.

Node-property implementation. Trade and Genre use the full snapshot variant of the same mechanism. Their propagation graph contains all registered nodes and the current snapshot edges, without merging a fixed context graph. Only memories in the active/closure set $\mathcal{U}_k$ are replaced, while spatial representations are recomputed for every node in V_k. Feedback uses the full preceding snapshot graph, and the descriptor is degree (optionally log-transformed) together with the actual active/closure mask. The selector, endpoint decoder, and residual propagation equations are shared with the local version. Thus locality of the operators does not imply that these node-property runs execute only on a sparse active node set.

A.2 Persistent selective state-space memory

For each updated node, the input selector produces $B_{u,k} \in \mathbb{R}^{d_h \times d_q}$ and an effective step $\delta_{u,k} > 0$. The shared generator is full lower triangular, not diagonal:

$$A = \mathrm{tril}_{-1}(A_{\mathrm{free}}) - \mathrm{diag}(\mathrm{softplus}(a)), \qquad B_{u,k} = \mathrm{reshape}(W_B q_{u,k} + b_B). \tag{18}$$

The step selector combines content and elapsed time:

$$\begin{aligned} s &= \mathrm{softplus}(s_0) + 10^{-4}, \\ \delta_{u,k} &= \min\left\{\delta_{\max},\, \mathrm{softplus}\left(w_q^\top q_{u,k} + b_q + w_t \log\left(1 + \frac{\max(\Delta_k, 0)}{c_\Delta s}\right)\right)\right\}. \end{aligned} \tag{19}$$

The fixed scale c_Δ expresses gaps in dataset units; the link runner's automatic setting uses the median positive training inter-snapshot gap. The default scale is one when not overridden. The standard cap is $\delta_{\max} = 0.25$. The first update uses $\Delta_0 = 0$, which does not force $\delta_{u,0} = 0$ because the selector also has a content term and bias. For subsequent updates, Δ_k is the global gap between processed snapshots, not the time since node u was last active. Inactive nodes receive no intervening SSM updates. Consequently this channel does not distinguish different node-inactivity durations under identical stored states and inputs. An unrelated processed batch may change the next global gap. A constant gap contributes a constant selector-bias term; its ablation alone would not establish irregular-time modeling.

Holding the selected input and parameters fixed over the effective step gives the exact zero-order-hold recurrence

$$\begin{aligned} E_{u,k} &= \exp(\delta_{u,k} A), \\ m_{u,k} &= E_{u,k} m_{u,k-1} + A^{-1}(E_{u,k} - I) B_{u,k} q_{u,k}, \qquad h_{u,k} = \rho(m_{u,k}). \end{aligned} \tag{20}$$

The implementation evaluates a matrix exponential and solves a linear system with A rather than explicitly forming its inverse. "Exact" refers to this held-input linear subproblem, up to numerical precision; it is not an exact solution of the entire nonlinear temporal graph model. The readout ρ is either the identity or LayerNorm, as configured. The stored state remains m, including when the downstream modules use a normalized readout. The state persists across snapshots.

HiPPO initialization. With indices $i, j = 0, \ldots, d_h - 1$, the initial generator and input vector use the LegS structure (Gu et al., 2020):

$$(A_H)_{ij} = \begin{cases} -\sqrt{(2i+1)(2j+1)}, & i > j, \\ -(i+1), & i = j, \\ 0, & i < j, \end{cases} \qquad (b_H)_i = \sqrt{2i+1}. \tag{21}$$

We initialize A to A_H and each input column of the reshaped selector bias b_B to $b_H/\sqrt{d_q}$, with small random selector weights. The content-step weights start at zero with a positive initial step sampled log-uniformly between 10^{-3} and 0.1 before gap conditioning. Training changes the generator and input-dependent selectors. This is a structured initialization inspired by HiPPO and selective SSMs (Gu et al., 2021; Gu & Dao, 2023), not the canonical time-rescaled LegS projection dynamics. Polynomial-projection optimality and timescale invariance are consequently not assumed for the trained system.

Algorithm 1 Temporal sheaf diffusion for link forecasting

Require: Timestamped event batches, node inputs, context graph, learned parameters
1: Initialize persistent memory and spatial state to zero; initialize history cache
2: **for** each batch k in chronological order **do**
3: **if** a preceding spatial state is available **then**
4: Score batch k from that state and previously ingested history
5: **end if**
6: Form the active/closure set $\mathcal{U}_k$ and its local propagation graph
7: Compute Δ_k and lagged feedback using Eq. (17)
8: Form $q_{u,k}$; update $m_{u,k}$ and read $h_{u,k}$ for $u \in \mathcal{U}_k$
9: Decode restriction maps once; assemble $\widetilde{L}_k$
10: Initialize $Z_k^{(0)}$ from current inputs and memory readouts
11: Apply L residual spatial steps under the fixed $\widetilde{L}_k$
12: Store updated node states, event graph, timestamp, and event history
13: **end for**

A.3 Restriction maps, normalization, and propagation

For each directed incidence corresponding to $e = \{u, v\}$, the shared endpoint decoder computes

$$a_{u\to v} = \tanh(W_\Phi[h_{u,k}; h_{v,k}]), \qquad R^{(k)}_{e\leftarrow u} = \mathrm{Orth}(a_{u\to v}). \tag{22}$$

The reverse ordered pair produces $R^{(k)}_{e\leftarrow v}$. The reported runs use the Householder-based orthogonal parameterization. Each pair of maps is decoded once per snapshot and reused across all spatial layers. The Laplacian is assembled from their products as in Eq. (1).

Since the restrictions are orthogonal, the block degree is $D_{uu} = \deg_{G_k}(u) I_{d_s}$. The evaluated connection-Laplacian builder uses augmented normalization:

$$S = (D + I_{n_k d_s})^{-1/2}, \qquad \widetilde{L}_k = S L_{\mathcal{F}_k} S. \tag{23}$$

Thus the diagonal block of $I - \widetilde{L}_k$ is $I_{d_s}/(\deg(u)+1)$ and the off-diagonal block along an edge is

$$(I - \widetilde{L}_k)_{uv} = \frac{(R^{(k)}_{e\leftarrow u})^\top R^{(k)}_{e\leftarrow v}}{\sqrt{(\deg(u)+1)(\deg(v)+1)}}. \tag{24}$$

This exposes the mechanism: neighboring values are transported between local stalk coordinates, scaled by degree, and combined with a self contribution. Endpoint restrictions can differ even though the assembled Laplacian is symmetric; this does not make the propagation Laplacian a directed operator. Directed and typed predictions are handled by the task head.

The initializer P_z is affine. The spatial block uses Eq. (15) with $\tau_\ell = \mathrm{softplus}(\theta_\ell)$, tied W_1, W_2, and ELU when activation is enabled. Training dropout is applied to the input of the vector field at each layer. If the local graph has no edges, the event implementation returns $Z_k^{(0)}$ directly. Updated states are scattered back by node identity, leaving all other stored values unchanged. Initial raw memories and stored spatial values are zero.

A.4 Update order

The first batch initializes state when no preceding representation is available; the predict–update loop then scores subsequent batches. Evaluation reconstructs historical state from the training stream before validation/test scoring, as specified in Appendix E.1.

A.5 Prediction heads, inputs, and training

Neural event scorer. Write $p = d_s f$ and let $z_u^- \in \mathbb{R}^p$ be the stored spatial representation available before incorporating the queried batch. The head uses

$$e_u = \mathrm{ELU}(W_x x_u^{\mathrm{head}} + b_x), \qquad v_u = \mathrm{LayerNorm}(W_{\mathrm{sig}} e_u + W_{\mathrm{state}} z_u^-), \tag{25}$$

$$s_{\mathrm{neural}}(u, r, v) = \langle W_{\mathrm{src}} v_u,\ a_r \odot W_{\mathrm{dst}} v_v \rangle + \beta + \beta_r. \tag{26}$$

Table 2: Head/input choices in the reported link runs. T, U, and S denote typed, untyped directed, and symmetric REC channels. Type embeddings enter the core input; they are not added directly to the head input.

Dataset	REC channels	Relation aggregate	Type embedding
Wiki	T	no	no
Smallpedia	T,U,S	yes	no
POLECAT	T,U	yes	no
ICEWS	T,U	yes	no
Wikidata	T,U,S	yes	no
Software	T,U	yes	yes
Forum	T,U	yes	yes

The signal, state, source, and destination matrices are bias-free $p \times p$ projections. LayerNorm has learned scale and bias. The relation factor $a_r \in \mathbb{R}^p$ and scalar bias β_r are learned; without a supplied relation ID or relation table we use $a_r = \mathbf{1}$ and $\beta_r = 0$. The reported link models have $p = 16$, zero input/hidden dropout, and one affine input encoder followed by ELU. Scores are raw, without a sigmoid or softmax. This head has no direct query-time input.

Recurrence-aware head (REC). The typed channel uses the directed key (u, r, v), with $r = 0$ when relations are absent. Optional channels use (u, v) and the symmetric key $(\min(u, v), \max(u, v))$. Each channel c stores a count n_c and latest timestamp ℓ_c for its key among previously ingested events. For query time t, define

$$a_c = \mathbf{1}[n_c > 0], \qquad d_c(t) = \begin{cases} \max(t - \ell_c, 0), & n_c > 0, \\ 0, & n_c = 0, \end{cases} \tag{27}$$

$$f(t) = \operatorname{concat}_{c=1}^{C}[a_c, \log(1 + n_c), \log(1 + d_c(t))], \tag{28}$$

$$g(f) = w_2^\top \operatorname{ReLU}(W_1^{\mathrm{REC}} f + b_1) + b_2, \tag{29}$$

$$s_{\mathrm{REC}}(u, r, v, t) = \begin{cases} g(f(t)), & \sum_c n_c > 0, \\ 0, & \sum_c n_c = 0, \end{cases} \tag{30}$$

$$s(u, r, v, t) = s_{\mathrm{neural}}(u, r, v) + s_{\mathrm{REC}}(u, r, v, t). \tag{31}$$

The MLP has input width $3C$, hidden width 16, ReLU, and a scalar affine output, with both affine biases enabled. Its output weight and bias are initialized to zero. If all channels are unseen, the bonus is exactly zero, independently of the MLP biases; if any is seen, all channel features enter the MLP, with zeros for unseen channels. There is no additional learned mixture coefficient. Counts include repeated events; recencies use original timestamp units and each query's event timestamp.

The implementation stores sorted keys, counts, and latest times, plus a pending buffer. Pending observations commit at the next state update; query-time lookup also counts pending observations with strictly earlier timestamps. Under the evaluated predict-from-previous protocol, those pending observations belong to an already ingested earlier batch. The queried batch itself is scored before ingestion, including events earlier within that batch. With an unseen key in every channel, changing only the future query time cannot change the complete score.

Relation IDs, unseen interactions, and core-off. The runner preserves relation IDs supplied by the dataset, including inverse-relation IDs, and allocates their neural factors using the dataset's relation vocabulary. It performs no additional inverse augmentation or inverse parameter tying. Inverse IDs therefore have separate typed keys and neural factors; untyped channels deliberately ignore those distinctions. The symmetric channel can count both directions when both are present in the input. In-vocabulary relations unseen during training have no special learned fallback; IDs outside the allocated vocabulary are unsupported. An unseen interaction is scored by the neural head, with the REC contribution defined by Eq. (30). Core-off keeps the pointwise input pathway, the current-input spatial projection, and the prediction head; it is not a REC-only predictor.

Node inputs and edge information. The loader uses supplied node features when available and correctly sized; otherwise it generates a fixed standard-normal node-feature matrix. Defaults are

dimension 64 and a dedicated feature seed 43; the feature seed is separate from the model-training seed. The reported runs disable learned node-ID embeddings. Software and Forum add learned node-type embeddings to core inputs. The head directly encodes the unaugmented node input in all reported link runs. Where enabled, the mean embedding of current raw incident relation IDs (including both endpoints) is appended to the recurrent input. Repeated events contribute to this mean, while propagation edges are deduplicated. Raw event message vectors and scalar edge weights are not otherwise passed to the core by this runner.

Training loss and negatives. For each positive query (u, r, v, t), we sample $K = 32$ destinations independently and uniformly with replacement from the configured destination-ID interval, excluding only v. That interval is the minimum to maximum destination ID in the loaded dataset, or the whole node vocabulary when no restriction is needed. Source, relation, and query timestamp remain fixed. Other true destinations are not filtered, and there is no additional relation/type validity filter. The reported runs use

$$\mathcal{L}_k = \frac{1}{|\mathcal{B}_k|K} \sum_{(u,r,v,t)\in\mathcal{B}_k} \sum_{j=1}^{K} \operatorname{softplus}\big(s(u, r, v_j^-, t) - s(u, r, v, t)\big). \tag{32}$$

The implementation can use sampled-softmax cross-entropy, but that option is disabled in the reported runs. Validation/test ranking uses the official negative sets queried by source, destination, original event timestamp, relation ID, and split. Link training uses a separate optimizer step per next-batch loss and detaches state afterward; persistent state is retained with a one-update gradient truncation.

For node-property tasks the head is an affine projection from the spatial representation to all output classes. If $y_{u,t}$ is the nonnegative target vector and $\hat{y}_{u,t} = y_{u,t}/\max(\sum_j y_{u,t,j}, 10^{-12})$, training averages $-\sum_j \hat{y}_{u,t,j} \log p_{u,t,j}$ over labeled nodes, then over consumed label timestamps. Evaluation averages NDCG over label timestamps without weighting by the number of labeled nodes.

Householder restriction maps. For each ordered endpoint pair, let $a_{u\to v} = \tanh(W_\Phi[h_u; h_v]) \in \mathbb{R}^{d_s(d_s-1)/2}$; W_Φ is bias-free and shared. Place these parameters in the strict lower triangle of T, following row-major triangular index order, and set its diagonal to one and upper triangle to zero. With column t_j of T, the implemented Householder product is

$$R_{e\leftarrow u} = \prod_{j=1}^{d_s} \left(I_{d_s} - 2\frac{t_j t_j^\top}{t_j^\top t_j} \right). \tag{33}$$

This is evaluated by `torch_householder_orgqr`; the reverse ordered pair supplies $R_{e\leftarrow v}$. All reported runs use this parameterization, tanh, and no extra fixed low/high-pass stalk dimensions. Zero parameters produce $-I_{d_s}$ at both endpoints, yielding identity transport $R_{e\leftarrow u}^\top R_{e\leftarrow v} = I_{d_s}$ with the same augmented normalization in the identity-map control.

A.6 Incidence maps and node-frame geometry

At a fixed time define orthogonal transport $Q_{uv} = R_{\{u,v\}\leftarrow u}^\top R_{\{u,v\}\leftarrow v}$. A node-frame parameterization $Q_{uv} = U_u^\top U_v$, as in TSNN propagation (Asif et al., 2026), has identity product around every cycle by cancellation of adjacent $U_v U_v^\top$. Incidence maps need not cancel, because a node's restriction can depend on its neighbor. This distinction concerns the propagation operator, not the expressivity of an entire nonlinear recurrent predictor or its relation-conditioned head.

For a concrete member of the evaluated decoder family, use $d_s = 2$ and a scalar coordinate ξ_u of the memory readout, with $\xi_0 = -a, \xi_1 = 0, \xi_2 = a$ for $a > 0$. Choose the shared decoder $p_{uv} = \tanh(\xi_u + 2\xi_v)$, with all other coordinate weights zero. Equation (33) gives

$$R(p) = \frac{1}{1+p^2} \begin{pmatrix} p^2-1 & 2p \\ -2p & p^2-1 \end{pmatrix} = \operatorname{Rot}(\pi + 2\arctan p).$$

Consequently $Q_{01}Q_{12}Q_{20} = \operatorname{Rot}(\Theta)$, where $\Theta = 8\arctan(\tanh a) - 4\arctan(\tanh 2a)$. To see that this product is nonidentity, let $b = \tanh a \in (0, 1)$ and $c = \tanh 2a = 2b/(1 + b^2)$. Since $\tan(2\arctan b) = 2b/(1 - b^2) > c > b$, we obtain $0 < 2\arctan b - \arctan c < \pi/4$, hence

$0 < \Theta < \pi$. Degree-normalization scalars are excluded from these transport products. On a tree there is no such cycle obstruction: frames can be assigned recursively from a root.

This example establishes structural freedom, not its use or predictive benefit in trained models. Nontrivial cycle transport can reduce the space of globally consistent signals and need not help a task. The node-frame control in Table 7 uses the same memory and head and shows no uniform TSD gain across Wiki and Forum. Learned cycle products were not measured.

A.7 Mathematical scope of the mechanism

The observations below identify structural properties retained by the evaluated implementation. Appendix B proves the continuous, discrete, and sheaf-regularized HiPPO constructions; Appendix C explains how the implementation allocates their temporal and geometric roles to compact learned modules.

Sheaf geometry and energy. For fixed restriction maps, $\widetilde{L}_k = (\delta_{\mathcal{F}_k} S)^\top (\delta_{\mathcal{F}_k} S)$ is positive semidefinite. Along the linear heat flow $dZ/ds = -\widetilde{L}_k Z$,

$$\frac{d}{ds}\mathcal{E}_{\mathcal{F}_k}(Z) = -2\,\mathrm{tr}(Z^\top \widetilde{L}_k^2 Z) = -2\|\widetilde{L}_k Z\|_F^2 \le 0. \tag{34}$$

Zero energy means that the degree-scaled node values satisfy the edge consistency constraints. These facts explain the geometry underlying sheaf diffusion. The learned residual field in Eq. (15) includes mixing and nonlinearities and is not this heat equation, so neither monotone energy descent nor prevention of oversmoothing follows automatically.

Linear generator versus coupled-system stability. Equation (18) has strictly negative diagonal entries and is triangular, so A is Hurwitz. For any fixed positive step δ, all eigenvalues of $\exp(\delta A)$ lie strictly inside the unit circle. This does not imply contraction in Euclidean norm for a non-normal generator, nor does it bound the input-dependent forcing, spatial feedback, or nonlinear coupled trajectory. The step cap is a numerical safeguard, not a stability theorem for the full model.

Ordered causal composition. The memory update produces node-indexed readouts, the decoder turns endpoint readouts into incidence maps, and those maps induce an operator on spatial cochains. The output cochains have the nodewise type required by the next feedback readout. This defines a well-typed, ordered recurrence on an observed graph sequence. Arbitrary node/edge changes do not themselves provide canonical maps between sheaves at different times; the implementation carries states by node identity and constructs the current sheaf anew. Because that sheaf depends on the updated memory, reversing temporal and spatial computation generally changes the output. The finite-step ordering is analyzed in Appendix B.2, and Appendix C explains the practical memory and propagation choices.

Sparse computation and empirical attribution. With fixed widths and layer count, sparse propagation scales with the local graph size, while core state storage scales with the number of tracked nodes. REC separately stores growing tables of distinct historical keys; neither fixed-width core memory nor local propagation implies stream-length-independent total storage. The full SSM matrix exponential adds a nontrivial width-dependent cost per updated node; the implementation chunks these computations and uses tiled candidate scoring. Measured costs are reported separately in the experimental appendix. The architecture permits history-dependent transport, but this capability alone does not establish a predictive advantage from long-range memory or history-conditioned maps. Those are empirical questions addressed by the controls in the main experiments.

B Continuous dynamics, discretization, and sheaf-based HiPPO

This appendix develops the continuous temporal sheaf system, a consistent discretization, and exact sheaf-regularized polynomial memory. These constructions analyze temporal and geometric design principles; the standard well-posedness and consistency results delimit their mathematical scope rather than establish predictive superiority. Appendix C explains how fixed-width learned memory, selective update clocks, and finite spatial computation turn that structure into the evaluated predictor;

Appendix A gives its specification. We state the conditions for each exact construction and identify the structural guarantees retained by the practical realization.

B.1 A WELL-POSED COUPLED CONTINUOUS MODEL

On an interval with fixed finite graph $G = (V, E)$, all node and edge stalks have fixed dimension d_s. Endpoint maps depend locally Lipschitz-continuously on memory readouts and time. Write $\delta(t, m)$ for the resulting coboundary, $L = \delta^\top \delta$, and $D = \mathrm{blockdiag}(L)$ for its block degree. With a fixed $\varepsilon > 0$, define

$$S = (D + \varepsilon I)^{-1/2}, \qquad \mathsf{L}(t, m) = SLS. \tag{35}$$

Here and below $\|\cdot\|_2$ is the Euclidean operator norm and $\|\cdot\|_F$ the Frobenius norm. The augmented normalization is defined even for rank-deficient restrictions or isolated nodes.

Proposition B.1 (Uniform bound for a varying sheaf)**.** *For arbitrary finite restriction maps on a loop-free graph,*

$$0 \preceq \mathsf{L} \preceq 2I, \qquad \|I - \mathsf{L}\|_2 \le 1. \tag{36}$$

The bound is independent of the memory values and the magnitudes of the restrictions.

Proof. For any cochain v, summing $\|R_{e\leftarrow u}v_u - R_{e\leftarrow v}v_v\|^2 \le 2\|R_{e\leftarrow u}v_u\|^2 + 2\|R_{e\leftarrow v}v_v\|^2$ over edges gives $0 \preceq L \preceq 2D$. Congruence by S gives $0 \preceq SLS \preceq 2SDS \preceq 2I$, since $SDS = D(D + \varepsilon I)^{-1} \preceq I$. The norm bound follows from the spectrum. □

Let $m_u \in \mathbb{R}^{d_h}$ and $Z \in \mathbb{R}^{nd_s \times f}$. Consider the following continuous realization, with temporal and spatial feedback on the same physical clock:

$$\begin{aligned} q_u(t, Z) &= [x_u(t); \mathcal{R}_u(t, Z); \psi_u(t)], \\ \dot{m}_u &= r(q_u)\{Am_u + B(q_u)q_u\}, \\ \dot{Z} &= f(t, m, Z) = \sigma[(I - \mathsf{L}(t, m))(I_n \otimes W_1)ZW_2]\,. \end{aligned} \tag{37}$$

The sheaf is decoded from $h_u = \rho(m_u)$. Assume that x_u, ψ_u are bounded and continuous on each compact time interval; $\mathcal{R}_u(t, Z)$ is affine in Z with bounded continuous coefficients; ρ, the decoder, $r \ge 0$, and B are locally Lipschitz in their state arguments, uniformly on compact time intervals, and continuous in time. The matrices A, W_1, W_2 are fixed and finite. The entrywise activation is globally Lipschitz with constant L_σ and $\sigma(0) = 0$. In particular, q_u has no direct dependence on m: temporal memory acts on the spatial field through the normalized sheaf. These assumptions include bounded exogenous graph descriptors and linear local spatial feedback.

Theorem B.2 (Global forward existence on finite horizons)**.** *Under these assumptions, every finite initial condition at t_0 determines a unique solution of Eq.* (37) *on every finite interval $[t_0, T]$. With $\kappa = L_\sigma\|W_1\|_2\|W_2\|_2$,*

$$\|Z(t)\|_F \le e^{\kappa(t-t_0)}\|Z(t_0)\|_F. \tag{38}$$

There are finite constants a_T, b_T, determined by this spatial bound and the input/selector bounds, such that the stacked memory satisfies

$$\|m(t)\| \le e^{a_T(t-t_0)}\{\|m(t_0)\| + b_T(t - t_0)\}. \tag{39}$$

Proof. Regularization by εI makes the inverse square root smooth on the positive-definite matrices. Thus the vector field is continuous in time and locally Lipschitz in (m, Z), giving a unique local solution. Proposition B.1 gives $\|\dot{Z}\|_F \le \kappa\|Z\|_F$, so Gronwall's inequality proves Eq. (38) independently of m. Bounded affine readout and bounded exogenous inputs now confine every q_u to a compact set on $[t_0, T]$. Local boundedness of r and B gives $\|\dot{m}\| \le a_T\|m\| + b_T$, and another application of Gronwall gives Eq. (39). Neither state can escape to infinity in finite time, so the local solution extends to T. □

This is an existence and finite-horizon growth result, not asymptotic stability or monotone energy descent. It does not require A to be Hurwitz. If the selectors additionally depend on unbounded memory values, the argument requires new growth assumptions. The positive normalization offset is essential to the stated regularity at singular degrees.

Input injection and zero-state invariance. The spatial field in Eq. (37) is unforced: $f(t, m, 0) = 0$. Uniqueness therefore implies $Z(t_0) = 0 \Rightarrow Z(t) = 0$ between resets, even if observations change memory and geometry. A nonzero initial cochain supplies the values propagated by this construction. The evaluated model injects inputs and memory through $Z_k^{(0)} = P_z[x_k; h_k]$ at every observation; that reset is not generated by the unforced ODE. A continuously driven alternative is stated below.

Changing graphs. A finite number of prescribed graph changes on a finite horizon can be handled by solving Eq. (37) between changes and applying specified finite, locally Lipschitz state-transfer maps at each change. Copying persistent node states, deleting departed coordinates, and initializing new nodes are examples. Prescribed input jumps can be included in the same event partition with identity state transfer. Use the appropriate one-sided input values on each interval. Induction over the intervals gives a unique hybrid trajectory. Node identity supplies these transfers; arbitrary graph changes alone do not supply canonical sheaf morphisms. Infinitely accumulating events and state-dependent event times are outside this statement.

B.2 Consistent discretization and update order

For a physical-time mesh t_k with $h_k = t_{k+1} - t_k > 0$, freeze $q_{u,k} = q_u(t_k, Z_k)$ and $r_{u,k} = r(q_{u,k})$. An ordered scheme first solves this held-input memory subproblem, then takes a spatial Euler step under the updated memory:

$$\begin{aligned} m_{u,k+1} &= e^{h_k r_{u,k} A} m_{u,k} + \left(\int_0^{h_k r_{u,k}} e^{sA}\, ds \right) B(q_{u,k}) q_{u,k}, \\ Z_{k+1} &= Z_k + h_k f(t_k, m_{k+1}, Z_k). \end{aligned} \tag{40}$$

If A is invertible, the integral is $A^{-1}(e^{h_k r_{u,k} A} - I)$; the integral form also covers singular A and zero rates.

Proposition B.3 (First-order approximation, without order invariance)**.** *Assume the hypotheses of Theorem B.2. On each closed interval between prescribed events, assume the coefficient maps and vector fields admit one-sided extensions with finite joint time/state Lipschitz bounds on compact state neighborhoods, including both endpoints. On a fixed finite horizon, the scheme in Eq.* (40)*, initialized at the exact initial state, has global error $O(h_{\max})$ as $h_{\max} = \max_k h_k \to 0$. Across finitely many prescribed events, require the mesh to contain every event time, the specified transfer maps to be locally Lipschitz, and each step to use the right-sided data of its interval. Reversing the order of two consistent substeps gives the same limiting ODE, but need not give the same finite-step states.*

Proof. Write $y = (m, Z)$ and $F = (g, f)$ for the coupled field. On a compact neighborhood of the exact trajectory, the memory step expands as $m^+ = m + hg(t, m, Z) + O(h^2)$. Local Lipschitz continuity gives $f(t, m^+, Z) = f(t, m, Z) + O(h)$, hence the full step is $y^+ = y + hF(t, y) + O(h^2)$ uniformly there. The exact integral equation has the same expansion, so the local defect is $O(h^2)$. The numerical map is Lipschitz with constant $1 + Ch$ on this neighborhood. Consequently the error satisfies $e_{k+1} \le (1 + Ch_k)e_k + Ch_k^2$, and discrete Gronwall gives $\max_k e_k \le C_T h_{\max}$. A standard stopping argument on a larger compact neighborhood justifies the bounds for all sufficiently small meshes: the error bound prevents a first exit. Each prescribed reset multiplies the accumulated error by a finite local Lipschitz constant, so finitely many resets preserve the order. Reversing the stages preserves the first-order expansion, but not its higher-order terms. □

Noncommutation occurs even within normalized sheaf dynamics. Take a two-node, one-edge scalar sheaf with both restrictions equal to e^m, normalization offset one, identity activation, and unit mixing. On its antisymmetric spatial mode, $I - \mathsf{L}$ acts by $-\tanh(m)$. With shared temporal state $\dot{m} = 1$, the exact frozen subflows are $T_h(m, z) = (m + h, z)$ and $S_h(m, z) = (m, e^{-h \tanh(m)} z)$. Consequently,

$$S_h T_h(m, z) - T_h S_h(m, z) = \left(0,\, [e^{-h \tanh(m+h)} - e^{-h \tanh(m)}] z\right), \tag{41}$$

which is nonzero for $h > 0$ and $z \neq 0$. Thus temporal and spatial substeps generally produce different finite-step states when their order is reversed. Proposition B.3 establishes their common continuous limit through consistency and regularity.

Proposition B.4 (An input-driven continuous extension)**.** *On a fixed-graph interval $[a, b]$ between prescribed events, assume the hypotheses of Theorem B.2 and let the exogenous input $U(t) \in \mathbb{R}^{nd_s \times f}$ admit a bounded continuous one-sided extension to the whole interval. Replacing the spatial equation by $\dot{Z} = f(t, m, Z) + U(t)$ preserves unique forward existence throughout $[a, b]$, with, for $a \le t_0 \le t \le b$,*

$$\|Z(t)\|_F \le e^{\kappa(t-t_0)}\|Z(t_0)\|_F + \int_{t_0}^{t} e^{\kappa(t-s)}\|U(s)\|_F \, ds. \tag{42}$$

Finite horizons with finitely many prescribed events follow by restarting this bound after each specified finite, locally Lipschitz state transfer. If the one-sided extension of U is also Lipschitz in time on each event interval, adding $h_k U(t_k)$ to the spatial increment in Eq. (40) *preserves the first-order conclusion of Proposition B.3, under its remaining assumptions.*

Proof. The inequality $\|\dot{Z}\|_F \le \kappa\|Z\|_F + \|U(t)\|_F$ and Gronwall give Eq. (42) independently of memory. This bounds q on each event interval, so the memory bound and continuation argument of Theorem B.2 still apply; finite state transfers allow induction over the intervals. Exogenous forcing does not change state-Lipschitz constants; its time regularity preserves the $O(h_k^2)$ local defect used in Proposition B.3. □

A node-local choice $U_u(t) = P_{\mathrm{in}} x_u(t)$ allows streaming inputs to create nonzero spatial values. This unevaluated continuous extension does not identify the model's finite resets with infinitesimal forcing or establish physical-time convergence for the benchmark predictor. Arbitrary unbounded memory-dependent forcing would require additional growth assumptions.

Computational-depth guarantees. Within a snapshot, hold $\widetilde{L}_k$, the memories, and W_1, W_2 fixed, and disable training dropout. Equation (15) is Euler integration of its stated autonomous field in computational time s, on a mesh with steps τ_ℓ. The field is globally Lipschitz with constant at most κ. For fixed total depth $S_* = \sum_\ell \tau_\ell$, its endpoint error tends to zero as $O(\max_\ell \tau_\ell)$ under mesh refinement, by the same local-defect argument. Moreover,

$$\|Z_k^{(L)}\|_F \le \prod_{\ell=0}^{L-1}(1 + \kappa\tau_\ell)\|Z_k^{(0)}\|_F \le e^{\kappa S_*}\|Z_k^{(0)}\|_F. \tag{43}$$

This conditional bound concerns an executed propagation block; the edge-free local event bypass returns its initializer. The learned model does not enforce $\delta_{u,k} = h_k r_{u,k}$ or $S_* = h_k$, may leave inactive memories unchanged, and resets spatial values from inputs and memories at snapshots. Its fixed number of learned steps is therefore not covered by the physical-time convergence assertion of Proposition B.3. The exact held-input memory formula in Eq. (20) remains valid on its selected effective clock.

B.3 Energy under a changing sheaf

For the separate linear heat flow $\dot{Z} = -\mathsf{L}(t)Z$ with a positive-semidefinite sheaf operator that is absolutely continuous between prescribed jumps, differentiation gives, almost everywhere on those intervals,

$$\begin{aligned} \frac{d}{dt}\|Z\|_F^2 &= -2\,\mathrm{tr}(Z^\top \mathsf{L} Z) \le 0, \\ \frac{d}{dt}\,\mathrm{tr}(Z^\top \mathsf{L} Z) &= \mathrm{tr}(Z^\top \dot{\mathsf{L}} Z) - 2\|\mathsf{L}Z\|_F^2. \end{aligned} \tag{44}$$

Here $\dot{\mathsf{L}}$ is the total derivative, including any dependence on evolving memory. Norm dissipation survives changing geometry, but instantaneous sheaf energy need not decrease. At an operator jump the energy changes by $\mathrm{tr}(Z^\top(\mathsf{L}^+ - \mathsf{L}^-)Z)$ if Z is continuous. The fixed-sheaf formula in Eq. (34) is recovered when $\dot{\mathsf{L}} = 0$.

For a frozen operator, the heat Euler step $(I - h\mathsf{L})Z$ is nonexpansive in norm and does not increase its fixed sheaf energy when $0 \le h\lambda_{\max}(\mathsf{L}) \le 2$; this follows by expanding in an orthonormal eigenbasis. The exact heat step $e^{-h\mathsf{L}}Z$ has both properties for every $h \ge 0$. These statements apply to heat flow and its specified discretizations, not to the learned residual field with $I - \mathsf{L}$, mixing, and activation.

B.4 EXACT HiPPO WITH HISTORICAL SHEAF REGULARIZATION

We now give a projection characterization conditional on a prescribed sheaf path. Fix an identified cochain space $\mathcal{V} \simeq \mathbb{R}^p$, $p = nd_s$, a bounded, piecewise-continuous signal $X : [0, T] \to \mathcal{V}$, and a bounded, piecewise-continuous positive-semidefinite sheaf operator $\mathcal{L}(s)$ on that space. Edge changes are allowed; changes of stalk dimension require an explicit identification into a common space before applying this result. The path may be obtained from observations, but is held fixed when minimizing over memory coefficients. This is not a well-posedness claim for an implicit closure $\mathcal{L} = \mathcal{L}(H)$.

Let $N \geq 1$, $\alpha \geq 0$, $Q(s) = I_p + \alpha\mathcal{L}(s)$, and $\pi_j^{(t)}(s) = \sqrt{2j+1}\, P_j(2s/t - 1)$ for $j = 0, \ldots, N-1$, where P_j is the Legendre polynomial. The vector π_t is orthonormal for $d\mu_t(s) = ds/t$ on $[0, t]$. A memory $H \in \mathbb{R}^{p\times N}$ reconstructs the history as $H\pi_t(s)$. The historical-sheaf objective is

$$J_t(H) = \frac{1}{t}\int_0^t \left\{\|X(s) - H\pi_t(s)\|^2 + \alpha\,(H\pi_t(s))^\top \mathcal{L}(s)(H\pi_t(s))\right\} ds. \tag{45}$$

The projection identities below hold for any bounded positive-semidefinite operator path on the identified space. The sheaf instantiation specifies that path through local transformed endpoint discrepancies. This gives relational meaning and graph support to the penalty, without proving an advantage over other regularizers or claiming that the evaluated recurrent state computes the projection.

This extends polynomial memory (Gu et al., 2020) and Laplacian-regularized graph memory (Li et al., 2024) to endpoint-dependent sheaf comparisons. Crucially, the geometry in this objective is evaluated at the historical time s, not replaced by its value at t.

Theorem B.5 (Historical sheaf projection and finite-state dynamics)**.** *Define the ordinary moments $C(t) \in \mathbb{R}^{p\times N}$ and the Gram state $K(t) \in \mathbb{R}^{pN\times pN}$ by*

$$C(t) = \frac{1}{t}\int_0^t X(s)\pi_t(s)^\top ds, \qquad K(t) = \frac{1}{t}\int_0^t [\pi_t(s)\pi_t(s)^\top] \otimes Q(s)\, ds. \tag{46}$$

With column-wise vectorization, the unique minimizer is

$$\operatorname{vec} H_\star(t) = K(t)^{-1} \operatorname{vec} C(t), \qquad K(t) \succeq I_{pN}. \tag{47}$$

Use the N-dimensional A_H, b_H of Eq. (21) *and set $R_H = A_H + \frac{1}{2}I_N$. In logarithmic time $\tau = \log(t/t_{\mathrm{ref}})$, $t_{\mathrm{ref}} > 0$, the exact states obey almost everywhere*

$$\begin{aligned} \frac{dC}{d\tau} &= CA_H^\top + X(t)b_H^\top, \\ \frac{dK}{d\tau} &= (R_H \otimes I_p)K + K(R_H^\top \otimes I_p) + (b_H b_H^\top) \otimes Q(t). \end{aligned} \tag{48}$$

Physical-time derivatives are $1/t$ times these right-hand sides. For a fixed sheaf path the coefficient extractor is nonexpansive:

$$\|H_\star[X] - H_\star[Y]\|_F \leq \|X - Y\|_{L^2(\mu_t)}. \tag{49}$$

Proof. Expanding Eq. (45) gives a constant plus $h^\top Kh - 2h^\top c$, where $h = \operatorname{vec} H$ and $c = \operatorname{vec} C$. Since $Q(s) \succeq I_p$ and the polynomials are orthonormal, $K \succeq I_{pN}$, proving strict convexity and Eq. (47).

The shifted Legendre basis satisfies $\pi_t(t) = b_H$ and $t\partial_t\pi_t(s) = (A_H + I_N)\pi_t(s)$. For completeness, differentiating $P_j(2s/t - 1)$ uses $(x+1)P_j'(x) = jP_j(x) + \sum_{i<j}(2i+1)P_i(x)$; the stated identity follows after normalization. Differentiating the integrals, the moving endpoint contributes $X(t)b_H^\top$ and $(b_H b_H^\top) \otimes Q(t)$, respectively. The derivative of $1/t$ subtracts C or K; the one or two basis derivatives produce the remaining terms in Eq. (48). The identities hold almost everywhere for the specified bounded inputs. Finally, $\|K^{-1}\|_2 \leq 1$ and Bessel's inequality for the ordinary moments give Eq. (49). □

The coefficient trajectory itself satisfies $\dot{h}_\star = K^{-1}(\dot{c} - \dot{K}h_\star)$ wherever these derivatives exist. A jump in historical $Q(t)$ or $X(t)$ does not jump C, K, or $H_\star$ at a positive time: changing an integrand at a single point has zero measure. Past geometries remain represented in K. The explicit Gram representation uses a potentially dense state of size $pN \times pN$; it is a theoretical online characterization, not a lower bound on storage or a claim of sparse cost for the model.

Proposition B.6 (Exact discrete updates for held histories)**.** *Suppose $X(t) = X_k$ and $Q(t) = Q_k$ on $[t_k, t_{k+1})$ with $t_k > 0$. Set*

$$\eta_k = \log(t_{k+1}/t_k), \quad E_k = e^{\eta_k A_H}, \quad J_k = e^{\eta_k R_H}, \quad v_k = A_H^{-1}(E_k - I_N)b_H. \tag{50}$$

The following recurrence exactly reproduces the moments and minimizer for this held-input history:

$$\begin{aligned} C_{k+1} &= C_k E_k^\top + X_k v_k^\top, \\ K_{k+1} &= (J_k \otimes I_p)K_k(J_k^\top \otimes I_p) + (I_N - J_k J_k^\top) \otimes Q_k, \\ \operatorname{vec} H_{\star,k+1} &= K_{k+1}^{-1} \operatorname{vec} C_{k+1}. \end{aligned} \tag{51}$$

For every $\eta_k > 0$, $K_k \succeq I$ implies $K_{k+1} \succeq I$.

Proof. The moment recurrence follows by variation of constants in log time. Direct inspection of the LegS entries gives

$$A_H + A_H^\top = -I_N - b_H b_H^\top, \qquad R_H + R_H^\top = -b_H b_H^\top. \tag{52}$$

Thus $\int_0^{\eta_k} e^{sR_H} b_H b_H^\top e^{sR_H^\top} ds = I_N - J_k J_k^\top \succeq 0$. Solving the Lyapunov equation for K gives the second recurrence. Its two terms are bounded below by $(J_k J_k^\top) \otimes I_p$ and $(I_N - J_k J_k^\top) \otimes I_p$, respectively, proving the lower bound. The minimizer then follows from Theorem B.5. □

For nonconstant histories, this recurrence is exact for the chosen piecewise-constant reconstruction, not for unobserved variation between samples. To represent all of $[0, t_k]$, its initial C_k, K_k must equal the integrals in Eq. (46). If X, Q are continuous at zero, their limits are $C(0^+) = X(0)e_0^\top$ and $K(0^+) = I_N \otimes Q(0)$, with $e_0 = (1, 0, \ldots, 0)^\top$. The logarithmic recurrence starts at a positive time using the corresponding history moments; setting arbitrary zero states there does not recover the full-history projection.

B.5 Storing operator moments instead of the full Gram matrix

The full Gram state is a transparent representation, but it contains redundant polynomial-product information. The following identity separates persistent history storage from the subsequent coefficient solve.

Proposition B.7 (Exact operator-moment representation)**.** *Under the assumptions of Theorem B.5, let $M = 2N - 1$ and extend the normalized shifted Legendre basis through degree $M - 1$. Define symmetric operator moments*

$$\mathcal{M}_r(t) = \frac{1}{t}\int_0^t Q(s)\pi_{t,r}(s)\, ds, \qquad \gamma_{ijr} = \int_0^1 \pi_{1,i}(u)\pi_{1,j}(u)\pi_{1,r}(u)\, du, \tag{53}$$

for $0 \le r < M$ and $0 \le i, j < N$. The (i, j) block of K is exactly

$$K_{ij}(t) = \sum_{r=0}^{M-1} \gamma_{ijr}\mathcal{M}_r(t). \tag{54}$$

Let $A_H^{(M)}, b_H^{(M)}$ be the LegS generator and input vector of order M. In log time,

$$\frac{d\mathcal{M}_r}{d\tau} = \sum_{j=0}^{M-1} (A_H^{(M)})_{rj}\mathcal{M}_j + (b_H^{(M)})_r Q(t). \tag{55}$$

If $Q = Q_k$ is held on $[t_k, t_{k+1})$, set $E_k^{(M)} = e^{\eta_k A_H^{(M)}}$ and $v_k^{(M)} = (A_H^{(M)})^{-1}(E_k^{(M)} - I_M)b_H^{(M)}$. Then

$$\mathcal{M}_{r,k+1} = \sum_{j=0}^{M-1} (E_k^{(M)})_{rj}\mathcal{M}_{j,k} + (v_k^{(M)})_r Q_k. \tag{56}$$

With history-consistent initialization, these moments reconstruct exactly the Gram matrix of Theorem B.5, hence $K \succeq I$. On a first held interval $(0, t_0]$, initialize $\mathcal{M}_0(t_0) = Q_0$ and $\mathcal{M}_r(t_0) = 0$ for $r > 0$.

Proof. The product $\pi_{t,i}\pi_{t,j}$ has degree at most $2N-2$ and thus expands exactly in the first M orthonormal polynomials. Substitution $u = s/t$ makes its coefficients γ_{ijr} independent of t. Multiplication by $Q(s)$ and integration give Eq. (54). Each matrix entry of $\mathcal{M}_r$ is an ordinary LegS moment; applying its derivative identity gives Eq. (55), and variation of constants gives Eq. (56). Orthonormality gives the stated first-interval initialization. The integral representation is retained by these exact updates, so the reconstructed K has the original lower bound. □

This representation stores $2N-1$ symmetric $p \times p$ matrices, or $O(p^2N)$ persistent operator-history values, in addition to the $O(pN)$ signal moments. This count excludes any precomputed product/transition tables and solve workspace: a dense coefficient solve can still materialize K and incur its larger cost. Individual higher moments $\mathcal{M}_r$ need not be positive semidefinite; positivity belongs to the reconstructed Gram matrix for history-consistent states. At $N = 1$, the representation reduces to the single average $\mathcal{M}_0 = t^{-1}\int_0^t Q(s)\,ds$. No minimal-state or inexpensive-inverse claim follows.

B.6 Current geometry, initialization, and the learned model

A lower-cost alternative applies the *current* geometry to the entire represented history. Replace $\mathcal{L}(s)$ in Eq. (45) by $\mathcal{L}(t)$, held fixed during minimization. The penalty becomes $\alpha\,\mathrm{tr}(H^\top \mathcal{L}(t) H)$, giving a distinct, precisely specified objective.

Current graph filtering of polynomial memory and its time-varying correction also appear in CTDG-SSM (Raghuvanshi et al., 2026). Its reconstruction objective applies a graph filter to the represented signal; here the additive quadratic penalty yields the resolvent Q^{-1}. The derivative of the inverse filter is a shared identity, while placing $\mathcal{L}(s)$ inside the historical integral gives the distinct Gram construction above.

Proposition B.8 (Current-sheaf projection and its geometric correction)**.** *With $Q(t) = I_p + \alpha\mathcal{L}(t)$, the current-geometry minimizer is $H_\star(t) = Q(t)^{-1}C(t)$. If Q is locally absolutely continuous between prescribed jumps, then almost everywhere on those intervals*

$$\dot{H}_\star = \frac{1}{t}\{H_\star A_H^\top + Q^{-1}X(t)b_H^\top\} - Q^{-1}\dot{Q}H_\star. \tag{57}$$

At a jump of Q at $t > 0$,

$$H_\star(t^+) = Q(t^+)^{-1}Q(t^-)H_\star(t^-). \tag{58}$$

For a held signal on a positive-time interval, an exact discrete update is

$$H_{\star,k+1} = Q_{k+1}^{-1}\{Q_k H_{\star,k}E_k^\top + X_k v_k^\top\}, \tag{59}$$

where E_k, v_k are defined in Eq. (50) *and Q_k, Q_{k+1} are the selected endpoint geometries. The intervening path of the current geometry does not enter this endpoint characterization.*

Proof. Orthonormality makes the normal equation $QH_\star = C$. Differentiate it and substitute $\dot{C} = (CA_H^\top + Xb_H^\top)/t$ to obtain Eq. (57). Continuity of C gives the jump condition, and its exact held-signal recurrence gives Eq. (59). □

For a fixed sheaf, Q is constant, both objectives coincide, and the correction vanishes. Spatial filtering and the homogeneous temporal transition then commute on the coefficient matrix: $Q^{-1}(CE_k^\top) = (Q^{-1}C)E_k^\top$. This separability concerns fixed linear operators; it does not make the learned feedback updates commute. Omitting $\dot{Q}$ for a smoothly changing current sheaf, or replacing the historical Gram state by $I_N \otimes Q(t)$ for changing historical geometry, changes the projection being computed.

Historical geometry cannot generally be replaced by a current filter. Consider the unit antisymmetric mode v of a two-node sheaf with both restrictions $\sqrt{s/2}$ on $[0, 1]$, choosing the unnormalized path $\mathcal{L}(s) = L_{\mathcal{F}(s)}$. Its eigenvalue on that mode is s. Let the scalar signal on that mode be $X_v(s) = 1$, $\alpha = 1$, and $N = 2$. Direct integration gives the moments and Gram block restricted to this mode:

$$C_v(t) = (1, 0), \qquad K_v(t) = \begin{pmatrix} 1 + t/2 & \sqrt{3}\, t/6 \\ \sqrt{3}\, t/6 & 1 + t/2 \end{pmatrix}.$$

At $t = 1$, historical coefficients are $(9/13, -\sqrt{3}/13)$, while current-geometry filtering gives $(1/2, 0)$. The nonzero degree-one coefficient records the history of the penalty even though the observed signal is constant. This exact example shows why changing historical geometry couples polynomial degrees through K; it is not an experiment on the learned recurrence.

Sheaf-based HiPPO initialization. The fixed/current-sheaf characterization supplies a principled initialization on cochains: initialize temporal modes with A_H, b_H, use the sheaf filter Q^{-1} on the signal, and initialize coefficient state from $Q(t_0)^{-1}C(t_0)$. For historical regularization, initialize C, K from the observed history instead. The filter acts in the sheaf stalk space, including heterogeneous endpoint comparisons. On a fixed-sheaf eigenmode of eigenvalue $\lambda \geq 0$ it multiplies coefficients by $(1 + \alpha\lambda)^{-1}$, preserving $\ker \mathcal{L}$ exactly. It does not promise retention of every predictive signal or prevent oversmoothing as regularization grows.

Equation (52) further proves $\|e^{sA_H}\|_2 \leq e^{-s/2}$ for $s \geq 0$: differentiate the squared norm of a homogeneous trajectory and apply Gronwall. This Euclidean contraction is a property of the exact initialized generator. Arbitrary trained triangular generators with negative diagonal remain Hurwitz, as stated in Appendix A.7, but need not retain this norm bound.

The model uses this temporal structure to initialize a compact learned memory: A starts at A_H, and the input-selector bias is $b_H \mathbf{1}^\top / \sqrt{d_q}$ with small random selector weights. The bias initially injects the scalar aggregate $\mathbf{1}^\top q / \sqrt{d_q}$ into shared temporal modes. Prediction training then adapts the generator, content-dependent injection, and effective clock; separate sheaf propagation supplies relational processing and subsequent feedback. Appendix C.1 motivates this allocation of roles and its storage benefit. The exact historical Gram state, current-sheaf filter, canonical clock, and coefficient initialization remain requirements of the corresponding projection constructions. Theorem B.5 and Propositions B.6–B.8 justify those constructions; the evaluated recurrence uses their structured initialization without asserting exact projection optimality.

C Practical realization of temporal sheaf dynamics

The model allocates temporal compression, geometric comparison, and spatial computation to separate trainable modules. The choices below reduce state size and computation. They define a learned representation whose relationship to the exact constructions in Appendix B is architectural; no approximation bound is claimed.

C.1 Budgeted memory with a structured initialization

Exact historical regularization can store $2N - 1$ operator moments instead of the full Gram matrix (Proposition B.7). Their persistent state costs $O(p^2 N)$ in the general dense case, where $p = nd_s$; coefficient recovery still involves a coupled pN-dimensional system, with separate solve costs. The model allocates $O(n(d_h + d_s f))$ temporal and spatial core values to a fixed tracked-node set and supplies graph processing through sparse propagation and feedback. This is a nodewise learned representation rather than a derived approximation to the exact solver. S4 motivates initializing trainable state-space parameters from HiPPO; S5 motivates a shared multi-input state (Gu et al., 2020; 2021; Smith et al., 2022). Our shared input bias $b_H \mathbf{1}^\top / \sqrt{d_q}$ has operator norm $\|b_H\|_2$, independent of input width. This controls its initial gain without allocating independent optimal polynomial coefficients to every input channel. GraphSSM similarly motivates combining temporal compression with learned graph processing (Li et al., 2024). The exact Gram projection and current-geometry resolvent additionally require their prescribed state, clock, and initialization.

C.2 Selective clocks and exact local transitions

Input-dependent steps and input maps follow selective SSMs (Gu & Dao, 2023). In the scalar held-input system $\dot{m} = -am + v$, $a > 0$,

$$m^+ = e^{-a\delta} m + (1 - e^{-a\delta}) v/a. \tag{60}$$

The effective step controls retention relative to selected forcing. Our matrix recurrence uses a full triangular generator and the exact held-input transition, with a capped, batch-gap-conditioned step. It inherits neither a diagonal SSM's fused-scan cost nor a full-system stability guarantee. The global gap describes the observation stream, not each node's inactivity; it can change when unrelated batches are inserted. On constant-gap streams its contribution can be absorbed into a bias. Appendix E.5 compares node-specific clocks and measures cap saturation; Appendix D.1 identifies the explicit timing distinctions a saturated selector removes. Physical-time approximation instead requires the clock relation and state evolution in Proposition B.3.

C.3 Observation-driven scheduling and cached feedback

Following the separation of event memories and graph embeddings in TGN (Rossi et al., 2020), m persists while $P_z[x; h]$ injects current values into each spatial computation. Cached feedback makes the next memory update explicit, without a within-snapshot fixed-point solve. Link updates use active nodes and a hop-limited closure; node-property propagation covers the full snapshot. Hop limits do not bound neighborhood size on arbitrary graphs. Holding inactive states avoids continuous integration between observations, an alternative studied in ODE-RNN (Rubanova et al., 2019); it also prevents the backbone from modeling unobserved evolution on its own. Query-time dependence is supplied by REC as specified in Appendix A.5. Training-prefix context and node-label schedules remain separate information-availability choices (Appendix E.1).

C.4 Reusable geometry and finite learned propagation

Maps are decoded once and reused within a block, a computation-saving pattern also used by fixed-coefficient neural diffusion (Chamberlain et al., 2021). Orthogonal maps make the degree blocks scalar and augmented normalization inexpensive, following Neural Sheaf Diffusion (Bodnar et al., 2022). They retain positive-semidefinite geometry and the normalized spectral bound. The residual update performs learned transport with mixing and activation; its linear special case includes a reaction term. The applicable guarantee is the conditional computational-depth bound in Eq. (43), rather than heat-energy descent. Reusing geometry, orthogonality, and finite depth constrain the architecture; none has an isolated predictive advantage established by the present experiments.

C.5 Cost and empirical scope

Fixed-width core state does not make the entire predictor constant-memory or inexpensive. The affine B selector has $d_h d_q (d_q + 1)$ parameters and costs $O(d_h d_q^2)$ per updated node; the dense matrix exponential adds width-dependent cost, typically cubic in d_h. REC's key/count/time tables grow with distinct observed keys, while candidate scoring and negative sets add further cost. Appendix E.2 reports parameter counts and measured stage times. The same-head comparisons in Section 6 show that smaller recurrent alternatives can match or outperform the model. The larger allocation has no established predictive return in these comparisons. Appendix D specifies which motivating temporal properties these feasibility choices can forgo and which structural guarantees remain.

D Which temporal properties survive practical realization?

The continuous system and exact projection constructions in Appendix B describe mathematical capabilities under explicit assumptions. Their validity does not imply that every finite-state predictor using their components inherits those capabilities. The choices in Appendix C reduce computation and storage, but some also discard temporal distinctions or learning paths. We identify these losses below. They provide mechanisms through which practical realization can limit the motivating dynamics; attributing a particular benchmark gap to them requires a controlled intervention.

D.1 A CAPPED CLOCK CAN ERASE PHYSICAL-TIME DISTINCTIONS

Fix the incoming memory m^-, recurrent input q, and model parameters. Write $v(q, \Delta)$ for the uncapped positive selector in Eq. (19), $\delta(q, \Delta) = \min\{\delta_{\max}, v(q, \Delta)\}$, and

$$F(m^-, q, \Delta) = e^{\delta(q,\Delta)A} m^- + A^{-1}\big(e^{\delta(q,\Delta)A} - I\big)B(q)q. \tag{61}$$

Here A is the invertible generator of Eq. (18), and $\delta_{\max}$ is a fixed cap.

Proposition D.1 (Loss of the explicit gap channel under saturation)**.** *Suppose $v(q, \Delta) > \delta_{\max}$ throughout an open interval I of physical gaps. Then $F(m^-, q, \Delta)$ is constant on I, so $\partial_\Delta F = 0$. At a strictly saturated, differentiable parameter point, any parameter whose only path into F is through this step selector also has zero derivative through that path.*

Proof. On I the minimum selects the same constant $\delta_{\max}$. All remaining terms of Eq. (61) are fixed. Strict saturation persists in a parameter neighborhood by continuity, giving the derivative statement there. At the boundary $v = \delta_{\max}$ the minimum need not be differentiable, and that case is excluded. □

This is a conditional loss of the *explicit gap channel*, not a statement that the whole model ignores time: stored memory, selected forcing $B(q)q$, graph descriptors, and REC can still convey temporal information. For a readout restricted to this encoded state, it can cause a representational loss. Consider the scalar physical system $\dot{m} = -m$, $m(0) = 1$, and the implemented subproblem $A = -1$, $m^- = 1$, $B(q)q = 0$. For distinct positive gaps a, b that both saturate a selector, the implemented transition returns the same value $c = e^{-\delta_{\max}}$; the physical endpoints are e^{-a} and e^{-b}. For every deterministic readout g whose only varying-case information is this encoded state,

$$\max\{|g(c) - e^{-a}|, |g(c) - e^{-b}|\} \;\geq\; \tfrac{1}{2}|e^{-a} - e^{-b}| > 0. \tag{62}$$

The triangle inequality proves the bound. Providing the physical gap separately to the readout would remove this particular indistinguishability. The example establishes possible damage to physical-time representation, without assuming that these endpoints are the targets of a benchmark.

The global batch gap introduces a separate identification problem: two updates with the same (m^-, q, Δ) produce identical outputs even when the node's last activity occurred at different times. A node-specific clock can distinguish these cases explicitly. Stored memory may already distinguish some histories, so this observation does not assert that global clocks eliminate all node-history information.

D.2 EXACT LOCAL TRANSITIONS AND PHYSICAL-TIME CONSISTENCY

Zero-order hold is exact for the selected linear system with held input. The scalar example isolates the choice of clock: an exact transition on an effective clock need not approximate evolution over the physical interval. Away from prescribed resets, a consistent one-step approximation to a fixed finite continuous vector field $\dot{y} = \mathcal{V}(t, y)$ must satisfy

$$\Phi_h(t, y) = y + h\mathcal{V}(t, y) + o(h), \qquad h \downarrow 0. \tag{63}$$

For example, if the selected step remains $\delta_{\max} > 0$ as the physical increment h tends to zero, the scalar zero-input update tends to $e^{-\delta_{\max}} m$ rather than m for $m \neq 0$. It therefore fails even the necessary identity limit for a physical-time integrator. The implemented selector does not enforce vanishing steps at zero physical gap, although a particular learned trajectory need not be saturated there.

Proposition B.3 avoids this problem by coupling the effective memory step to h and evolving the spatial state with physical-time increments. Finite resets $P_z[x; h_u]$, inactive-state holding, and propagation lengths independent of elapsed time define a different recurrence. They do not satisfy that proposition's hypotheses merely because the component update formulas use exponentials or Euler steps. Frozen coefficients and ordered splitting have controlled first-order error under the proposition's refinement assumptions; without those assumptions, reducing batch width alone is not a convergence guarantee. Thus some compromises change the represented dynamics, beyond ordinary finite-step numerical error.

D.3 Observation timing and temporal credit assignment

Batch forecasting deliberately scores every event from the state before the batch is incorporated. Fix that state, the complete pre-batch REC cache (including pending observations from earlier ingested batches), and all query and head inputs. Two streams that differ only in earlier-than-query events within the current batch receive the same prediction for that query: those events have not yet entered either stored state or REC. An event-by-event predictor could use them. This is an information-availability difference, not a solver defect or leakage claim. Batch-dependent forecasting protocols and their consequences are studied by Lampert et al. (2026).

Training introduces another distinct compromise. In a one-step truncated recurrence, write $s_j = F_\theta(\mathrm{sg}(s_{j-1}), x_j)$, where sg preserves the forward value and stops its derivative. For comparison, the untruncated derivative would contain

$$\frac{ds_j}{d\theta} = \partial_\theta F_\theta(s_{j-1}, x_j) + \partial_s F_\theta(s_{j-1}, x_j)\frac{ds_{j-1}}{d\theta}. \tag{64}$$

Detachment omits the second term. A parameter occurrence whose only path to a later loss crosses a detach boundary receives no gradient from that loss along this path. Shared parameters can still receive gradients at later occurrences, and the forward state can retain old information: a short training horizon is not a one-step memory horizon. Nevertheless, retaining a delayed cue and learning how to retain it are different requirements. Prior work documents this truncation issue in recurrent temporal graph models (Bravo et al., 2024); the present experiments do not identify its effect on TSD's observed scores.

D.4 Structured initialization and guarantees that remain

The historical-sheaf projection retains moments of both the signal and past geometry. The current-sheaf construction instead filters the ordinary polynomial coefficients by the present operator and includes its geometric correction. These are exact, different objectives, not interchangeable accuracy guarantees. A compact learned node memory does not inherit those identities merely from HiPPO initialization: the sufficient statistics, canonical clock, and coefficient initialization are also part of the exact construction. Substituting a trainable fixed-width representation can therefore forgo the projection property that motivated the initialization without making the projection theorem false.

The normalized sheaf remains positive semidefinite with its spectral bound; the held-input memory transition remains exact on its selected clock; and the conditional propagation-block bound still applies. Neither exact projection nor physical-time consistency alone guarantees superior forecasting risk. Restoring one such property may improve, leave unchanged, or worsen predictive accuracy under a finite data and optimization budget.

D.5 What the empirical audit does and does not establish

The matched controls and targeted diagnostics are reported in Appendix E.5. A separate Wiki global-clock checkpoint exhibits fully capped test updates, demonstrating that the saturation mechanism can occur in the trained system. That checkpoint is distinct from the five primary matched runs, whose cap rates were not recorded. Node-specific-clock results show modest gains, but saturation varies sharply by seed. Shortening Wiki batches from 600 to 300 seconds improves mean MRR from 0.7623 to 0.7810 on five seeds while doubling optimizer updates at the same epoch count. This intervention changes the observation frontier, update frequency, graph construction, and optimization budget together, so its gain cannot be attributed to discretization error alone. Delayed-cue tasks do not establish the intended memory advantage, and the experiments include no matched restoration of the gradient horizon to test the truncation explanation.

These choices can remove temporal distinctions and learning paths, and the diagnostics find some of these restrictions in practice. The negative component comparisons neither refute the conditional theory nor demonstrate that undoing the compromises would yield a sheaf-specific predictive advantage. Establishing the latter requires restoring an identified property while holding the prediction task, head, optimization budget, and relevant controls fixed.

E ADDITIONAL EXPERIMENTAL DETAILS

E.1 EVALUATION PROTOCOL AND IMPLEMENTATION

Model backbone and inputs. Each node retains an SSM state across snapshots. The lower-triangular generator is initialized from HiPPO-LegS and has a negative diagonal by construction. The exact zero-order-hold update uses a content-dependent step with a physical-gap input, normalized by the median positive training snapshot gap. The default clock is the gap between processed snapshots. Orthogonal restriction maps are decoded once per snapshot, and residual Euler steps use tied stalk and feature transformations. Their learned step lengths do not sum to physical elapsed time. Appendix A gives the exact implementation.

REL denotes incident-relation embeddings aggregated into the recurrent input; TYPE denotes learned node-type inputs. REC adds a score from past-interaction indicators, counts, and recency. Knowledge and heterogeneous graph configurations use typed (s, r, o) and untyped (s, o) channels; Smallpedia and Wikidata additionally use an unordered-pair channel. Wiki has one relation and uses pair recurrence. Node-property experiments use their node prediction head without REC.

Chronology and scoring. For link tasks, a snapshot's edges and active set do not construct the representations from which that snapshot is scored. Training computes the preceding output with gradients attached, scores the next snapshot, and detaches history according to the BPTT schedule. Final evaluation rebuilds state from the retained training stream, processes validation, and continues into test without parameter updates. The fixed context graph uses the first 50k retained training edges. It contains no validation/test information, but early training snapshots can access later edges within that prefix; training is therefore not an exclusively online edge-arrival procedure.

Official per-query negative sets are evaluated in full. Variable-length lists are padded and masked; MRR is averaged over queries. Dense candidate scoring and chunking preserve candidate sets. The evaluator guards non-finite scores, which could otherwise affect ranks. We therefore inspect raw positive and negative logits before guards and padding masks, record intended masks separately, and compare raw, guarded, and conservative reciprocal ranks.

Numerical audit. The audit contains 22 retained-checkpoint replay runs: three seeds on each of seven link datasets plus a second Smallpedia seed-47 replay. Across 72,517,031 validation/test query checks, it records zero affected queries, zero non-finite state/cache snapshots, and identical raw, guarded, and conservative metrics within each replay. This rules out score-replacement inflation in these audited evaluations. It does not imply bitwise checkpoint reproducibility: Smallpedia seed 47 has retained MRR 0.611180 and replay MRRs 0.610411/0.610309; ICEWS seed 46 differs by 0.000062. CUDA accumulation nondeterminism is a possible explanation, not a uniquely established cause.

Across 219 logged evaluations, including replays and smoke runs, 741,853,731 query checks likewise have zero affected queries. These are repeated checks, not distinct data examples. The 197 runs with nonzero recorded training steps report 2,249,742 steps and zero non-finite-loss, non-finite-gradient, or skipped-step events. Finite-gradient clipping remains active. Training counters are absent for 79 reused records, so historical training skip rates remain unknown. Checkpoints require finite selection loss and MRR.

Node-label trace. We traced our runner and the reference TGN label-consumption schedule using py-tgb 2.2.0. Our runner seeks the first label not earlier than a split's first edge, ingests a whole snapshot, then scores outstanding labels up to its last event. Trade uses yearly snapshots. Across all 30 label years and 6,543 node-years, the stored year-Y target matches normalized year-Y edges within 2×10^{-8}. Both schedules have already observed those edges when predicting that target; copying available edges gives NDCG@10 of 1. Their scored label sets nevertheless differ: our validation/test years are 2010–2013/2014–2016, versus 2009–2012/2013–2015 in the reference schedule, which does not consume the final 2016 label. The observed score is thus not evidence of next-year forecasting.

Genre uses 24-hour buckets anchored to each split's first event. The two traced schedules score the same 1,250/163/166 training/validation/test labels, but ingest different event prefixes at some label times. Their median overshoot past the label timestamp is approximately one hour versus

0.2 hours. A seven-day future window is the closest tested reconstruction of the targets, but still has substantial error; exact target construction is not established by this comparison. Neither trace establishes protocol equivalence with every leaderboard submission. We therefore retain node scores descriptively and withhold their external ranks.

E.2 Configurations, selection, and compute

Table 3: Headline link-task budgets. Training-event counts include inverse events supplied by TGB. Exact means one snapshot per distinct timestamp. Epochs are maxima; selection uses the stated validation prefix and final evaluation uses complete splits.

Dataset	Snapshot	Train events	Max epochs	Val. prefix
`tgbl-wiki`	600 s	110,232	8	8,000
`tkgl-smallpedia`	Exact (year)	775,514	40	50,000
`tkgl-polecat`	Exact (day)	2,493,112	10	30,000
`tkgl-icews`	Exact (day)	2,000,000	6	30,000
`tkgl-wikidata`	Exact (year)	2,000,000	4	300,000
`thgl-software`	3,600 s	1,042,866	20	20,000
`thgl-forum`	600 s	4,000,000	4	20,000

All headline link configurations use stalk dimension 2, 8 feature channels, two Euler steps, one closure hop, 32 training negatives per positive, and normalized memory readout. Memory width is 64 except on Wikidata (32); feedback width is 16. One-step BPTT preserves the gradient from a scored snapshot to its preceding update. Adam learning rates are 10^{-3} for Wiki, Software, and Forum, 10^{-2} for Smallpedia, and 3×10^{-3} for POLECAT, ICEWS, and Wikidata. Wiki selects each arm's learning rate using seed-43 tracking validation on the grid $\{3 \times 10^{-4}, 10^{-3}\}$, then locks it for seeds 44–47; all arms select 10^{-3}. Its dedicated training-negative generator is seeded by (seed, epoch), so architecture-dependent random draws cannot change the sampled-negative stream at fixed batching. Other datasets retain their task configuration and earlier RNG regime.

Headline Wiki, ICEWS, and Forum use seeds 43–47; the other link rows use 43/46/47. Smallpedia's component comparison uses a separate 15-epoch budget, not the headline 40-epoch budget. Its TSD reference is 0.6106 ± 0.0040, versus headline 0.6103 ± 0.0067. For training suffixes, omitted earlier interactions do not warm REC.

Both node configurations use stalk dimension 4, 64 feature channels, 64-dimensional memory, 32-dimensional feedback, three Euler steps, two closure hops, BPTT length 8, and normalized memory readout. Trade uses Adam with learning rate 0.01248794, at most 200 epochs, validation every two epochs, patience 60, and a minimum of 80 epochs. Genre uses learning rate 0.005, at most 40 epochs, validation every two epochs, patience 12, and a minimum of 16 epochs. Weight decay is approximately 1.98×10^{-7}. Selection uses finite full-validation NDCG and loss. Trade validation NDCG@10 is 0.752 ± 0.007 (five seeds); Genre is 0.466 ± 0.000 (three).

Table 4: Validation and observed final-evaluation costs for the headline seeds. Hours include retained-training replay and full validation/test scoring. GiB is peak allocated GPU memory. Costs mix retained and expanded runs and are not controlled throughput comparisons.

Dataset	n	Validation MRR	Hours	GiB
`tgbl-wiki`	5	0.779 ± 0.004	0.17–0.57	0.19–0.22
`tkgl-smallpedia`	3	0.643 ± 0.004	0.16–0.18	19.6–19.6
`tkgl-polecat`	3	0.251 ± 0.003	1.90–3.45	9.6–9.6
`tkgl-icews`	5	0.316 ± 0.008	9.80–13.45	10.37–11.71
`tkgl-wikidata`	3	0.642 ± 0.002	0.73–0.84	8.6–8.6
`thgl-software`	3	0.381 ± 0.003	0.20–0.28	2.5–2.6
`thgl-forum`	5	0.614 ± 0.014	0.19–5.37	4.16–4.40

Expanded runs record an NVIDIA A100-SXM4-80GB, float32 computation, PyTorch 2.8.0 with CUDA 12.8, PyG 2.6.1, and py-tgb 2.2.0. They run on a shared server with varying concurrent workloads. An equivalent vectorized edge-index construction was introduced during the campaign;

Table 5: Registered parameters and observed training/selection hours for the same headline seeds. Selection includes training-history replay and validation-prefix evaluation. Ranges combine historical stage logs and newly serialized timers; they exclude final evaluation and total development cost.

Dataset	Parameters	Training hours	Selection hours
`tgbl-wiki`	444,686	1.13–3.04	0.84–2.64
`tkgl-smallpedia`	648,803	1.42–1.69	1.58–1.89
`tkgl-polecat`	631,133	12.70–15.02	12.21–14.04
`tkgl-icews`	655,883	0.30–15.55	0.31–14.83
`tkgl-wikidata`	355,253	1.08–1.19	1.17–1.35
`thgl-software`	630,795	2.22–4.51	2.10–4.00
`thgl-forum`	630,271	0.43–7.13	0.14–6.79

historical timings use its slower predecessor. Timers use unsynchronized wall-clock boundaries. These differences preclude a controlled architectural efficiency ranking. New logs separate replay/validation/test and record cache bytes and CPU RSS; older records do not.

The affine input selector has $d_h d_q(d_q + 1)$ parameters (435,584 when $d_h = 64, d_q = 82$; 620,928 when $d_q = 98$). Each updated node requires a dense matrix exponential. REC's committed key/count/latest-time arrays use approximately 24 bytes per key, before temporary buffers. Expanded core-off runs bypass their unused SSM and map computations; their active-parameter counts exclude unused modules. Fixed-width backbone state does not bound the complete predictor's cache or candidate-scoring cost.

E.3 Additional comparison results

The board is not an exhaustive literature ranking: TSNN reports 0.8282 ± 0.0005 on Wiki-v2 and 0.7113 ± 0.0036 on Software (Asif et al., 2026), above our means.

Baseline numbers in Table 6 are taken from the public TGB leaderboard pages for node prediction, link prediction, knowledge graphs, and heterogeneous graphs, accessed September 21, 2026. This is a selection of reference methods rather than an exhaustive literature survey. Their reported uncertainty, when available, belongs to the original evaluation and is not a paired estimate against our seeds. The Wiki reference is the `tgbl-wiki-v2` leaderboard.

Table 6: Selected published reference scores. Node tasks use NDCG@10; all other tasks use MRR. Recurrency denotes the training-tuned baseline; tw and unl. denote time-window and unlimited-history EdgeBank.

Dataset	Leading entry	Neural comparator	History baseline
Trade	NAVIS: 0.863	TGNv2: 0.735	Persistent: 0.855
Genre	NAVIS: 0.528	TGNv2: 0.469	Moving avg.: 0.509
Wiki	TPNet: 0.827	DyGMamba: 0.739	EdgeBank (tw): 0.571
Smallpedia	CEN: 0.612	RE-GCN: 0.594	Recurrency: 0.605
POLECAT	TLogic: 0.228	CEN: 0.184	Recurrency: 0.198
ICEWS	Recurrency: 0.211	CEN: 0.187	EdgeBank (tw): 0.020
Wikidata	EdgeBank: 0.535	—	EdgeBank (unl.): 0.535
Software	STHN: 0.731	TGN (type): 0.424	EdgeBank (unl.): 0.449
Forum	TGN (type): 0.729	TGN: 0.649	EdgeBank (unl.): 0.617

References: NAVIS (Feldman et al., 2025), TGNv2 (Tjandra et al., 2024), TGN (Rossi et al., 2020; Gastinger et al., 2024a), TPNet (Lu et al., 2024), DyGMamba (Ding et al., 2024), CEN (Li et al., 2022), RE-GCN (Li et al., 2021), TLogic (Liu et al., 2022), Recurrency (Gastinger et al., 2024b), EdgeBank (Poursafaei et al., 2022), STHN (Li et al., 2023), and node-history baselines (Huang et al., 2023).

E.4 Component comparisons

All controls retain task inputs and the non-REC scorer. Identity maps use the same augmented Laplacian normalization as TSD. Current-only maps change only the decoder's conditioning, while

memory still enters spatial initialization. No-memory sets that readout to zero and decodes maps from current features; core off additionally removes propagation. Node frames share one decoded orthogonal frame across each node's incidences, and attention gates replace matrix transport with scalar edge weights. GRU and diagonal-SSM controls use ordinary propagation and the same allowed recurrent inputs; the diagonal SSM retains a selective step but uses a fixed input matrix. These are functional comparisons, not capacity- or runtime-matched architectures. On Wiki, active parameter counts are 444,686 (TSD), 33,338 (GRU), 10,126 (diagonal SSM), and 8,419 (core off).

The five-seed comparisons in Table 7 use the same task configuration within each dataset. Wiki's per-arm learning-rate selection consults tracking validation only; no test result selects the rate. Forum and ICEWS expand the retained seeds 43/46/47 with seeds 44/45. Paired differences are descriptive across training seeds, not an equivalence test or a multiple-comparison significance claim.

Table 7: Core comparisons with REC enabled. Test MRR, mean $\pm$ sample standard deviation over paired seeds 43–47. Wiki uses per-arm validation-selected learning rates and an architecture-independent training-negative stream; Forum uses its retained task configuration. Simpler cores have fewer active parameters (Appendix E.4).

Core variant	Wiki	Forum
TSD	0.762 ± 0.003	0.625 ± 0.010
Current-only maps	0.760 ± 0.007	0.634 ± 0.015
Identity maps	0.763 ± 0.002	0.640 ± 0.008
GRU + ordinary	0.764 ± 0.006	0.642 ± 0.012
Diagonal SSM + ordinary	0.762 ± 0.006	0.647 ± 0.006
Attention gates	0.755 ± 0.005	0.630 ± 0.009
Node frames	0.754 ± 0.006	0.640 ± 0.010
Core off	0.735 ± 0.007	0.615 ± 0.001

TSD improves over core off on all five paired seeds on Wiki and Forum, by 0.027 and 0.010 MRR in mean, establishing a benefit from the combined recurrent/spatial core over this anchor. Identity maps and GRU propagation closely match TSD on Wiki. On Forum, identity, GRU, diagonal SSM, and node frames exceed TSD on all five seeds; current-only maps do so on four, and attention on three. The expanded five-seed ICEWS controls give 0.334 MRR for both identity maps and GRU, versus 0.329 for TSD and 0.310 for core off. Thus the strong complete-predictor result on ICEWS also does not isolate a geometric-memory benefit. Together with the variable REC-off comparisons, these results establish no consistent predictive return for history-conditioned maps under the tested configurations (Appendix E.4).

Table 8: Expanded Forum comparisons with REC disabled. Mean $\pm$ sample standard deviation. All available completed seeds for each listed arm are included: $n = 5$ means 43–47; current-only lacks seed 47, no-gap uses 43/46/47, and no-memory uses 43/46. Different seed counts require a matched-subset TSD reference.

Core variant	Test MRR	n
TSD	0.353 ± 0.071	5
Current-only maps	0.393 ± 0.012	4
Identity maps	0.321 ± 0.068	5
GRU + ordinary	0.376 ± 0.026	5
Diagonal SSM + ordinary	0.375 ± 0.059	5
Attention gates	0.330 ± 0.038	5
Node frames	0.382 ± 0.032	5
Core off	0.226 ± 0.008	5
No physical gap	0.367 ± 0.020	3
No memory	0.396 ± 0.018	2

Without REC, TSD exceeds core off on all five Forum seeds (mean gain 0.1267), but its contrasts with the other five-seed cores change sign. Identity exceeds TSD on two seeds, so the map benefit is not consistent across the five-seed sample. For current-only's four seeds, the matched TSD mean is 0.3544; for the no-gap three-seed subset it is 0.3806, and for no-memory's two seeds it is 0.3974. With REC, the retained no-memory subset scores 0.6501 versus matched TSD 0.6293 on seeds

43/46/47. These observations do not establish a general predictive benefit from reading persistent memory into the map decoder.

Table 9: Expanded ICEWS controls with REC enabled, paired seeds 43–47. Test MRR, mean ± sample standard deviation. Other core families were not evaluated in this expansion.

Core variant	Test MRR
TSD	0.329 ± 0.011
Current-only maps	0.328 ± 0.008
Identity maps	0.334 ± 0.009
GRU + ordinary	0.334 ± 0.002
Core off	0.310 ± 0.007

ICEWS TSD exceeds core off on four of five seeds (mean gain 0.0191), while current-only, identity, and GRU contrasts have mixed signs. The separate retained no-memory control has only seeds 43/46: 0.3282 versus matched TSD 0.3364, with losses $0.0091/0.0073$. This limited memory-removal contrast should not be substituted for the expanded five-seed comparison.

Table 10: Additional paired contrasts with REC, seeds 43/46/47. Entries are arm-minus-TSD MRR mean ± sample standard deviation of paired differences; signs count positive/negative differences. Smallpedia uses the separate 15-epoch component budget. This table selects two common controls, not every available run.

Dataset	Control	Difference	Signs
Smallpedia	GRU + ordinary	-0.01136 ± 0.00003	$0+/3-$
	Core off	-0.0358 ± 0.0576	$1+/2-$
Software	GRU + ordinary	$+0.0007 \pm 0.0018$	$2+/1-$
	Core off	-0.0092 ± 0.0018	$0+/3-$
POLECAT	GRU + ordinary	$+0.0004 \pm 0.0049$	$1+/2-$
	Core off	-0.0194 ± 0.0067	$0+/3-$
Wikidata	GRU + ordinary	-0.0040 ± 0.0078	$1+/2-$
	Core off	-0.0061 ± 0.0087	$1+/2-$

The other map comparisons remain heterogeneous. Software's attention gates exceed TSD by 0.0052 on average and on all three seeds; its identity/current-only means differ by less than 0.002. POLECAT's retained identity, no-gap, and current-only controls have only seeds 43/46; identity and no-gap differences change sign, while current-only is higher on both. Wikidata has only seed-43 identity/current-only/no-gap controls. A separate Wiki gap-removal comparison uses three seeds at learning rate 3×10^{-4} (0.7331 full versus 0.7264 no-gap); it is a separate configuration, not an ablation of the 10^{-3} headline. None of these subsets establishes a consistent advantage for the complete memory–map coupling.

E.5 Expanded temporal and history audits

Clock and batch-width comparisons. The global clock measures time between processed snapshots. Node-update measures time since a node's previous memory update, including closure updates; node-interaction measures time since its previous observed endpoint event. First observation has gap zero. All other model and selection settings are held fixed in the Wiki clock comparison.

Table 11: Wiki timing comparisons, seeds 43–47. Test MRR and paired difference from the primary matched global-clock TSD. All means use five seeds; uncertainty is the sample standard deviation.

Clock / batch width	MRR	Difference
Global / 600 s	0.7623 ± 0.0030	—
Node-update / 600 s	0.7654 ± 0.0032	$+0.0031 \pm 0.0016$
Node-interaction / 600 s	0.7636 ± 0.0033	$+0.0013 \pm 0.0045$
Global / 300 s	0.7810 ± 0.0026	$+0.0187 \pm 0.0042$

Node-update and 300-second batching improve all five paired seeds; node-interaction improves three. The separate global-clock seed-43 diagnostic run scores 0.759702, whereas the primary matched seed-43 run scores 0.757525. They share model/protocol settings but are separately trained checkpoints, so we use the primary run in Table 11. A 1200-second pilot scores 0.732249 on seed 43 only. The 300/600-second runs each use 110,232 training edges and eight epochs, but their recorded training steps are 49,664/24,832. Batch width also changes the historical observations available before each prediction, local graphs, loss grouping, and closure updates. Its effect is not isolated to the physical-gap selector.

Step-size saturation. Diagnostics of the separate global-clock seed-43 run record 127,646/127,665 endpoint updates and 894,215/894,276 closure updates capped during training-history replay. At test, all 28,587 endpoint and 175,893 closure updates use step 0.25. These are final-checkpoint evaluation statistics; no corresponding diagnostics were saved for the five primary matched TSD checkpoints. They do not reconstruct training-time gradients.

Saturation is strongly seed-dependent. Node-update seed 43 has only 6/28,587 capped endpoint test updates and no capped closure updates; seeds 44–47 instead cap roughly 96–97% of endpoint updates and at least 99.93% of closure updates. Node-interaction and 300-second runs also mix mostly capped and mostly uncapped seeds. The cap can remove local step-size sensitivity while preserving state dependence through the recurrent input; these measurements do not identify saturation as the cause of any score gap (Appendix D).

Query-time and gap interventions. For a retained Wiki seed-43 checkpoint at learning rate 3×10^{-4}, a fixed-state probe shifts query times by 0, one hour, one day, or 30 days over the first 20 validation batches (700 queries). Neural logits are unchanged exactly; MRR changes from 0.6729 to 0.6686, 0.6666, and 0.6632 through REC. This tests the scorer interface with fixed historical state, not an altered event history. A second intervention keeps snapshot membership fixed and scales the gap supplied to memory by $0.25, 0.5, 1, 2, 4$ on the first 30k validation events after training replay. MRR is $0.7516, 0.7518, 0.7521, 0.7548, 0.7553$. The small range does not prove the training effect of removing the gap or generalize beyond this checkpoint and prefix.

History strata. The Wiki analysis joins per-query ranks on identical full-test edge IDs for seeds 44–47. A recurring pair has appeared earlier in raw event order; this may include earlier events within the same snapshot, so recurrence does not imply visibility to the current neural state. Novel pairs have no earlier directed occurrence. Source-gap quartiles use time since the source's previous source event, not the model's clock. Recurring pairs comprise 89.2% of test queries.

Table 12: Wiki full-test strata, four seeds (44–47), with REC. MRR averages are over the same queries within each stratum. The longest source-gap quartile starts above 8,162 seconds.

Core	Recurring pair	Novel pair	Longest gap
TSD	0.8542	0.0118	0.6688
Identity maps	0.8539	0.0103	0.6673
GRU + ordinary	0.8529	0.0105	0.6688
Core off	0.8194	0.0140	0.6286

TSD exceeds core off by 0.0349 on recurring pairs and 0.0402 in the longest-gap stratum on all four seeds. On novel pairs, core off is higher by 0.0022 on average and on three seeds. Identity/GRU scores closely match TSD in the recurring and long-gap strata, so these gains do not isolate history-conditioned maps or a particular temporal mechanism.

A separate Forum prefix analysis used one seed and the first 1M test events. It found REC-on novel-pair MRR 0.138 for TSD versus 0.090 for core off, with overall prefix scores 0.623/0.607. Its novelty and source gaps use only history committed before each snapshot; 42.8% of that prefix is novel. This task-, prefix-, and definition-specific result is not interchangeable with the full-test Wiki analysis and cannot establish a general novelty-transfer benefit.

Delayed-cue synthetic task. The delayed-cue generator uses fixed random features, hidden communities, irregular event gaps, and observable cue interactions that set a source's target community.

Later destinations are drawn from that community using a hidden popularity distribution. The base instance has 400 nodes, eight communities, and 60k events; 62% of test interaction pairs are novel. Evaluation uses fixed sets of 32 uniform negatives, 20-second batches, and prediction before ingestion. Training uses six epochs and one-step BPTT. Repeated training seeds share the same generated data, so their variability does not measure uncertainty across generators.

Without REC, the five-seed base-instance MRRs are 0.2098 ± 0.0008 (TSD), 0.2091 ± 0.0022 (GRU), and 0.2125 ± 0.0020 (core off). Core off exceeds TSD on all five seeds. A shorter-delay instance uses 100 nodes, five communities, and typed cues available through relation inputs; its novel fraction is about 11%. Table 13 reports the five model variants on five training seeds.

Table 13: Shorter-delay cued-history instance with REC disabled. Five training seeds on fixed data; mean $\pm$ sample standard deviation. The history reference respects the prediction cutoff but also knows generator communities, which are hidden from the models.

Predictor	Interaction test MRR
TSD	0.265 ± 0.007
Current-only maps	0.270 ± 0.010
Identity maps	0.276 ± 0.023
GRU + ordinary	0.281 ± 0.009
Core off	0.268 ± 0.005
Generator-informed history reference	0.531

GRU exceeds TSD by 0.0160 on all five seeds in the shorter-delay instance; TSD does not exceed core off in mean. The history reference infers the target from cues before the current batch and uses observed indegree, but also accesses true candidate-community assignments. It is temporally causal with privileged generator knowledge, not an information-matched attainable ceiling. These experiments provide no positive evidence for temporal map conditioning under the tested budget. They neither identify one-step truncation or the timing cap as the cause of failure nor show that every realization of the continuous model must fail.

E.6 Planted-transport synthetic study

Generator and protocol. The generator has 100 nodes in four communities, four-dimensional unit-norm latent states, and planted orthogonal intercommunity transports. Evolving states rotate at 0.01–0.05 radians per time unit; static states have zero rotation rate. Both have small event-driven perturbations and bursty event gaps. A source repeats a past partner with probability 0.1; otherwise it samples by transported-state similarity at temperature 0.03. Features expose initial states and communities. We use 100k events, 20-time-unit snapshots, a chronological 70/10/20 split, seed 43, 32 feature channels, and eight training epochs. Prediction precedes updating; evaluation ranks destinations against all other nodes.

Table 14: Synthetic test MRR under matched model controls (seed 43). All columns use the same observed features and REC head. Current-only changes map conditioning and retains memory in the spatial state.

Regime	Full	Identity maps	No gap	Current-only
Static	0.849	0.846	0.849	0.847
Evolving, rotations	0.276	0.277	0.276	0.276
Evolving, identity transports	0.290	0.290	0.288	0.289

All variants in Table 14 share REC and observed features. Their near-equal scores do not establish benefits from learned maps, gap input, or temporal map conditioning under this budget. Current-only maps retain memory, so this is not evidence that memory is unnecessary. In the evolving-rotation test, only 0.375% of events involve novel pairs. Pair repetition can occur outside the explicit 10% repeat branch; that branch probability is not the realized repeat fraction. These properties limit interpretation as latent-transport recovery: the study is a null result for mechanism separation.

Known-state diagnostics on the last 15% of evolving-rotation events yield MRRs of 0.178 from initial states, 0.461 from previous-snapshot states, and 0.821 from event-time states and transports. These are reference scores, not a bound on model performance: the model tests on the last 20%, and diagnostic recurrence labels mark the generator's repeat branch rather than all previously observed pairs. No recovered fraction of an oracle ceiling is claimed.

Rescaling test timestamps by $0.25/4$ gives full-model MRRs $0.264/0.281$ and no-gap MRRs $0.266/0.281$, both from 0.276 originally. Rescaling also changes batching and REC timing. The similar no-gap response therefore does not demonstrate a benefit from the SSM gap selector.

F Reconstruction with prescribed changing geometry

F.1 Sensor histories and comparison protocol

The reconstruction study directly evaluates the prescribed-geometry memories of Section 3. It uses a fixed undirected graph obtained by symmetrizing four-nearest-neighbor connections among 20 random points in the unit square, with two-dimensional stalks. Each sensor has an initial random orientation $R_u(0)$. A randomly chosen fraction of sensors changes orientation by an independently signed angle θ at time τ on the unit window. Endpoint restrictions are $R_{e\leftarrow u}(s) = R_u(s)$. We use $\mathcal{L}(s) = \delta_s^\top \delta_s / d_{\max}$, where $d_{\max}$ is the maximum graph degree. This scalar normalization preserves physically consistent readings in $\ker \mathcal{L}(s)$ and bounds its spectrum by 2. It is a prescribed PSD operator covered by Theorem 3.1; the predictor uses its own block-degree normalization. The sensor maps form a node-frame sheaf, so this study does not test the additional cycle transports of incidence-specific maps.

The clean reading is $y_u(s) = R_u(s)^\top g(s) + d_u(s)$ in scenario A (physical change), and $y_u(s) = R_u(1)^\top g(s) + d_u(s)$ in scenario B (retrospective calibration). Each component of g is a sum of five random sinusoids with frequencies between 0.5 and 4 cycles per window; each sensor deviation d_u uses three sinusoids with amplitude scale 0.2. Observations are sampled at the midpoints of 400 equal intervals, with independent additive Gaussian noise, and held constant on each interval. Noise scale is set from scenario A's clean-signal power; the two scenarios share the same additive noise draw. The supplied operator path uses the orientations believed at observation time in both scenarios. Thus B deliberately makes the earlier supplied calibration outdated, while the final calibration describes its whole clean history.

All compressed methods use the same N polynomial modes and observation history. Historical memory solves $K \operatorname{vec} H = \operatorname{vec} C$ with columnwise vectorization; current filtering uses $Q(1)^{-1}C$, input filtering compresses $Q(s)^{-1}X(s)$, and plain LegS sets $\alpha = 0$. A pointwise, uncompressed filter is also retained in the saved records as a separate reference. The reconstruction sweep computes the Gram state using a quadrature-equivalent held-history recursion and solves it through a symmetric eigendecomposition; direct matrix-exponential updates and moment-based conjugate gradients (CG) are checked separately below.

Regularization is selected independently for each configuration, scenario, and method from $\{0, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100\}$ by mean whole-window relative error on ten validation draws (seeds 100000–100009). Evaluation uses 50 disjoint draws (seeds 0–49), shared across methods. Clean validation histories are available for this selection. At 2,000 uniformly spaced midpoint evaluation locations, the metric on segment S is $\|Y_S - \widehat{Y}_S\|_F / \|Y_S\|_F$, comparing reconstruction with the underlying clean signal. The pre-change segment is $s < \tau$ for a single switch. Standard deviations describe variability over generated arrays, fields, and noise, rather than training randomness. Paired 95% bootstrap intervals use 4,000 resamples of the 50 draw-wise error differences and condition on the selected α; they are pointwise intervals without a multiple-comparison adjustment.

F.2 Scope and sensitivity of the crossover

The sweep varies rotation ($0°$–$90°$, at both 10 and 0 dB), changed-sensor fraction $(0.1, 0.25, 0.5, 1)$, switch time $(0.2, 0.5, 0.8)$, mode count $(4, 8, 16, 32)$, and SNR $(-10, -5, 0, 10, 20, 30$ dB), with one three-dimensional-stalk configuration. Four further configurations perturb the supplied maps, delay their switch by 0.1, replace a jump by gradual rotation over 0.3 of the window, or use five smaller reorientations. The gradual path is sampled and held on the same 400 intervals; it tests a resolved

Table 15: Clean-history reconstruction at $\theta = 45°$, $N = 16$, change fraction 0.5, and $\tau = 0.5$. Relative error, mean $\pm$ sample standard deviation over 50 evaluation draws; lower is better. SNR is in dB. A denotes physical change and B retrospective calibration. Each method is tuned separately on whole-window validation error.

SNR	Scenario	Segment	H_{hist}	H_{in}	H_{cur}	LegS
10	A	Pre-change	0.0901 ± 0.0228	0.0901 ± 0.0228	0.0920 ± 0.0229	0.0914 ± 0.0229
10	A	Whole window	0.0854 ± 0.0123	0.0854 ± 0.0123	0.0861 ± 0.0125	0.0865 ± 0.0126
10	B	Pre-change	0.0666 ± 0.0100	0.0666 ± 0.0100	0.0650 ± 0.0102	0.0669 ± 0.0104
10	B	Whole window	0.0630 ± 0.0020	0.0630 ± 0.0020	0.0619 ± 0.0024	0.0636 ± 0.0020
0	A	Pre-change	0.1807 ± 0.0338	0.1807 ± 0.0340	0.2156 ± 0.0303	0.2204 ± 0.0359
0	A	Whole window	0.1718 ± 0.0167	0.1717 ± 0.0167	0.1939 ± 0.0089	0.2089 ± 0.0085
0	B	Pre-change	0.2024 ± 0.0261	0.2024 ± 0.0261	0.1681 ± 0.0317	0.2106 ± 0.0328
0	B	Whole window	0.1842 ± 0.0075	0.1842 ± 0.0075	0.1605 ± 0.0178	0.2002 ± 0.0062

drifting history. For drift and multiple changes, the reported early segment extends through the end of the transition or the last reorientation, respectively.

After counting the duplicated 0 dB/45° configuration once, 28 of 31 distinct nonzero-rotation sensor configurations have paired intervals favoring H_{hist} in A and H_{cur} in B on that early segment. The same seeds are reused across configurations, so this is a descriptive sensitivity count. At $N = 4$ and 8, A favors historical memory but the B comparison is unresolved; at 30 dB the compressed methods select $\alpha = 0$ and coincide. At 20 dB the differences remain resolved but are only about 2×10^{-5} in A and 7×10^{-5} in B. All four perturbations preserve the early-segment crossover at the 10 dB baseline. With a delayed switch, A's post-change comparison reverses: applying outdated maps can harm reconstruction once the physical change has occurred.

The historical and input-filter constructions are mathematically distinct, but the central experiment offers little incentive to pay for the coupled solve. At 0 dB/45°, their mean early-segment differences are below 2×10^{-5} in both scenarios. The largest departure in the sensor sweep occurs at -10 dB in A, where input filtering improves relative error by about 0.013 over historical memory. They also separate when regularization is strong, so their agreement is task- and tuning-dependent. A further Gaussian coefficient-prior check aligned with historical regularization is included in the saved records; it is a numerical consistency check, rather than independent evidence of task generality. No learned nodewise TSD reconstruction model or real-sensor dataset is evaluated here.

F.3 Numerical equivalence and resource cost

Five float64 checks use $p = 40$, 200 held intervals, $\alpha = 3$, and $N \in \{8, 16, 32\}$, including single changes, sampled drift, and multiple changes. Direct batch integration uses per-interval Gauss–Legendre quadrature, exact for the held-input polynomial integrands. Relative discrepancy means the maximum absolute entrywise difference divided by the reference's maximum absolute entry. The matrix-exponential recurrence in Eq. (8) matches batch C and K to below 2.0×10^{-14}; the operator moments reconstruct K to below 3.9×10^{-13}. The quadrature-based recurrence used in the sweep agrees to below 8.0×10^{-13}. Moment-based CG agrees with a dense solve to below 1.1×10^{-12}, and the current-geometry recursion agrees with the direct resolvent to below 4.4×10^{-14}. Eigenvalues obey $1 \leq \lambda(K) \leq 7$ to floating-point tolerance. These checks validate implementation of the held-history identities; they do not measure convergence from a continuously varying path to its held approximation.

Table 16: Saved cost measurements for 100 held intervals, float64, $\alpha = 1$. Geometry-state sizes in MiB; recursion and final solve times in seconds. The moment recursion stores dense operator moments; CG uses their sparse representation. These are individual server measurements, without timing uncertainty estimates.

		Geometry state		Recursion		Final solve	
p	N	Gram	Moments	Gram	Moments	Dense	CG
40	16	3.12	0.38	5.01	0.72	0.035	0.014
80	32	50.00	3.08	99.30	3.45	1.302	0.145
150	32	175.78	10.81	54.65	2.62	2.104	0.044

The geometry-state measurements in Table 16 count only the stored Gram blocks or operator moments. They exclude signal moments, basis-product tables, and solver workspace. Gram-recursion timings include signal and geometry updates; moment-recursion timings cover geometry alone and exclude sparsification and product-table preparation. Final-solve timings start from the already formed states.

Across the nine tested (p, N) combinations, CG uses 15–16 iterations at $\alpha = 1$ and 33–46 at $\alpha = 10$, with a relative residual stopping tolerance of 10^{-12} and maximum 500 iterations. This is consistent with the bound $\mathrm{cond}_2(K) \leq 1 + 2\alpha$ for these normalized operators; observed iteration counts also depend on the spectrum and right-hand side. Entries of magnitude at most 10^{-14} are removed when converting dense moments to sparse matrices for CG. The cost records use single-threaded BLAS, but do not record enough hardware and timing metadata for a portable speed claim. Allocation peaks measured by Python tracing exclude pre-existing arrays and are not total process-memory estimates. The moment representation reduces persistent geometry storage and permits a matrix-free solve, while the reconstruction comparisons establish no general accuracy advantage over the less expensive input-filter construction.

F.4 Clean-environment reproduction

We regenerated the synthetic reconstruction study from a fresh clone at a fixed source revision in a new Python 3.9 environment, using NumPy 2.0.1, SciPy 1.13.1, pandas 2.3.3, Matplotlib 3.9.2, and pytest 8.3.3. The rerun used PyPI wheels with OpenBLAS, while the saved reference records used MKL. Gram matrices are explicitly symmetrized before eigendecomposition, and the Gaussian coefficient-prior check uses a symmetric inverse square root to make its draws independent of the eigenvector basis chosen within degenerate eigenspaces.

The reproduction passed all ten unit tests and completed all 2,160 sweep jobs, followed by regularization selection, numerical-equivalence checks, cost measurements, and figures. All 630,000 per-draw error records matched their reference keys, with maximum absolute error difference 3.15×10^{-10}. All 350 regularization choices were unchanged. The maximum absolute differences were 6.30×10^{-12} for tuned means and standard deviations and 2.21×10^{-11} for paired means and interval endpoints. All 32 Table 15 means and the quoted paired contrasts agreed to the displayed precision; the 28/31 crossover count, numerical-equivalence bounds, and CG iteration counts were preserved. Sign counts were compared for contrasts with reference mean magnitude above 10^{-9}; 84 near-zero contrasts were excluded from that sign comparison. Hardware-dependent timing and allocation measurements were regenerated but were not agreement targets. This check covers the synthetic reconstruction pipeline described in this appendix.